Using master games statistics on the squares played

Keigo Matsubara[1], Tsuyoshi Hashimoto[2], Hiroyuki Iida[1,3]
[1] Department of Computer Science, Shizuoka University
[2] Faculty of Engineering, Shizuoka University
[3] PRESTO, Japan Sicence and Technology Agency
E-mail: {cs0086,hasimoto,iida}@cs.inf.shizuoka.ac.jp

Abstract

Most state-of-the-art Shogi programs adopt variants of the fixed-depth alpha-beta search algorithm. However, there are numerous search extensions that explore "interesting" lines of play more deeply. Most search extensions make the decision to extend a certain line of play dynamically. We propose a static topological frequency based search extension. That is, the decision to extend searching of a certain move is dependant on the frequency the relevant square was played in master games. As this method was inspired by and akin to Realization Probability Search[1] we named it Topological Frequency Bias (TFB). The proposed TFB has been incorporated to our computer Shogi program, TACOS, and two experiments proved its effectiveness.

1 Introduction

The alpha-beta algorithm is the most popular method for searching game-tree in adversary board game as Shogi. Over the years, this algorithm has been improved with various methods and more efficient variants have been introduced. The most basic algorithm searches to some fixed depth. However pure fixed depth is rarely used in game-play programs. In most of the state-of-the-art Shogi programs, some promising move sequences are explored beyond the fixed depth limit. In TACOS, our computer Shogi program, we first adopted Realization Probability Search[1]. However, infrequent mistakes introduced by the algorithm have lead us to look for alternatives.

As these facts show, search extension of promising moves is a very effective method. Many related ideas of it have been proposed and most of them can be categorized as dynamic methods. That is, the extension is dependant only on information gathered during the course of the game.

Essentially, it is considered that each game has a statistical bias. Namely, best moves occupy certain squares more frequently than others. It is hypothesized that using the domain's statistical bias may allow for more effective search implementations. These ideas in contrast to dynamic search extensions, make use of precalculated static criteria.

This paper is organized as follows; Section 2 surveys related work. Section 3 presents our method of obtaining the statistical information from game records of professional Shogi players. Section 4 describes the implementation and integration of the calculated statistics to our computer Shogi program, Tacos. Section 5 presents the results of 2 experiments supporting the effectiveness of TFB. Finally, Section 6 summarizes and suggests directions for future work.

2 Related Work

Most computer Shogi programs explore promising moves more deeply and numerous ideas were proposed.

IS Shogi explores sequential patterns of good moves (i.e. "tesuji" in Japanese) more deeply [4]. Such a pattern (consisting of a sequence of moves) is then considered as a move.

Kanazawa Shogi extends 2 ply at the maximum on recaptures. In addition, when a move is considered the best response to an opponent's threat (i.e. check or attacking a castle formation) an extension of 1 ply is assigned.

YSS adopts "half extension algorithm"[3]. This method allows an extension according to the quality of the moves leading from the root to the current position. In particular, it adds 0.5 ply of search extension for any best move (as was decided by iterative deepening) found in the variation leading to the current move.

GEKISASHI adopts Realization-Probability Search[1] that expands nodes as long as the realization probability is greater than a given threshold. Therefore, it spends little computational resource on unrealistic moves. TACOS developed by us also adopt it.

3 Search Extention Using Statistics

Essentially, it is considered that each game has a statistical bias. Namely best moves occupy certain squares more frequently than others. Figure 2 shows the example of Shogi. This example shows the number of Silver's moves in the endgame. In this figure, own king is on lower left and opponent's king is on upper right and both rooks are in the opponent's camp. It is created from about 10000 game records of the professional Shogi players. It shows Silver is dropped around the opponent's king. This fact describes that Silver is often used to attack castle formations. We expect that, by using this kind of statistical bias, game-tree search effectiveness can be improved.

In practice, for every piece and every square in a certain situation we register how often that piece was played onto that specific square in professional games. Then, the length of the search extension is determined by the precalculated frequency. Since the statistical information is prepared in advance, the overhead in real-time of using it is slight. This in turn, allows the program to find good moves more effectively.

9	8	7	6	5	4	3	2	1	
		9	10	18	9	2	2	4	一
		1	3	4	1		1		二
		7	6	8	7	3		1	三
	5	3	10	2	1	1		1	四
	1	2	1	6				4	五
1			1	3		1		4	六
1	2	3	1		2	1			七
	1	3						1	八
		3							九

Figure 1: The number of Silver's moves in endgame of Shogi(extracted from 10000 professional Shogi players' game records)

4 Application for Computer Shogi

This section describes how to apply TFB to a computer Shogi program. We implement it in TACOS, which is a computer Shogi program developed.

4.1 Creation of Statistics

This statistics is extracted from about 10000 game records of some professional Shogi players. Game positions are classified into 24 kinds by the items of Table 1. In addition, we use symmetry between first player's (Japanese "sente") and second player's (Japanese "gote") moves as well as left right symmetry of the board. These symmetries improve the bias of the frequencies.

The position of kings is referred to because Shogi is a game aiming at making kings checkmated. It is clear that the position of kings introduces a topological deviation of a position's frequencies.

In this research, after performing the above-mentioned squares conversion, it classifies into two

Table 1: Category of Statistics

King	Rook	Stage of Game
Vertical	both own camp	opening game
Slanting	both opponent's camp	middle game
	others	early-endgame
		late-endgame

kinds, "Vertical" and "Slanting". "Vertical" is the case where both kings are on the left side. "Slanting" is the case where Black's king is on the left side of the board, and White's king is on the right side of board.

We have conceived of the classification by the rook to be effective, because we think it indicates the principal attacking zone. In this research, the position of a Rook was made into two kinds: one case when a Rook is in the own camp, and another case when the Rook is either in the opponent's camp or in one's hand(Japanese "mochigoma"). Then, Rook dependant position's classification was made into three kinds: "both in own camp" , "both in opponent's camp" and "others".

The third classification is according to 4 phases of stage of a game. In this implementation we have used: "opening", "middlegame", "early-endgame", and "late-endgame". The importance of a piece changes with the stage of a game. The value of a piece is important at opening and middlegame. However, the speed of approaching the king is more important than piece value at end-game. Therefore, we thought that the classification by the stage of a game is also required.

4.2 Implementation

The length of search extension is determined by statistics made by the method shown in section 4.1. However, we do not apply this method in opening game, because we often play standard sequences in opening game. Therefore we consider it is irrelevant to use statistics.

The proposed method has been implemented in our Shogi program called TACOS. Since we apply Realization Probability Search[1] to TACOS, we increase realization probability to search more deeply.

First, based on statistics, we calculated the probability P that a piece moved to the square. The probability P is calculated as follows.

$$P = \frac{D_{i,j}}{D}$$

$D_{i,j}$ means the number of the moves on i rank and j file in the present situation. D represents the total of moves in it. Furthermore, P is multiplied by the suitable weight W. Therefore, modified Realization Probability is calculated as follows.

$$RP' = RP * (1 + P * W)$$

RP means the Realization Probability by the conventional method. RP' is the modified Realization Probability by our proposed method. When this formula is applied, the larger the frequency the more deeply we search.

5 Experiment

In order to evaluate our method we prepared two different versions of TACOS. The first, TACOS-RPS, utilizes the original version of RPS. The second program, TACOS-TFB, is using our modified version; namely, TFB.

5.1 Solving Problem Test

Matsubara and Iida(1998) provided a standard test suite for computer Shogi programs. This test suite consists of 48(mainly middle and endgame) positions extracted from game records played by professional Shogi players. After implementing TFB, we experimented with 30 different instants of the W parameter and in order to find the most appropriate parameter setting for TACOS-TFB. Table 2 shows the results. The number of correct answers increased by two questions relatively to the base version TACOS-RPS.

Table 2: Performance in Solving Problem Test

Program	Correct
TACOS-RPS	19
TACOS-TFB	21

5.2 Self-Play Experiment

In addition to the test suite experiments we have performed a self-play experiment using TACOS-RPS and TACOS-TFB. 314 matches were played using same time controls (both programs were given 10 seconds for each move). To prevent the programs from playing the same game repeatedly, 157 opening positions were used. The programs played each opening twice; once as the first player to move, and once as the second player to move. Table 3 shows the results of the match. TACOS-TFB shows slight improvement over TACOS-RPS.

Table 3: Win percentage of TACOS-TFB against TACOS-RPS

wins	draws	losses	win percentage
174	139	1	0. 554

6 Conclusion and Future Work

In this paper, we proposed a modified variation of Realization Probability Search[1]. Our method, TFB, is based on topological frequency of master level moves. Experimental results show our proposed method achieves effective search with a suitable parameters' set up. Using precalculated statistics, promising moves are explored more deeply.

We believe further improvements of some aspects of our current implementation can enhance the overall effectiveness of this method. Among those are the formulas of search extension, the classification method and the parameters tuning. The formula we used for search extension was simplified in order to save calculation time. It may be better to calculate more effective values if calculation resources permit. About a setup of a suitable parameter, we experimented using two kinds of parameters(drop or move). In addition, it is considered that parameters are likely to change in the stage of a game. There is still scope for improvement through further classification of statistics. Lastly, we think that using the obtained statistics to pruning bad moves is a promising direction.

Acknowledgments

The research was supported by the Grant-in-Aid for Exploratory Research of the Japan Society for the Promotion of Science (#2267).

References

[1] Tsuruoka,Y: "Game-Tree Search Algorithm Based on the Realization Probability", ICGA Journal Vol.25 No.3, pp.145-152, 2002

[2] Matsubara, H., Iida, H.: Evaluation of Computer Shogi by Next-Move Test(no.1), In Matsubara, H., editor, Computer Shogi Progress 2, pp.61-111, Kyoritsu Shuppan Co, 1998.(in Japanese)

[3] Yamashita, H.: About its Datastructures and Algorithm, In Matsubara, H., editor, Computer Shogi Progress 2, pp.112-142, Kyoritsu Shuppan Co, 1998.(in Japanese)

[4] Tanase, Y.: The algorithm of IS Shogi, In Matsubara, H., editor, Computer Shogi Progress 3, pp.1-14, Kyoritsu Shuppan Co, 2000.(in Japanese)

[5] Kanazawa, S.: The algorithm of Kanazawa Shogi, In Matsubara, H., editor, Computer Shogi Progress 3, pp.15-26, Kyoritsu Shuppan Co, 2000.(in Japanese)