Plausible Move Generation Using Move Merit Analysis in Shogi

Reijer Grimbergen
Electrotechnical Laboratory
1-1-4 Umezono, Tsukuba-shi, Ibaraki-ken, Japan 305-8568
E-mail: grimberg@etl.go.jp

Hitoshi Matsubara
Future University-Hakodate and PRESTO
116-2 Kameda-nakanocho, Hakodate-shi, Hokkaido, Japan 041-8655
E-mail: matsubar@fun.ac.jp

Abstract

In games where the number of legal moves is too high, it is not possible to do full-width search to a depth sufficient for good play. Plausible move generation (PMG) is an important search alternative in such domains. In this paper we propose a method for plausible move generation in shogi. During move generation, Move Merit Analysis (MMA) assigns a value to each move based on the plausible move generator(s) that generated the move. PMG with MMA on average reduces the number of moves to 54% of the total number of legal moves with 99% accuracy. Tests show that PMG with MMA outperforms full-width search in shogi.

1 Introduction

Full-width search has been very successful in two-player complete information games. Deep Blue in chess [17], Chinook in checkers [16] and Logistello in Othello [4] are examples of well-tuned full-width search programs that perform at the level of the human world champions.

In full-width search all legal moves in any given game position are searched. Based on domain-dependent heuristics, selectivity is added: some moves will be searched deeper than other moves. Examples of methods to add selectivity to the search are *quiescence search* [2], *singular extensions* [1] and *futility pruning* [7].

Full-width search has not always been the main approach. *Plausible Move Generation* (PMG) was very important in the early days of chess research. A plausible move generator would select a small number of moves using domain-specific knowledge [14, 3, 6]. The remaining candidates were then searched as deep as possible with alpha-beta search. For example, Bernstein's chess program [3] generated only 7 plausible moves in any position. Plausible move generation is the ultimate form of selectivity: discarding moves without any search. In chess, the risk of discarding a good search candidate was too high and full-width search has been the dominant approach since the Chess 4.5 program in the early seventies [18].

However, there are games in which it is impossible with current technology to search deep enough with standard full-width search to get a high performance program. Examples are games with a large average number of legal moves like *Go* and *shogi* [13] and single agent search problems such as *sokoban* [9]. Here plausible move generation can be a good alternative to full-width search.

Plausible move generation is also interesting for cognitive science. Despite the success of the full-width search approach, there has been debate about the level of artificial intelligence used in these programs (a discussion on this topic can be found in [11]). Cognitive science research has shown that human experts are capable of narrowing the search candidates to a very small number of relevant candidate moves [5]. Full-width search, looking at all possible moves in any game position, cannot be considered a model of human problem solving in game playing situations. By looking at fewer moves in any given game situation, plausible move generation comes closer to the simulation of human problem solving. Therefore, plausible move generation might give us insights in the way humans play games.

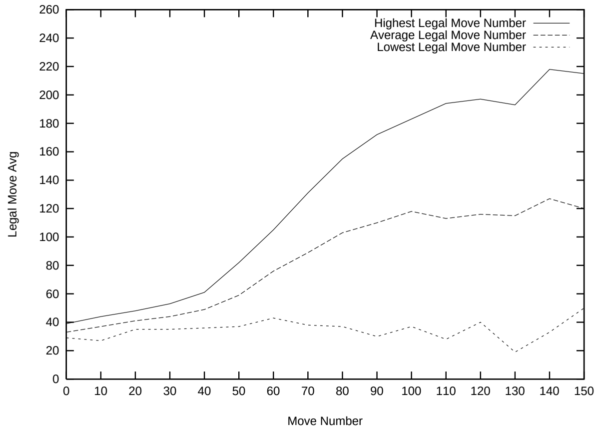

Figure 1: Highest number of legal moves, average number of legal moves and lowest number of legal moves by move number in 100 test games.

In this paper we propose a method of plausible move generation for shogi. Shogi is a game where plausible move generation is a good alternative to full-width search because of the large average number of legal moves. A set of plausible move generators for shogi will be defined. We will explain how analysing the merit of a move using these plausible move generators can improve move ordering and give additional cuts in the number of candidate moves. We will show that plausible move generation with move merit analysis outperforms full-width search in shogi.

2 Plausible Move Generation in Shogi

2.1 Why is Plausible Move Generation Necessary in Shogi?

The main difference between chess and shogi is the possibility of re-using pieces. A piece captured from the opponent becomes a *piece in hand* and at any move a player can *drop* a piece he captured earlier on a vacant square instead of moving a piece on the board. As a result of these drop moves, the number of legal moves in shogi is on average much larger than in chess. The average branching factor of the search tree in chess is about 35, while in shogi the average branching factor is about 80 [12].

In shogi the average branching factor does not tell the whole story. In chess the branching factor rapidly decreases towards the endgame and finally gets to a point where the exact theoretical game value can be retrieved from endgame databases [19]. This is not the case in shogi, where the branching factor of the search tree increases as the game progresses. To illustrate this behaviour, we have analysed the number of legal moves in 100 expert shogi games. The games have been selected to give a good coverage of the different types of positions that occur in shogi. The games therefore involve many different expert players (112) and have many different opening strategies (15).

The number of legal moves in the test games is given in Figure 1. This figure shows that the average branching factor of shogi tends to increase as the game progresses. As more pieces get captured, the number of possible drop moves increases, leading to an average branching factor higher than 100 in the endgame. The top line in Figure 1 shows that peaks of more than 200 can also be expected. The result of a shogi game is often decided in the endgame, so being able to deal with such a high branching factor can mean the difference between winning and losing.

It is not only the high branching factor that is a problem for building a strong shogi program. There is also the problem of strict time constraints. In shogi, the available time for a game under tournament conditions is much less than in chess. There are two reasons for this. First, the average game length of shogi is about 115 ply [8], while the average game length of chess is about 80 ply. Therefore, even under the same tournament conditions, a shogi program will have 30% less available time per move. Second, the tournament conditions for shogi programs are much stricter than in chess. In the annual CSA tournament, the unofficial computer shogi world championships, the available time per game is only

20 minutes. Therefore, even with the help of an opening book, on average only about 30 seconds per move are available for search.

To deal with large search trees under strict time constraints it is necessary to make good decisions about which moves to spend search time on. Plausible move generation is a method to make such a decision as it determines the moves that should be inspected further.

2.2 A Set of Plausible Move Generators for Shogi

As the general game-playing system METAGAMER [15] shows, a wide range of games have the notion of goal, threat and positional improvement in common. The goal of a game can be to either win material (e.g. chess, checkers), to occupy the largest territory (e.g. Go, Othello), or to reach a certain board configuration (e.g. five in a row, sokoban). Some goals are more important than other goals. For example, in chess mating (winning the king) is more important than winning a queen, which in turn is more important than winning a pawn. A threat is a move that, if not defended against, will reach a goal on the next move or after a forced sequence of moves. Finally, there are moves to improve the player's position without actually threatening to reach a goal. An example is too improve the mobility of a piece. It is also possible to defend against this type of move by playing a move that makes such a positional improvement impossible. If the move that improves the position is played, the other player would reach a goal. An example in chess is pinning a piece.

For each of these move classes a plausible move generator can be build which generates the moves in this class:

1. **PMG-Goal**:
   Moves that reach a goal.
2. **PMG-Th**:
   Moves that threaten to reach a goal.
3. **PMG-DefTh**:
   Moves that defend against a threat.
4. **PMG-PIm**:
   Moves that improve the position.
5. **PMG-DefPIm**:
   Moves that make it impossible for the opponent to improve its position.

For each game in which this basic set of plausible move generators is used, the PMGs have to be refined to reflect the features of that specific game. In shogi, the goal of the game is different than in Go and *PMG-Goal* is therefore different as well. For shogi, we have refined the five basic PMGs as follows:

1. **PMG-Goal:**
   - *Capture material.*
   - *Promote piece.*
2. **PMG-Th:**
   - *Check.*
   - *Attack king.*
   - *Attack material.*
   - *Discovered attack.*
     Moving a blocking piece leads to check or to a material attack.
   - *Threaten promotion.*
3. **PMG-DefTh:**
   - *Defend against checks.*

- *Defend king.*
- *Defend material.*
- *Defend discovered attacks.*
- *Defend against promotion threat.*

4. **PMG-PIm:**
   - *Defend pins.*
   - *Tie improvement.*
     If a piece cannot move because it is tied to the defence of another piece P, defend piece P.
   - *Defend undefended pieces.*
   - *Defend against exchange of pieces.*
   - *Cover squares in own camp.*
     Moves that gain control over a square in one's own camp.
   - *Develop pieces.*
     Patterns and move sequences taken from expert games for 1) standard opening sequences, 2) building castles, and 3) positional pattern moves.

5. **PMG-DefPIm:**
   - *Pin piece.*
     Moving the pinned piece puts the king in check or loses material.
   - *Cover squares in opponent camp.*
     Moves that gain control over a square in the opponent camp.
   - *Avoid development.*
     Moves that do not allow the opponent to develop its position.

The hierarchy of PMGs can be used to improve the savings. An example is the dominance of *PMG-Goal*, *PMG-Th* and *PMG-DefTh* over *PMG-Pim* and *PMG-DefPim*. In our implementation, *PMG-Pim* and *PMG-DefPim* are not invoked if there is a strong threat to win material or a strong threat against either king.

2.3 Move Merit Analysis

At first glance, having multiple plausible move generators is not very efficient. Generating each plausible move only once is faster than having the same move generated several times by different move generators. However, the possibility of generating a move by more than one PMG is vital for our approach. If a move is generated by more than one PMG, it is often better than a move that is generated only once. For example, moving a piece away from an attack is in general more powerful if it is attacking an opponent piece at the same time.

Knowledge about which PMGs generated a move can be used to analyse the merit of the move. In our method, each PMG assigns a value to the generated moves based on the importance of the PMG. For example, the PMG generating piece captures will give a value to each capture based on the value of the piece that was captured.

Values in *PMG-Goal* are usually higher than values in *PMG-Th*, but not always. For example, the value of a check can be higher than the value of taking a pawn. Moves generated by *PMG-Goal* will always have higher values than moves generated by *PMG-Pim* or *PMG-DefPim*. Negative values are also possible, for example in case of a sacrifice.

The results of this *Move Merit Analysis* (MMA) can be used to make a move ordering. This in itself will improve the performance of alpha-beta search. However, the main advantage of using MMA is to make additional cuts in the number of candidate moves generated by the PMGs. A natural cut-off is to discard all candidate moves with a negative MMA value. In this paper, plausible move generation without MMA will be called *PMG-All* and plausible move generation that cuts all moves with a negative MMA value will be called *PMG-MMA*.

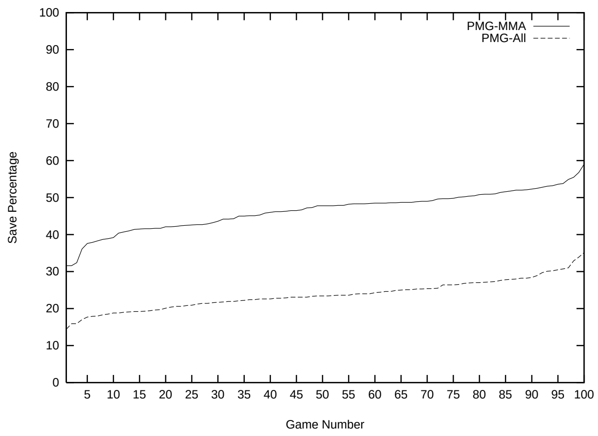

Figure 2: Savings of *PMG-All* and *PMG-MMA* in 100 test games, ordered by percentage of moves saved.

3 Results

We have analysed the behaviour of *PMG-All* and *PMG-MMA* with four tests: a plausible move generation test, a move ordering test, a search comparison test and a self play experiment between full-width search, *PMG-All* and *PMG-MMA*.

3.1 Plausible Move Generation Test

First, we looked at the savings and move generation accuracy of *PMG-All* and *PMG-MMA*. To do this, we used the 100 test games described in Section 2.1. These 100 test games have a total of 12097 positions. We tested the accuracy of *PMG-All* and *PMG-MMA* by checking if the move played by the expert was generated by *PMG-All* or *PMG-MMA*. We also calculated the savings of our approach, i.e. the difference between the total number of legal moves and the total number of moves generated by the two PMG versions. The savings for *PMG-All* and *PMG-MMA* are given in Figure 2. This figure shows that there are small areas of good and bad results, but that the majority of the savings are close to the average. It is also clear from the figure that the savings of *PMG-All* are much worse than the savings of *PMG-MMA*.

Vital is the balance between the savings of the plausible move generation and the accuracy. The savings and accuracy results of *PMG-All* and *PMG-MMA* are summarised in Table 1. *PMG-All*, the basic plausible move generation without any additional cuts, can generate 99.4% of all expert moves in the test games. This version on average reduces the number of moves with 23.7% compared to the total number of legal moves. The savings can be almost doubled if MMA is used. *PMG-MMA* saves 46.5% of all moves at the cost of 0.5% accuracy compared to *PMG-All*.

Version	NG	Ac(%)	Sv(%)
PMG-All	81	99.4%	23.7%
PMG-MMA	144	98.9%	46.5%

Table 1: Results of the *PMG-All* and *PMG-MMA* on 12097 positions. *NG* is the number of moves played by the expert, but not generated by the PMG version; *Ac* is the accuracy of the PMG version; *Sv* are the savings of the PMG version.

3.2 Move Ordering Test

We have also looked at the move ordering results of MMA for each position in the set of test games. The results for are given in Figure 3. MMA orders the expert move first in 17.3% of the positions. Almost half of the expert moves (49.6%) are ordered among the best five moves. If the first ten moves in the ordering are considered, 66.9% of the expert moves are produced by the PMGs.

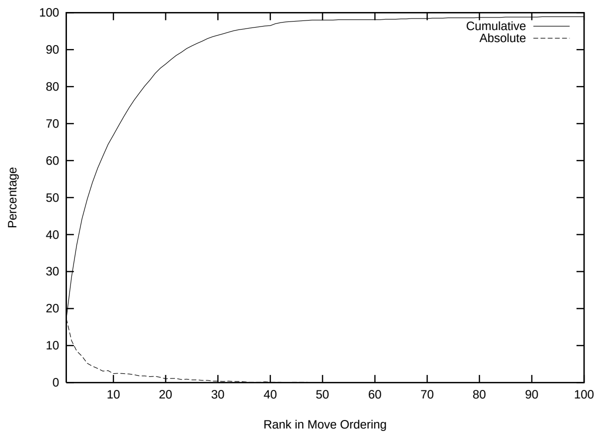

Figure 3: Absolute and cumulative move ordering results in 100 test games.

The absolute move ordering curve in Figure 3 shows that there are only very few expert moves ordered lower than 30.

3.3 Search Comparison Test

The most important question when using plausible move generation with MMA is whether the effort spent on the generation of plausible moves is worthwhile. To answer this question, we compared the search performance of *PMG-All* and *PMG-MMA* with full-width search. First, we used tactical shogi problems from the weekly magazine *Shukan Shogi*. The test set consists of 300 problems published in issues 762 to 811. Problems in each issue are divided into six classes, ranging from starting level to expert level. It should be noted that the starting level is already quite advanced and is too hard for beginners. Two of the problems in the test set are incorrect and have been removed from the test set.

All versions of the program were given 30 seconds per problem on a Pentium 700 MHz PC, which is about the same time as can be expected to be available under tournament conditions. All versions use the same evaluation function and the same standard alpha-beta scout search algorithm with transposition tables, history heuristic and search extensions for captures and checks.

The results of this test are given in Table 2. The categories in the table correspond to the categories in Shukan Shogi. The table shows that MMA is vital to our approach. There is almost no difference between the number of solved problems of full-width search and *PMG-All*. However, *PMG-MMA* solves significantly more problems than full-width search and *PMG-All*.

Cat	Tot	Full	PMG	
	Pos	width	All	MMA
1	50	17	21	22
2	50	9	8	13
3	50	10	10	11
4	50	9	7	8
5	50	5	4	4
6	48	4	5	6
Tot	298	54	55	64

Table 2: Results of full width search, *PMG-All* and *PMG-MMA* in tactical shogi problems.

3.4 Self Play Experiment

As a final experiment to compare full-width search, *PMG-All* and *PMG-MMA*, we played different versions of the same shogi program against each other. One program was using full-width search, one program was

No	Version	1	2	3	Result
1	PMG-MMA	x	15-5 (75%)	20-0 (100%)	35-5 (87.5%)
2	PMG-All	5-15 (25%)	x	16-4 (80%)	21-19 (52.5%)
3	Full-width	0-20 (0%)	4-16 (20%)	x	4-36 (10.0%)

Table 3: Results of a self play experiment between *Full-width*, *PMG-All* and *PMG-MMA*.

using *PMG-All* and one program was using *PMG-MMA*. Each program version played the other versions twenty times with a time limit of 20 minutes per side per game. This is the same time limit as used in the annual CSA tournament. The results of this tournament are given in Table 3. The results show that a PMG based program is playing better than a program without plausible move generation. Full-width search won only 4 out of 40 games. Also, *PMG-MMA* is much better than *PMG-All*, scoring 75% against *PMG-All*. This shows that move merit analysis is vital to our plausible move generation approach.

4 Related Work

Most of the top programs in computer shogi are made by professional programmers, not by researchers. Publications on the methods used in those programs are therefore scarce. From personal communication it seems clear that there are two basic approaches. One is to generate all legal moves and make dynamic decisions about which parts of the search tree to expand. This type of search is used by *Kanazawa Shogi*, winner of the 1999 CSA tournament and *Shotest*, the program that reached third place in 1999. Plausible move generation is the second approach and is used in *Kakinoki Shogi*, winner of the 1999 Computer Shogi Grand Prix and *YSS*, runner-up in the 1999 CSA tournament.

Kakinoki Shogi uses only 8 basic move categories [10]. Kakinoki's move categories correspond to the five basic move categories given earlier, with separate move categories for captures, promotions, king attack and king defence.

Yamashita's YSS [20] uses 29 move categories. His plausible move generation is strongly related to the search depth. Moves are only generated if the remaining search depth is enough to show that the move can actually reach the goal implied by the move category. For example, a move that attacks a piece is not generated at depth 1, because it is not possible to show that the attack will have a positive effect on the position. Such a detailed analysis of moves improves the quality of the plausible move generation, but also takes more time.

5 Conclusions

Full-width search has been the dominant approach in most game playing programs and has been the subject of much scientific research into two-player complete information games. In this paper we have argued that plausible move generation is an important alternative that deserves further investigation. There are games where the full-width search paradigm can not be successfully applied because of a large average number of search alternatives. Also, plausible move generation is interesting from a cognitive science point of view, as it is closer to human problem solving than full-width search.

We have proposed a plausible move generation method for shogi. This plausible move generation method uses five basic move categories that are general for a wide range of games. For each game these basic move categories need to be refined to match the specific features of the game under investigation. For shogi the five basic move categories resulted in 21 different plausible move generators. Each generated move is given a value based on the plausible move generator(s) that generated the move. This *Move Merit Analysis* (MMA) is an indication of the importance of a move. Moves with a negative merit value are natural candidates to discard, therewith further narrowing the candidate moves to search.

Results in shogi show that it is possible to get important savings in the search candidates without compromising accuracy. Savings of about 46% can be achieved, losing only 1% of moves chosen by expert players. Furthermore, the set of plausible move generators gives a good initial order of search candidates, which helps improve the performance of alpha-beta search.

In tactical shogi problems, plausible move generation without MMA performs only slightly better than full-width search, but PMG with MMA can solve significantly more problems. Finally, both plausible move generation with and without MMA play better than full-width search, but plausible move generation with MMA beat plausible move generation without MMA in 15 out of 20 test games. The tactical improvement and the considerable improvement in playing strength show that move merit analysis is vital to our plausible move generation method.

References

[1] T. Anantharaman, M.S. Campbell, and F. Hsu. Singular Extensions: Adding Selectivity to Brute-Force Searching. *Artificial Intelligence*, 43:99–109, 1990.

[2] D. Beal. A Generalised Quiescence Search Algorithm. *Artificial Intelligence*, 43:85–98, 1990.

[3] A. Bernstein and M. de V. Roberts. Computer v Chess-Player. *Scientific American*, 198:96–105, 1958.

[4] M. Buro. The Othello Match of the Year: Takeshi Murakami vs. Logistello. *ICCA Journal*, 20(3):189–193, September 1997.

[5] A.D. De Groot. *Thought and Choice in Chess.* The Hague, The Netherlands: Mouton & Co, 1965.

[6] R. Greenblatt, D. Eastlake III, and S. Crocker. The Greenblatt Chess Program. In *Proceedings of the Fall Joint Computer Conference*, pages 801–810, 1967.

[7] E. Heinz. Extended Futility Pruning. *ICCA Journal*, 21(2):75–83, June 1998.

[8] Japanese Shogi Federation. *Heisei 10 Shogi Nenkan.* Nihon Shogi Renmei, 1999.

[9] A. Junghanns and J. Schaeffer. Domain-Dependent Single-Agent Search Enhancements. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 570–575, 1999.

[10] G. Kakinoki. The Search Algorithm of the Shogi Program K3.0. In H. Matsubara, editor, *Computer Shogi Progress*, pages 1–23. Tokyo: Kyoritsu Shuppan Co, 1996. ISBN 4-320-02799-X. (In Japanese).

[11] R.E. Korf. Does Deep Blue Use Artificial Intelligence? *ICCA Journal*, 20(4):243–245, December 1997.

[12] H. Matsubara and K. Handa. Some Properties of Shogi as a Game. *Proceedings of Artificial Intelligence*, 96(3):21–30, 1994. (In Japanese).

[13] H. Matsubara, H. Iida, and R. Grimbergen. Natural developments in game research: From Chess to Shogi to Go. *ICCA Journal*, 19(2):103–112, June 1996.

[14] A. Newell, C. Shaw, and H. Simon. Chess Playing Programs and the Problem of Complexity. *IBM Journal of Research and Development*, 2:320–335, 1958.

[15] B. Pell. A Strategic Metagame Player for General Chess-like Games. *Computational Intelligence*, 12(2):177–198, 1996.

[16] J. Schaeffer. *One Jump Ahead: Challenging Human Supremacy in Checkers.* Springer-Verlag New York, Inc., 1997. ISBN 0-387-94930-5.

[17] J. Schaeffer and A. Plaat. Kasparov Versus Deep Blue: The Rematch. *ICCA Journal*, 20(2):95–101, June 1997.

[18] D. Slate and L. Atkin. Chess 4.5: The Northwestern University Chess Program. In P. Rey, editor, *Chess Skill in Man and Machine*, pages 82–118. Springer Verlag, New York, 1977.

[19] K. Thompson. 6-Piece Endgames. *ICCA Journal*, 19(4):215–226, December 1996.

[20] H. Yamashita. YSS: About its Datastructures and Algorithm. In H. Matsubara, editor, *Computer Shogi Progress 2*, pages 112–142. Tokyo: Kyoritsu Shuppan Co, 1998. ISBN 4-320-02799-X. (In Japanese).