

The Design and Development of a 9 by 9 GO Opening Game Knowledge Base System

Chun-Hsiang Hsieh and Jeng-Chi Yan

Abstract—This study used the hashing method to produce a 9 by 9 GO opening game knowledge base system, and conducted in-depth study to solve the various board situation symmetry problems. This system can analyze the winning rate of various moves in the opening game. If a sufficient number of chess manuals can be inputted into the system, it can effectively provide move choices in the computer GO opening game, and can be used for the references of GO players on opening games. In sum, according to this knowledge base system of this article, this study considers that computer GO may defeat human experts perfectly in the future.

Index Terms—Hashing method, knowledge base system in opening game, computer GO.

I. INTRODUCTION

A. Relevant Knowledge and Rules of GO

GO is a game of interpleading territory on a 19 by 19 board with black and white sides. It is now widely recognized as the most difficult and in-depth chess game. With simple rules, the variations in GO are very complex, and contain rich strategic and philosophical thinking, which is the most appealing feature of the game.

In the game, some stones may be captured by the opponent. If a number of stones of the same color are connected on the board, they are known as a “string”. Each string has the so-called “liberty”, which is defined as the number of the empty points around the string. In the gaming process, if the liberty of a certain string is 0, because it is surrounded by the opponent, the string will be captured by the opponent, and should be removed from the board.

At the end of the game, when there is no territory for interpleading, the game is ended. In this case, the points surrounded by both sides are calculated (the area of one surrounded point is known as one moyo), the player with more moyos is the winner [1].

In general, the GO board of 19 by 19 in size is wide in territory from corner and side to the center. For beginners, a 19 by 19 board is too big, thus, the 9 by 9 board may be more suitable. In addition, regarding whether the computer GO program can beat human experts, 9 by 9 GO can be regarded as a good touchstone in terms of complexity. Hence, more and more GO players are engaged in the study of 9 by 9

computer GO games [2], [3].

B. Concept of Knowledge Base System in Opening Game

In the game of 9 by 9 GO, as the board territory is relatively smaller, if the opening game is not well-played, the player may easily land in a disadvantageous situation at beginning. However, in this case, if the global search system is used, the results are not necessarily ideal; one of the reasons is that there are too many points for selection, and it is difficult to distinguish good choices. As a result, the search time will increase considerably, while the results are not necessarily correct. For example, as shown in Fig. 1, after the move of black 1, it is white’s turn, even if the symmetric parts are excluded, the white move can be in any place from A to N, which will be a very heavy load for search, and commonly used search skills, such as α - β pruning, can have very little effect in this case.

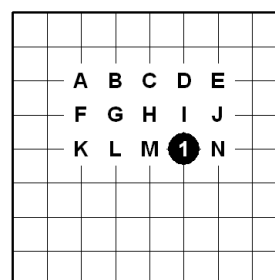


Fig. 1. Selection of move point in the opening game.

Another problem is the depth of the search. At the stage of the opening game, due to the limitations of search depth, it is difficult to determine the disadvantages and advantages of the board situation. With the board situation shown in Fig. 2 a) as an example, at the time of black 7, the advantageous side in this situation is very difficult to analyze; however, after the black plays at 23, as shown in Fig. 2 b), the board situation is much clear. However, as limited by the situation judgment function speed and the gaming time limit in general, the computer global search system cannot actually search in depth. Hence, it is very difficult to judge the future situation of the game.

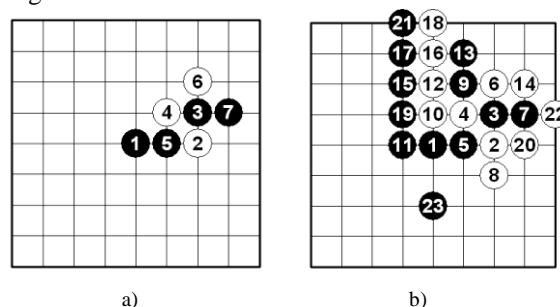


Fig. 2. The problem of search depth at the opening game stage.

Manuscript received April 9, 2013; revised June 24, 2013.

C. H. Hsieh is with the Institute of Computer Science and Information Engineering, National Chung Cheng University, Chia-yi, Taiwan (e-mail: hsiehch@cs.ccu.edu.tw).

J. C. Yan is with the Department of Information Technology, Overseas Chinese University, Taichung, Taiwan (e-mail: jengchi@ocu.edu.tw).

In view of the above reasons, this study believes that chess manuals should be inputted at the opening game stage. Only by following the instructions of the chess manuals can the deficiency at the opening game of a complete search be compensated. With the computer game of Chinese chess as an example, at the opening game, the program selects moves according to the instructions of the chess manuals. Until the move of the opponent is included in the chess manuals, the program will make decisions through the global search system, otherwise, it can easily land in some setups of the opening game (the terminological name of Chinese chess is known as “setup”) [4], [5]. Therefore, for a 9 by 9 computer GO game, it is necessary to establish a good knowledge base system in the opening game to obtain the skills and capacity of a professional player.

This paper uses the hashing method to develop the 9 by 9 GO opening game knowledge base system, and conducts in-depth study to solve various board situation symmetry problems. This system can analyze the winning rate of the various moves in the opening game. If a sufficient number of chess manuals can be inputted into the system, it can effectively provide move selections in the computer GO opening game, and can be used for the references of GO players on opening games.

II. METHOD

As described in the above section, the establishment of a good opening game knowledge base can greatly enhance the gaming power improvements of the 9 by 9 computer GO program. However, even at the opening game, the wide range of move selection and their derived changes are beyond imagination. To professional GO players, 9 by 9 GO is very simple. Fig. 3 a) and Fig. 3 b) illustrate the games of two 9-duan expert players. Since the double-line lower touch after White 4, the two players compete fiercely, during which each move contains complex changes and calculations. The black side attempts to expand territory in the upper left area until it plays at 21, as shown in Fig. 3 a), the situation seems to be generally set in development direction.

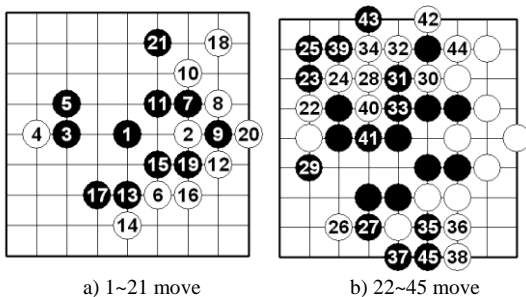


Fig. 3. An actual game of two 9-duan expert players.

However, since the white 22 climb, 24 cut, as shown in Fig. 3 b), the white side finds a way out of the dead-end situation. After the moves of white 30, 32 cut, it is indisputable that the white side has been successful in fighting its way out. Nevertheless, the black side maintains a solid base by reducing loss to the minimum, and winning the maximum endgame by moves of 35 and 37 cut to win the game by komi 6.5 points.

As shown above, in the 9 by 9 GO game, the two players usually compete fiercely, starting with the opening game, and the changes are complex. To list all the calculations, the computation sum is difficult to imagine; hence, it requires a large storage capacity to establish an opening game knowledge base rich in data, and the search of data should be very fast. Hence, this study used a very large file for the storage of the opening game chess manuals, and use the hash function to accelerate data search. The concept is as shown in Fig. 4.

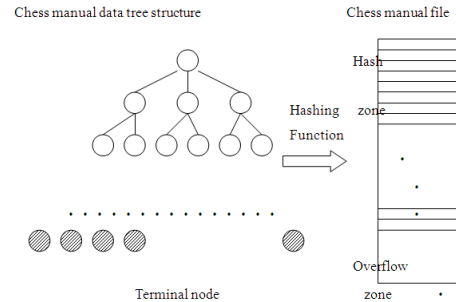


Fig. 4. Design concept of opening game knowledge base.

As shown in Fig. 4, we present each move in the chess manuals according to branches of the tree structure. The board situation is formed after the move is represented by the circular blank node. The final move of each game in the chess manuals is the terminal node of the tree structure, which is represented by a circular slash node. Then, regarding the board situation of each game (circular slash node), we use the hash function to calculate the address, and input the node information in the address file of the chess manual. If the address has been input from another board situation, it will result in collision, and thus, the node information will then be stored in the overflow zone. Regarding the information stored in each node, it contains information of the actual board situation, the winning and losing games of the board situation, and the storage addresses for all the moves and the following moves of the board situation.

Regarding the hash function design details, we first use the five unsigned long int to store the board situations of the 9 by 9 board. As each point of the board may have three situations: blank point (0), black (1), white (2), the total number of board situation combinations is 3^{81} . The size of each unsigned long int is 32 bits, and thus, five unsigned long ints occupy a total of 160 bits, as $2^{160} > 3^{81}$, the board situations can be recorded.

It is both unnecessary and impossible to store all possible board situations (3^{81}). We refer to the design method of computer Chinese chess opening game chess manuals and their file sizes, and determine to set the size of the chess manual file at 2GB, which contains the information of more than 16 million nodes. As each node occupies an address, it requires $2^{24} > 16,000,000$ bits; at the same time, we set 2/5 of the space as the overflow zone to process collision nodes. Hence, there are 961,287 nodes in the hash zone, which must compress the 3^{81} into a value below 10 million, and is the job of the hash function.

Up to the present, the total number of nodes input into the chess manual is 24,742, thus, 37 nodes may collide, which is relatively good, in terms of percentage, as the total number of games to be input is only 1000, and thus, it is too few to judge

the quality of the hash function. However, in terms of speed, the data search and decision making reaction time can be around 0.3 seconds; therefore, its efficiency and practical value are very high.

III. THE PROBLEM AND SOLUTION OF THE SAME BOARD SITUATION

The development of an opening game knowledge base system is not as simple as imagined. The tree structure, as shown in Fig. 4, only illustrates the concept, in fact, the board situations of some nodes are the same, in other words, and they should be combined. As shown in Fig. 5 a) and Fig. 5 b), although the sequences of the moves are different, the results are of the same board situations. For the uniformity and consistency of the subsequent move information, we naturally combine the two nodes. When storing each node, we use the hash function to calculate the storage address according to the board situation. Hence, situations, as shown in Fig. 5 a)-Fig. 5 b), will be stored in the same node, and thus, the problem is naturally solved.

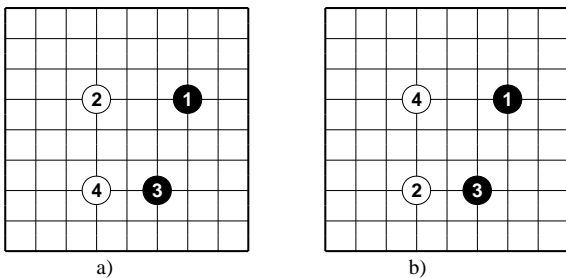


Fig. 5. Same board situations formed by different move sequences.

However, it will be more complex if we further expand the concept of the same board situation. With Fig. 6 a) and Fig. 6 b) as an example, the two board situations are actually the same. However, after rotation, the address calculated by the hash function will be different; in other words, if unprocessed, they will be regarded as two different board situations, and node information cannot be fully integrated.

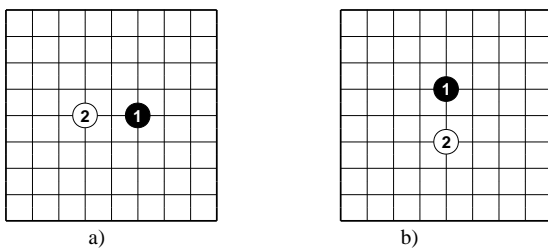


Fig. 6. The same board situations formed due to rotation and symmetry.

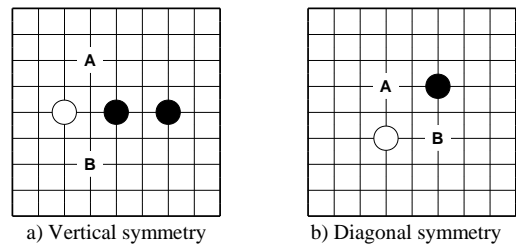
To solve this problem, after detailed observation and analysis, we develop the concepts of “symmetry” and “target region”, by which we can convert the different board situations formed by symmetry and rotation into the same, as elaborated in detail below.

A. Board Situation Symmetry Problem

In the gaming process, symmetrical board situations may frequently occur. With Fig. 7 a) as an example, regardless of white playing at A or B, the result is the same, namely, “vertical symmetry”. Such symmetry is not limited to

vertical.

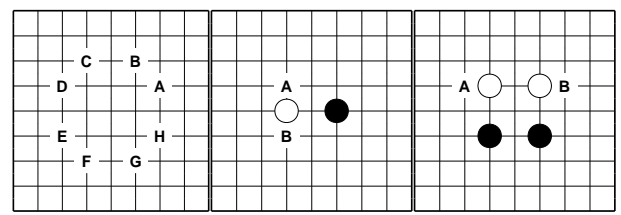
Fig. 7 b) illustrates the “diagonal symmetry”, when black plays at A or B is entirely identical in meaning.



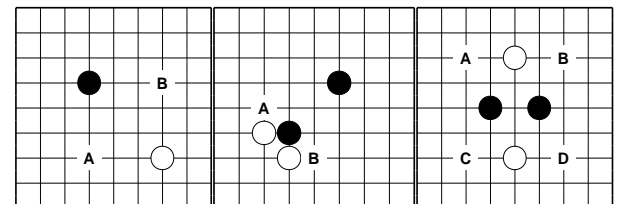
a) Vertical symmetry b) Diagonal symmetry

Fig. 7. Examples of symmetric board situations.

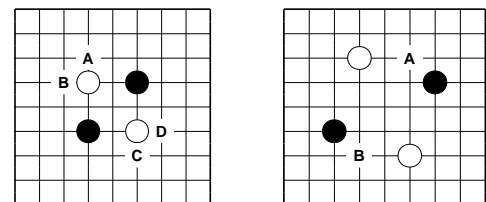
After detailed summary and analysis, it is found that there are eight types of symmetric board situations, as shown in Fig. 8 a)-Fig. 8 h). The points of the same meaning are represented by English characters when the next move is at the black side.



a) Octa-symmetry b) Vertical symmetry c) Horizontal symmetry

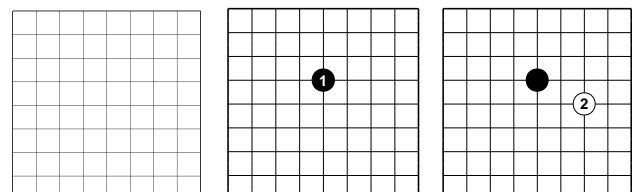


d) Positive skew symmetry e) Negative skew symmetry f) Horizontal and vertical symmetry

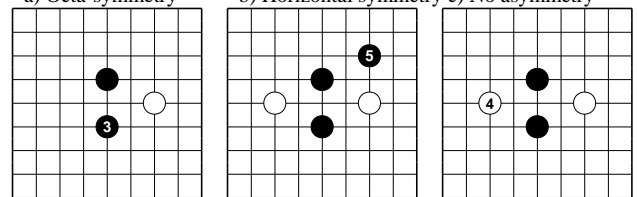


g) Positive/negative skew symmetry h) Center point symmetry

Fig. 8. Symmetrical board situations.



a) Octa-symmetry b) Horizontal symmetry c) No asymmetry



d) Vertical symmetry e) Horizontal and vertical symmetry f) No asymmetry

Fig. 9. Change in symmetrical relationships during the game.

Such symmetrical situations may occur at the beginning of a game. For example, the “octa-symmetry”, as shown in Fig.

8 a), may change gradually over moves. Interestingly, changes in such symmetrical situations are often expected. For example, as shown in Fig. 9, in the beginning, it is of an “octa-symmetry” (Fig. 9 a)), and the symmetrical situation continuously change over the moves of the black and white players, possibly from symmetry to asymmetry or vice versa. Hence, when using the hash function to process the board situation, the symmetrical situations should be considered from the start to the end.

B. Setting of Target Region

As shown above, once there is a symmetrical board situation, in most blank points on the board, there is at least a point of symmetry. If it is not processed in the opening game knowledge base, the number of subsequent nodes will be increased, and result in the impossibility of correct information integration. For the various symmetrical situations, as shown in Fig. 8, we need to set the “target region”. With the blank board, as shown in Fig. 10 as an example, the current board situation is of “octa-symmetry”. Next, before the move of the black side, regardless of the position of the move, we need to convert the location into a location with marks of Δ by rotation and mirror. All locations with Δ are the so-called “target regions”.

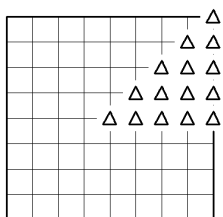


Fig. 10. Examples of target regions.

The eight types of symmetrical board situations, as shown in Fig. 8 a)-Fig. 8 h), have their respective target regions, as summarized in Fig. 11:

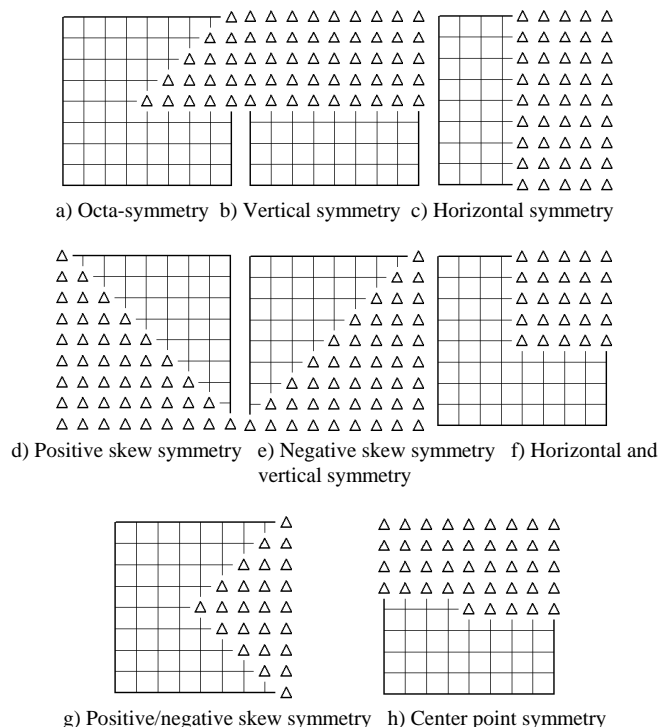


Fig. 11. Target regions of all symmetrical situations (marked by Δ).

After setting the various target regions of various symmetrical board situations, when faced with symmetrical board situations in subsequent input of chess manuals, it will automatically move to another target region. In this way, the board situations can be unified and integrated to render the information of chess manuals in the opening game knowledge base more accurate.

IV. TESTING RESULTS AND DISCUSSIONS

In the knowledge base system of an opening game, when we input the chess manual of each game and the number of moyos of the game, the system will reflect the winning or losing game to all the nodes of the game. Until the number of accumulated chess manuals of the system is sufficiently large, it can analyze the winning or losing probability of each move for selection.

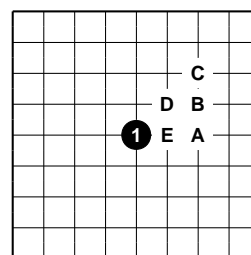


Fig. 12. Selection of moves in the opening game knowledge base.

As shown in Fig. 12, when black 1 is moved at the Center point, according to the information of the opening game system, the White player may choose from A to E. In the case of computer decision making, our system will review the number of winning or losing chances of the five moves to create statistics of the winning rates as the weights for random number decisions. For example, the white moves at point A (win, lose) are (32, 34), and the winning rate is 48 %. We then use 48 as the random number weight for the selection of Point A, and the winning or losing information of the remaining points from B to E are listed, as shown in Table I. The current winning rate at Point E is 0; therefore, it will be excluded in the selection of points for the next move; points from A to D can be selected by the system. The possibility of selecting Points C and A is relatively higher.

TABLE I: INFORMATION ABOUT THE LOSING OR WINNING OF VARIOUS POINTS IN THE OPENING GAME SYSTEM

Point	A	B	C	D	E
Number of winning games	32	8	2	1	0
Number of losing games	34	24	2	2	1
Winning rate (%)	48	25	50	33	0

The algorithm to select the move point is simple and reasonable; however, its disadvantages can be found, as shown in Table I, the number of games at Points C, D, and E is too low to reduce the value of reference. Of course, we have completed this part of the opening game knowledge base in fewer than two months, and with chess manuals of

only 1000 games having been input. Therefore, the number of winning and losing moves is relatively lower. However, we believe that, after the accumulation of a large number of chess manuals, the selection of the move point will be more accurate, and the system can better provide suggestions on moves at the opening game stage, which will be a great help to the computer system in beating human experts.

This knowledge base system uses the concept of hashing and target regions, which effectively solves all symmetric problems in board. This system was designed for 9 by 9 GO. However, it may have spotlight for 19 by 19 GO, Go-moku, connect 6, othello and others. Especially for 9 by 9 GO, the ranking of computer GO is very close to expert 9-dan. In sum, according to this knowledge base system of this article, this study considers that computer GO may defeat human experts perfectly in the future.

REFERENCES

- [1] J. C. Yan and S. J. Yen, "Design and implementation of pattern knowledge base," in *Proc. 2003 National Computer Conference (NCS2003)*, pp. 1784~1792.
- [2] Y. L. Lai, "The Design of Graphic Theory-based 9 by 9 GO Program and the Study on Machine Learning Strategy," *NCS Project Research Achievement Report*, NSC93-2213-E-415-004.
- [3] T. T. Liu, "Evolutionary learning in playing 9*9 GO games," Master's thesis, Department of Computer Science, National Tsing Hua University, 1995.
- [4] S. C. Hsu and K. M. Tsao, "The development of computer Chinese chess knowledge base system in opening game," *Bulletin of the*

College of Engineering, National Taiwan University, 1992, no. 53, pp. 75~86.

- [5] J. C. Chen, "Design and implementation of knowledge-base system for computer Chinese chess," Doctoral dissertation, Department of Computer Science & Information Engineering, National Taiwan University, 2005.



Chun-Hsiang Hsieh is working in the Institute of Computer Science and Information Engineering, National Chung Cheng University. His area of research focuses on artificial intelligence, NLP, computer game.



Jeng-Chi Yan is now working in the Overseas Chinese University, Taichung, Taiwan.

He has published following articles: 1 Research and design of computer go utilities using heuristics, pp.9-20, *Journal of Humanities and Applied Sciences*, Overseas Chinese University, Taichung, 2011/ 6. 2 The research of curing the cold with expert system, *The 3rd Information Education and Technological Applications Conference*, Taichung, 2009. 3 The Sacrifice Move, *ICGA Journal*, Vol. 28, No.4, 2005/12, pp. 222-233. Current and previous research interests: expert system, computer GO. He was supervisor of Taiwan connect6 association 2007.