Using Joseki for Tactics Deployment in Computer Go

Chung-Chih Li and Hikyoo Koh[1]

Beaumont, Texas, USA

ABSTRACT

We present a new idea for opening play to improve the playing strength of computer Go programs. It uses a collection of canonical sequences (Joseki) and their deviations with the aim to improve computer Go programs. Instead of trivially matching opening moves played to the collected sequences, we define a notion of similarity to extract the most suitable move from the candidate sequences. The simplicity of our method and its positive outcome make our approach a promising tool to be integrated into a complete computer Go program for a foreseeable improvement.

1. INTRODUCTION

To program a computer to play Go at an acceptable level is one of the most challenging tasks in Artificial Intelligence (Green, 1985). About 20 years before Kasparov's sensational defeat by the IBM Deep-Blue in 1997 (Hsu, 1999), Berliner (1978) stated: "Even if a full width search program were to become World Chess Champion, such an approach cannot possibly work for Go and this game may have to replace chess as the task par excellence for Artificial Intelligence". A straightforward analysis will convince AI researchers that the traditional search-based algorithm does not work for computer Go. Many different approaches have been suggested in the past decade, e.g., neural network approach, cognitive model, machine learning, fuzzy logic, and so on. We refer interested readers to (Müller, 1998; Bouzy and Cazenave, 2001) for more adequate surveys.

In this note we focus on pattern recognition. The idea can be traced back as early as 1970 (Zobrist, 1970), while GOGOL (Cazenave, 1996) is a more recent implementation using pattern recognition. The basic idea is to match the current game to some predefined game patterns and make a move accordingly. For this purpose we developed a Canonical Sequence Directed Tactics Analyzer (henceforth CSDTA). It uses canonical sequences as predefined patterns. The canonical sequence is also known as Joseki in Japanese, literally it means Fixed-Stone, which are sequences of local optimal moves with an influence that may extend to remote territory. We admit that our approach is not new to human players. A serious Go player should carefully study canonical sequences and a skillful player will easily remember hundreds of them. Each canonical sequence has been thoroughly evaluated by expert players for years, or even centuries. They are considered optimal plays for both sides. If one player plays out of the sequence by a mistake or for other trade-off reasons, the opponent will gain some significant benefit if he/she can response properly.

2. CSDTA AND ITS EVALUATION MATRICES

The underlying idea of CSDTA is that it (1) recognizes the configuration on the playing board and (2) consults the collective experience of expert Go players in form of canonical sequences to decide its next move through some evaluation matrices. Our database contains 1,278 canonical sequences and their deviations based on the collections in Ishida (1984) and Ha (1988). The deviations are beneficial to the player who plays the black stones if the opponent plays out of the canonical sequences. Traditionally, canonical

[1]Dept. of Computer Science, Lamar University, Beaumont, Texas 77710, USA. E-mail: {licc, koh}@hal.lamar.edu

sequences are classified into different categories according to their first moves. Each category carries a different agenda for further deployment. We continue to classify each category in our database according to their second moves as shown in Table 1. The second move of a canonical sequence determines how the opponent engages. Since we can easily rotate and symmetrically transform any canonical sequence and switch its stone colours, we store only one sequence for all its trivially mirroring variations. The black stone always makes the first move in our collection.

At this moment, CSDTA does not consider the strategy behind each canonical sequence. Thus, it is straightforward for CSDTA to determine the next move if the current game matches some canonical sequences. In contrast, if CSDTA cannot find an exact match, some similar canonical sequences will be selected for further investigations and weighting. There are two stages in CSDTA. The first stage defines similar canonical sequences and the second stage decides on the next move. Each stage uses a different evaluation matrix called *importance matrix* shown in Figure 1, where M_1 and M_2 denote the matrices for the first and second stages, respectively. One can obtain a complete matrix by rotating the quadrant to the rest three quadrants.

	1^{st} moves				(Total:1278)		
2^{nd} moves	3-3	3-4	4-4	5-3	5-4	6-3	6-4
3-3	-	1	3	33	5	1	-
3-4	-	-	2	193	117	1	2
3-6	-	-	298	-	-	-	-
4-4	10	-	-	-	-	-	-
4-5	2	-	-	53	-	-	-
4-6	4	1	6	-	-	-	-
5-3	-	267	-	-	-	-	-
5-4	-	182	-	-	-	-	-
6-3	-	25	-	-	-	-	-
6-4	3	36	-	-	-	-	-
Others	1	2	30	-	-	-	-
Total	20	514	339	279	122	2	2

Table 1: Collection of Canonical Sequences and Deviations.

Stage One – Selection of Similar Canonical Sequences. We first define the difference degree of each canonical sequence with respect to the current game. Following the philosophy that we should pay much more attention to the opponent's last move, we put the focus on the last move in stage one. For each canonical sequence in the database, we extract the initial segment of the sequence up to the focus (the last move of the game). Note that the number of moves in the initial segment of a concerned canonical sequence may not be the same as the current game. For convenience, let Ω denote the current game, $\star$ the focus, and ω the initial segment of a concerned canonical sequence up to $\star$. Consider the game played by CSDTA against a human player (white stone) in Figure 2. We shall explain how CSDTA enters 9_B in response to 8_W. We have

$$\Omega = \{(G,4)_B, (H,6)_W, (D,3)_B, (F,6)_W, (E,5)_B, (H,4)_W, (H,3)_B, (I,4)_W\}.$$

Here we consider the evaluation of canonical sequences (A) in Figure 3. Since $\star = (I,4)$, it follows that

$$\omega = \{(G,4)_B, (H,6)_W, (D,3)_B, (H,3)_W, (H,4)_B, (I,4)_W\}.$$

The calculation of the difference degree is rather straightforward. Find the mismatch points between Ω and ω and sum up the values of M_1 at the mismatch points with focus $\star$ landed at $(I,4)$. In this example, the mismatch points are $(F,6)_W, (E,5)_B, (H,3)_W$, and $(H,4)_B$. Thus, the corresponding values are $2, 3, 7$, and 8, and therefore the degree of difference is 20. Similarly, one can find the degrees of difference for canonical sequence (B), (C), and (D) in Figure 3, which are 22, 26, and 26, respectively. All shaded stones with numbers in *italics* in Figure 3 are moves after $(I,4)$, the focus, which are not yet played and thus will not be considered.

The difference degree of every sequence will be computed. Let d be the minimum difference degree. All canonical sequences with degree of difference less than or equal to $d \times (4/3)$ are called *similar canonical sequences* with respect to the current game, and they will be passed to stage two for further evaluation. The ratio, 4/3, is an experimental value that can provide better results based on our collection. The canonical sequence (A) in Figure 3 is the one with the minimum degree of difference, which is 20. Therefore, all canonical sequences with degree of difference less than or equal to 26 will be further evaluated in the next stage.

Stage Two – Selection of the Next Move. Since the vicinity of a potential move in the current board is critical in determining the goodness of the move, it is natural to put our focus on the move for evaluation, while the opponent's last move is equally important to us. Therefore, CSDTA uses two focuses in stage two: the last move of the game ($\star$) and the next move suggested by each similar canonical sequence. Let $\diamondsuit$

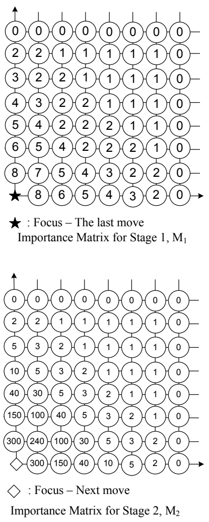

Figure 1:
Importance matrices.

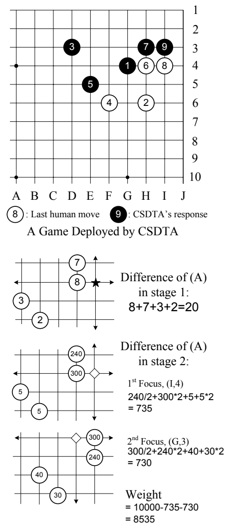

Figure 2:
An example game.

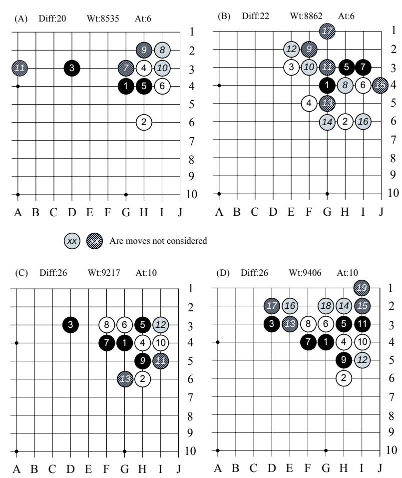

Figure 3: Four canonical sequences evaluated by CSDTA.

denote the latter one. It is clear that $\diamondsuit$, a move to be entered, is much more sensitive to its surrounding than the move that has been entered. Thus, we use a different importance matrix in which the weight reduced rapidly when the points moving away from $\diamondsuit$ (see M_2 in Figure 1). Nevertheless, we also use M_2 for focus $\star$ in this stage. Note that, after stage one, similar canonical sequences have been selected. If we use an equally rapid weight-change matric in stage one, we will lose too many canonical sequences for further evaluation. The pattern of M_2 is similar to Sander-Davies's (1985) influence pattern and Yen and Hsu's (2001) Rough Inference.

Moreover, we believe that the type of mismatch at each point should be considered. We therefore adjust the weight according to the type of mismatch as shown in Table 2, where $f(w)$ is defined in (1) and w is the entry of M_2. The final weight of each selected canonical sequence is simply 10,000 subtracting the sum of weights in M_2 adjusted by (1). Figure 3 shows four similar canonical sequences and their final weights. Consider sequence (A) again. For focus one, the weights corresponding to the four mismatch points are 240, 300, 5, and 5. After adjustment, we have $240/2 + 300 \times 2 + 5 + 5 \times 2 = 735$. For focus two, we have 300, 240, 40, and 30, and after adjustment, we have $300/2 + 240 \times 2 + 40 + 30 \times 2 = 730$. Therefore, the final weight of canonical sequence (A) in Figure 3 is $10,000 - 735 - 730 = 8,535$. Similarly, the final weight of canonical sequence (D) is 9,406, which turns out to be the one with the highest weight among all similar canonical sequences. As (D) becomes the winner, its next move 11_B at (I,3) becomes CSDTA's next move 9_B. Note that in (1) we need to find the number of liberties of a cluster. Thus, CSDTA needs to recognize every cluster on the board, which is the only unit higher than a single stone to be recognized by CSDTA. Each cluster is maintained by two linked lists for connected stones in the same colour and their liberties, respectively.

Canonical Sequence	Board (Current Game)		
	Empty	White	Black
Empty	0	$2w$	$f(w)$
White	$w/2$	0	$w/2$
Black	w	$2w$	0

$$f(w) = \begin{cases} 2w & \text{if the black stone reduces the liberties of the cluster;} \\ w & \text{otherwise.} \end{cases} \tag{1}$$

Table 2: Weight Adjustment in Stage Two.

3. PRELIMINARY ASSESSMENT

Given that CSDTA is not designed play a full game of Go and no other assistant evaluation is used, we refrain from making definitive statements on the method proposed. However, we are prepared give a preliminary assessment on CSDTA in this section. There are some obvious restrictions: (1) It plays only the black stone; (2) It considers only one quadrant of the board; (3) It cannot complete a game but plays only the first 10 to 15 moves in a quadrant; equivalently, this is roughly started from the deployment stage (opening-game) through the first half of contact-fighting stage (mid-game). Moreover, CSDTA does not consider the opening-game strategy and the strategies behind each canonical sequence. We simply let the user initiate the first move (Black) and the second move (White).

The Good. In general, CSDTA can respond with an excellent move if the selected similar canonical sequence has a degree of difference less than 20, and it can play fairly well when the degree is between 20 and 30. When the degree is between 30 and 40, CSDTA still plays acceptably in most cases. Only when the degree of difference exceeds 40, the quality becomes unpredictable. In the game shown in Figure 3, the winner is canonical sequence (D) which has 26 as the degrees of difference; CSDTA suggests (I,3) as the ninth move of the game. In this example, a good shape for Black has successfully deployed. In most cases, if the minimum degree of difference is less than 30, we are likely to find sequences with weights higher than 9,000. This indicates that CSDTA is not likely to suggest a bad move in a normal open game. We observe that the minimum degrees of difference remains under 40 for about 15 moves. In a quadrant, 15 moves would have taken the game into the mid-game stage. Extending to a full board, CSDTA can play about 60 good moves. We consider to have 60 good moves without too much computational cost a significant achievement.

The Bad. As no other technique next to pattern recognition is used in CSDTA, it may be not surprising that CSDTA is able to prepare a good tactics deployment for a battle, but that it cannot finish the battle and win a territory. Another major shortcoming is that it cannot handle some obvious abnormal moves. Yet, we do not consider both as real disadvantages, because CSDTA is not supposed to handle these situations. It should be easy to use the difference degree and weight to set a cutting line for CSDTA to quit and let another method to take on. In fact, many situations in which CSDTA handles badly by their nature can be solved by using other approaches, e.g., using search-based algorithms or another kind of patterns. (See the discussion in the next section.)

The Cost. Memory requirement and response time of CSDTA is not an issue with today's hardware standard. Our program uses 150 KB to load all 1278 sequences into main memory. Also, the response time for each move is negligible. Moreover, the memory can be released if the game enters the mid-game stage where canonical sequences are no longer useful.

4. FUTURE STUDY

There are four directions for future development. We briefly discuss them in the following paragraphs.

Overall Game Planning. For a human player, a canonical sequence is selected according to his/her overall deployment strategy. In other words, each canonical sequence has a strategic aim. Thus, it is more appropriate if the program can select canonical sequences according to its overall plan. One possible way is to associate a strategy to each canonical sequence in our database. Such strategy can be something like "this sequence attaches importance to the corner territory" or "this sequence trades the corner territory for influence in the centre". That information will become a built-in strategy for each canonical sequence. Once the goal is determined, we should weight the canonical sequences according to the goal.

Critical Patterns. The knowledge base of CSDTA contains only canonical sequences (Joseki). If the minimum degree of difference becomes large, CSDTA will still try to extract a similar part from *not-so-similar* canonical sequences to determine the next move. As a result, the quality becomes unpredictable. When the game varies from the standard sequence, it is more appropriate to consult a different kind of pattern known as "Critical Patterns". In general, such patterns provide urgent moves (*Kyuba*), vital moves (*Kyusho*), and tactical moves (*Tesuji*), which are not sequences of moves but only a few critical points that the player needs to occupy whenever a certain pattern is recognized. In particular, critical patterns should suggest how to make two eyes, to kill an intervening enemy, to escape stones in danger, to connect with friendly groups, etc. Adding some critical patterns can certainly extend the number of acceptable moves significantly.

Refining Importance Matrices. Our importance matrices are naive in a sense that the degree of importance radiates uniformly from a single point (the focus). In a human player's conception, this should not be the case. The formation of friend and enemy stones strongly affects the pattern of the radiation. Thus, to improve our evaluation, we may adapt the idea of Yen and Hsu's (2001) Rough Inference in designing the importance-matrices according to the shape on the playing board. Also, instead of using the same matrix, we should design two different matrices for the two focuses examined in stage two. Moreover, a different goal planned may direct CSDTA to select a different matrix.

Look-ahead Technique. Another direction for a promising improvement is to adapt a look-ahead program in CSDTA in a few places. First of all, we can replace our adjustment function f (see Table 2) used in stage two, which is obviously inadequate, by an actual search-based/look-ahead program. For example, MIGOS (MIni GO Solver) presented in Van der Werf, Van den Herik, and Uiterwijk (2003) is such a program using the well-known α-β tree to evaluate a 5×5 board. CSDTA can use MIGOS to examine the 5×5 vicinity around the focuses. Also, when the minimum degree of difference is higher than a constant (for example, 40), CSDTA should halt and use a look-ahead program to finish the battle or to occupy actually some territory. In this way, abnormal moves can be handled too.

5. CONCLUSION

Our proposed method is not meant for building a complete Go program. Instead, our goal is to provide a feasible approach in helping improve the opening play by computer Go programs by using canonical sequences. The notion of similar canonical sequences and weighting matrices make the use of canonical sequences more flexible than just recognizing predefined patterns. So, it is worthwhile to obtain 10 excellent moves for tactical deployment in a quadrant at little cost of computing time. If CSDTA cooperates with an overall strategic analyzer, a stage determining program, a focus shifter, and a global and local evaluator, it may be assumed to play more than 40 excellent moves on a whole board in the full game, and this is about the same outcome a human player expects to achieve by way of studying canonical sequences. We believe that CSDTA has a future.

6. REFERENCES

Berliner, H. J. (1978). A Chronology of Computer Chess and its Literature. *Artificial Intelligence*, No. 10, pp. 201–214.

Bouzy, B. and Cazenave, T. (2001). Computer Go: An AI oriented survey. *Artificial Intelligence*, Vol. 132, No. 1, pp. 39–103. citeseer.ist.psu.edu/bouzy01computer.html.

Cazenave, T. (1996). Automatic Acquisition of Tactical Go Rules. *Proceedings of the 3rd Game Programming Workshop* (ed. H. Matsubara), pp. 10–19, Hakone, Japan.

Green, H. S. (1985). Go and Artificial Intelligence. *Computer Game Playing: Theory and Practice*, pp. 141–51.

Ha, C. (1988). *Fundamentals of Protocols in Playing Go Games*, Vol. I and II. The Korean Go Association, Seoul, Korea.

Hsu, F. (1999). IBM's Deep Blue Chess Grandmaster Chips. *IEEE Micro*, Vol. 19, No. 2, pp. 70–81.

Ishida, M. (1984). *Canonical Sequence Dictionary*. World Culture Publishing, Taipei, Taiwan.

Müller, M. (1998). Computer Go: A Research Agenda. *Lecture Notes in Computer Science*, number 1558, pp. 252–264. Springer-Verlag, Heidelberg, Germany.

Sander, P. T. and Davies., D. J. M. (1985). A Strategic Approach to the Game of Go. *In Computer Game Playing: Theory and Practice*, pp. 167–76. Ellis Horword, New York.

Werf, E. van der, Herik, H.J. van den, and Uiterwijk, J.W.H.M. (2003). Solving Go on Small Boards. *ICGA Journal*, Vol. 26, No. 2, pp. 92–103.

Yen, S. and Hsu, S. (2001). A positional Judgment System for Computer Go. *Advances in Computer Games (ACG9 Proceedings)* (eds. H.J. van den Herik and B. Monien), pp. 313–326. IKAT, Universiteit Maastricht, Maastricht, The Netherlands.

Zobrist, A. L. (1970). Feature Extraction and Representation for Pattern Recognition and the Game of Go. Technical report, University of Wisconsin. Ph.D. dissertation.