Using Human Knowledge to Improve Opening Strategy in Computer Go

Jessica Mullins[1]

Undergraduate '10

Lewis & Clark College

0615 SW Palatine Hill Road

Portland, Oregon 97219

jmullins@lclark.edu

Peter Drake

Associate Professor of Computer Science

Lewis & Clark College

0615 SW Palatine Hill Road

Portland, Oregon 97219

drake@lclark.edu

2010 International Conference on Artificial Intelligence

Abstract

The opening is crucial in the game of Go. Monte-Carlo techniques do not fare well in the opening because there are far too many moves to consider. We extracted human knowledge from a database of professional games to build a book of strong opening moves. Our first opening book only had information about full-board opening sequences (fuseki). This book was quickly exhausted during games. We therefore incorporated joseki: common local sequences that lead to outcomes satisfactory for both players, such as striking a balance between corner territory and center influence. Both the fuseki and joseki books use transposition tables allowing us to quickly determine whether the current global or local board configuration exists in our database. We then combine these two books to use information from both databases. We present experimental results on the effects of the three books on the program's win percentage.

Keywords: artificial intelligence, Go game, fuseki, joseki, opening strategy

[1]Contact Author

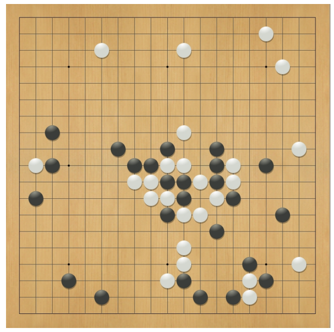

Figure 1: Orego without an opening book, 50 moves into a game.

1 Introduction

Go is a board game that originated in China, played by two players on the intersections of a grid (Shotwell, 2003). The board is conventionally 19x19, but Go can be played on smaller boards such as 9x9. For the purpose of this paper we will focus on games played on 19x19 boards. The object of the game is to control a larger portion of the board by surrounding intersections with stones and creating territory. Game play consists of each player alternating placing a stone of their color, black or white, on the board or passing and ends when both players pass consecutively.

Although there exist computer programs that play chess better than humans, computer Go programs still strive to be competitive with professionals. In fact, it was not until 2008 that a computer program, with the help of a large handicap, was able to defeat a professional player on the full 19x19 board (Garlock, 2008). One technique that has helped create considerable progress in computer Go is a Monte-Carlo. Monte-Carlo is a move evaluation technique that uses repeated random sampling of a large number of games which can be guided by a UCT algorithm (Kocsis and Szepesvari, 2006). While Monte-Carlo techniques exist that allow programs to significantly improve, there are still too many moves in the beginning for this technique to be effective for opening strategy.

2 Opening Strategy

The opening in Go is considered to be the hardest part of the game, even for professional players, because a bad opening can set a player back so far that they can not catch up. Opening strategy in Go is particularly complex because of the large search space. On a 19x19 board there are a possible 361 opening moves. In 19x19 opening, the most reasonable opening moves are ones that put the player in a position to build territory. Therefore, a sensible opening is one that uses the edges of the board to build against by playing in the corners and on the third or fourth lines. Moves made in the center of the board are less helpful because it is more challenging to build territory in the center (Ishigure, 1995).

Using the Monte-Carlo techniques with the large search space for potential first moves caused our program Orego to place stones haphazardly. Figure 1 shows an example of Orego playing without an opening book. Even once stones have been placed on the board there are still many different reasonable options and many different board configurations possible. Orego would play what seem like random moves including in the center of the board. For that reason we attempted to improve the opening play by incorporating human knowledge in a database of opening moves extracted from professional games to determine strong moves.

3 Fuseki Book

Fuseki refers to the first moves of a game and patterns that surface out of those moves (Shotwell, 2003). To create a fuseki book, or a look-up table of board configurations, we read in professional games and stored move sequences from each game into a database. The professional games were collected from databases found online; a list of the websites is contained on the reference page. Each game is saved in the Smart

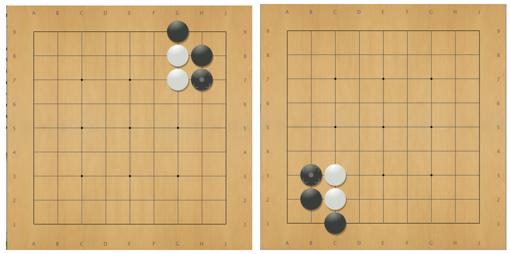

Figure 2: An example of a board that has been rotated.

Game Format (SGF) for computer-recorded Go games. We used over 4,600 games to produce our opening book database.

As we read in progressions of moves from the database, the response move to a particular board configuration was saved as well as the frequency of that response move. We stored only those moves that had appeared in at least 10 games, and we used the first 50 moves in each game. Once the fuseki book is called by Orego, the current board state is looked for in the database. If it is there, Orego extracts the response move and sends the information back to be used as the next move. The fuseki book is exhausted if the board state is not contained in the database, or if the number of moves has exceeded the depth of the fuseki book. The depth can be set by a command line argument or is a default value of 50 moves.

3.1 Reflections and Rotations

To improve the fuseki book, we accounted for different board configurations that could be produced by simply reflecting and rotating the given board state. The move sequences were reflected and rotated to produce eight distinct sequences for each game. The reflections and rotations allow Orego to act as if the database contained these additional games in our collection. Consequently Orego can respond to more situations. Figure 2 shows one board that has been rotated to produce two different board configurations.

3.2 Search Tree

The first implementation of the fuseki book incorporated a search tree. The first several moves in the collection of professional games were saved into a tree, the top level representing the first set of moves and the children of each move were the moves that immediately followed. When Orego chose a move to play, if the tree contained the current played sequence of moves, Orego would select the next move that had occurred most often in the database. If the program had exhausted the moves available in the tree, it would exit the opening book and commence normal play.

Although the resulting tree provided a reasonable fuseki book for Orego, it was slower than desired. The slow speed was due in part to the vast size of the tree since it included moves of thousands of games and requires Orego to traverse the tree each time a move needed to be extracted.

3.3 Transposition Table

Transposition tables are a good alternative to search trees to speed up the search for information. In this table, a board state is linked to the most common response move. When extracting moves from the professional games, we play each move on a board and save that board's condition. If the condition has been previously saved, we only need to track the response and a counter representing the frequency of that response move. One advantage of this structure is that in storing the current board state, the table prevents the storing of redundant information by accounting for the transposition of moves.

Different sequences of moves can result in the same board arrangement. For example if we have the move sequence A, B, C and the sequence C, B, A we end up with the same board state and that state is only saved once into the database. Figure 3 shows two boards with the same configuration from different move sequences. Furthermore, a board state reached by a different sequence of moves than those in the professional games will still be given a good response while Orego would have had to exit the book with such a sequence in the search tree.

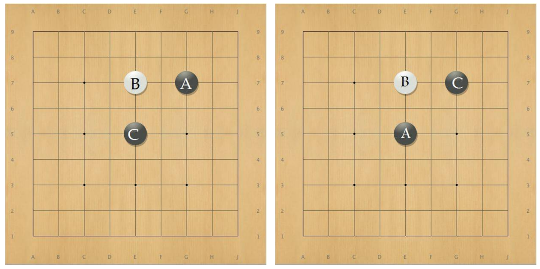

Figure 3: An example of a transposition of moves.

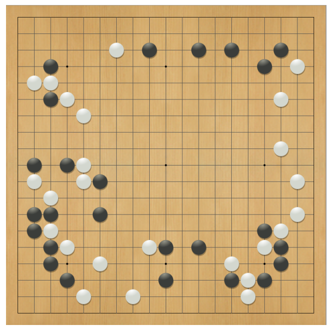

Figure 4: Orego with fuseki book, 50 moves into a game.

The transposition table also gives the added benefit of increased speed. By using a transposition table Orego needs only to submit the current board state to get the next move.

When Orego uses the fuseki book it is clear that there is less action in the center of the board, and more development on the edges and corners. Figure 4 shows an example of Orego using the fuseki book.

4 Joseki Book

The patterns that occur repeatedly during opening play, especially in the corners of the board, are called joseki (Kosugi and Davies, 1973). Josekis are generally sequences of moves that result in an equitable trade off of territory and influence for both players (Shotwell, 2003). There are many variations of joseki sequences and successful professional players will tend to rely more on feeling for good moves rather than memorized shapes of joseki patterns (Kosugi and Davies, 1973).

In order to enhance Orego's opening strategy, we wanted to expand the opening book to include joseki information. Incorporating these common sequences into our opening book would allow Orego to achieve better play in the corners and optimistically a better standing going into the middle stage of the game.

Joseki sequences were extracted from professional games similar to the way the fuseki database was generated, with board states linked to response moves saved into a database. However, with the joseki database we wanted to only consider those moves that were in the corners. When reading in the games the joseki book generator ignores the rest of the board, and stores the board state of only one corner at a time into a transposition table.

Reflections and rotations of the board configurations were also produced in the same way as in generating the fuseki book. To extract information from the joseki book, Orego submits the current board state and the joseki book considers only those moves played in one corner. The joseki book then checks for this situation to be contained in the transposition table. If it is a response move is generated. If not, the joseki book repeats the process for the next corner. The joseki book is exhausted once the total number of moves played exceeds the depth of the joseki book.

Figure 5 shows an example of Orego using the joseki book. Orego has some joseki-like sequences in the corner and edges, and still plays less in the center. This helps to establish a stronger position entering into the middle stage of the game.

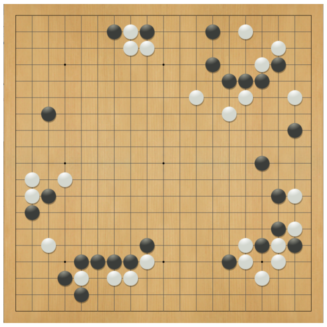

Figure 5: Orego with joseki book, 50 moves into a game.

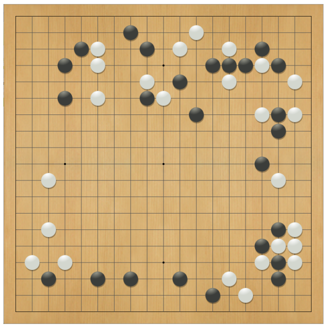

Figure 6: Orego with the combination book, 50 moves into a game.

5 Fuseki and Joseki Combination Book

Combining the fuseki and joseki information, Orego can respond to more situations in during opening play. In addition, more human knowledge is available to be used to build opening strategy. The combination book uses the fuseki and joseki book to extract information. The combination book first checks the fuseki book for a response move to the current board state. If one is produced, the move is passed on to Orego to play. If the board state is not contained in the fuseki book, the combination book checks the joseki book for a response move.

Figure 6 shows Orego with the combination of the fuseki and joseki book. The combination book visually looks to be making reasonable moves that would help Orego establish a strong basis.

6 Experimental Results

To study the effectiveness of all three books, experiments were run for four conditions: Orego with no opening book, with the fuseki book, with the joseki book, and with the combination book. In each condition, Orego played 560 games (280 as black, 280 as white) against GNU Go 3.7.11. All games were 19x19 with Chinese scoring and 7.5 points komi.

The experiments were run on an iMac with a 2 GHz Intel Core 2 Duo processor, 2 processor cores, running Mac OS 10.5.8. Orego was run using Java 1.6.0, using the command-line options –ea (enable assertion), -server (server mode, turning on the just-in-time compiler), and –Xmx1024 (allocating extra memory). In all conditions, Orego was given two threads, opening depth of 50 moves, and 16,000 playouts per move. Playouts per move is mutually exclusive with giving a set number of seconds per move.

Orego without an opening book won 201 of 560 games (35.9%). Orego with fuseki book won 213 games (38.0%). Orego with joseki book won 190 games (33.9%). Orego with the combination book won 159 games (28.4%). Although the fuseki book shows a higher win percentage, the results are not quite statistically significant (pooled sample proportion one tailed z-test, $p < 0.1$).

7 Discussion and Future Work

With the fuseki book, Orego does make a stronger opening by playing in the corners and around the edges. Although Orego's opening play is visually better with a fuseki book, it may be possible that Monte-Carlo is not able to exploit the benefit of a better opening. It is also possible that the fuseki book is exhausted early and therefore the full benefits of human knowledge are not used.

The joseki book did not result in a win percentage increase. This could be due to the fact that making a joseki move is not as good as making a move that seems good for the exact current situation despite Monte-Carlo's weak ability to evaluate.

It is somewhat surprising that the combination of the two books is the weakest of the four conditions. The combination book should have been able to utilize information from both books resulting in the most professional human knowledge, and the slowest to be exhausted. This seems to be a weaker option than allowing Orego to evaluate the board and decide a move.

This paper introduced ways to create a database from professional games and to use that database to generate opening moves. Although experimental results do not show that the opening books improve performance, the fuseki book does give us a stronger opening. In the future, perhaps the fuseki book should produce moves to be suggested to the move generator. Then that move generator could run the evaluation assessment on the board with extra emphasis on the suggested move. Using the opening book to make suggestions, rather than selecting a move, could be more beneficial because there are often more than one good response to any given board configuration. By using the suggestion with the evaluation function, we can weigh the different choices for response moves.

8 Acknowledgments

We would like to thank the Willamette Valley REU-RET, Lewis & Clark College, and the National Science Foundation for the opportunity to conduct this research. We would also like to thank our Orego team members for advice and support.

References

[1] Garlock, Chris. (2008). Computer Beats Pro at U.S. Go Congress. *American Go E-Journal*, Vol. 9, No. 40. http://www.usgo.org/EJournal/archive/20080807/20080807.htm

[2] Ishigure, I. (1995). *In the Beginning: The Opening in the Game of Go*. Kiseido Publishing: Tokyo, Japan. ISBN 4-906574-10-6.

[3] Kocsis, L. and Szepesvari, C. (2006). Bandit Based Monte-Carlo Planning. *Proceedings of the 17th European Conference on Machine Learning*, Sept. 18-22, pg. 282-293. http://www.sztaki.hu/~szcsaba/papers/ecml06.pdf.

[4] Kosugi, K. and Davies, J. (1973). *38 Basic Joseki*. Ishi Press: Tokyo, Japan. ISBN 4-87187-011-1.

[5] Shotwell, Peter. (2003). *GO! More Than a Game*. Tuttle Publishing: Boston, MA. ISBN 0-8048-3475-X.

SGF Game Databases:

[6] byheart go. http://byheartgo.googlepages.com/ (Accessed 9 July 2009).

[7] Go4Go. 2009. Professional Game Collection. http://www.go4go.net/ v2/modules/collection/ (Accessed 9 July 2009).

[8] GoBase. 2009. Special Game Collection: 9x9 games. http://gobase.org/9x9/ (Accessed 9 July 2009).

[9] Go Games. http://homepages.cwi.nl/~aeb/go/games/index.html (Accessed 9 July 2009).

[10] Hollosi, Arno and Pahle, Morten. 2009. Sensei's Library. http:// senseis.xmp.net/ (Accessed 22 July 2009).

[11] MidValley Hosting. 2009. Computer-go Mailing List. http://hosting.midvalleyhosting.com/mailman/listinfo/computer-go (Accessed 7 July 2009).

[12] Momoyama. 2003. Go News 2003. http://www.kyoto.zaq.ne.jp/ momoyama/news/news.html (Accessed 9 July 2009).

[13] Muller, Martin. 2009. CSU Alberta. Professional Games in Smart Go Format. http://www.cs.ualberta.ca/~mmueller/go/games.html (Accessed 9 July 2009).