

Plausible Move Generation in Two-player Complete Information Games Using Static Evaluation

Reijer Grimbergen

Electrotechnical Laboratory
grimberg@etl.go.jp, <http://www.etl.go.jp/etl/suiron/~grimberg/RESEARCH/>

Hitoshi Matsubara

Future University-Hakodate and PRESTO
matsubar@fun.ac.jp

keywords: heuristic search, game playing, plausible move generation, position evaluation, shogi

Summary

In games where the average number of legal moves is too high, it is not possible to do full-width search to a depth sufficient for good play. A way to achieve deeper search is to reduce the number of moves to search. In this paper a new method for *Plausible Move Generation* (PMG) will be presented that considerably reduces the number of search candidates. This plausible move generation method will be applied to shogi. We will present different types of plausible move generators for different types of moves, based on the static evaluation of a shogi position. Test results show that in shogi this set of plausible move generators reduces the number of moves to search by 33.2% on average. Plausible move generation is still very accurate: 99.5% of all expert moves in 12097 test positions were generated by our method. Search based on plausible move generation has also been compared with search without plausible move generation. First, in 298 tactical shogi problems, using plausible move generation increased the number of solved problems with 34%. Second, in a self-play experiment a shogi program based on plausible move generation beat a shogi program based on full-width search in 80% of the games.

1. Introduction

Full-width search has been very successful in two-player complete information games. DEEP BLUE in chess [Schaeffer 97b], CHINOOK in checkers [Schaeffer 97a] and LOGISTELLO in Othello [Buro 97] 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. Domain-dependent heuristics are then used to add selectivity: some moves will be searched deeper than other moves. Examples of methods to add selectivity to the search are *quiescence search* [Beal 90], *singular extensions* [Anantharaman 90] and *futility pruning* [Heinz 98a].

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 [Newell

58, Bernstein 58, Greenblatt 67]. The remaining candidates were then searched as deep as possible with alpha-beta search. For example, Bernstein's chess program [Bernstein 58] 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 [Slate 77].

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* [Matsubara 96] and single agent search problems such as *sokoban* [Junghanns 99]. Here plausible move generation can be a good alternative to full-width search.

Plausible move generation is also interesting for cog-

nitive 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 [Korf 97]). Cognitive science research has shown that human experts are capable of narrowing the search candidates to a very small number of relevant candidate moves [De Groot 65]. 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.

In this paper we propose a method of plausible move generation for two-player complete information games. This method is based on the general concept of goal and threat that these types of games have in common. In Chapter 2 we will define five basic categories for plausible move generation in two-player complete information games. In Chapter 3 we will apply this plausible move generation method to shogi. We will explain why shogi is a domain where plausible move generation is necessary and we will refine the basic plausible move generators for shogi. In Chapter 4 we will explain how position evaluation and plausible move generation interacts in shogi. In Chapter 5 we will give the performance results of our plausible move generation method. We will show that our method is accurate and gives considerable savings in the number of moves to search. We will also show that a shogi program with plausible move generation outperforms an identical shogi program that uses full-width search.

2. General PMG in Two-player Complete Information Games

METAGAMER [Pell 96] is a general game-playing system showing that 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 to 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. For example, in shogi the goal of the game is different than in Go and *PMG-Goal* is therefore different as well.

3. Plausible Move Generation in Shogi

3.1 Why is PMG 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 [Matsubara 94].

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 [Thompson 96]. This is not the case in shogi, where the branching factor of the search tree increases as the game pro-

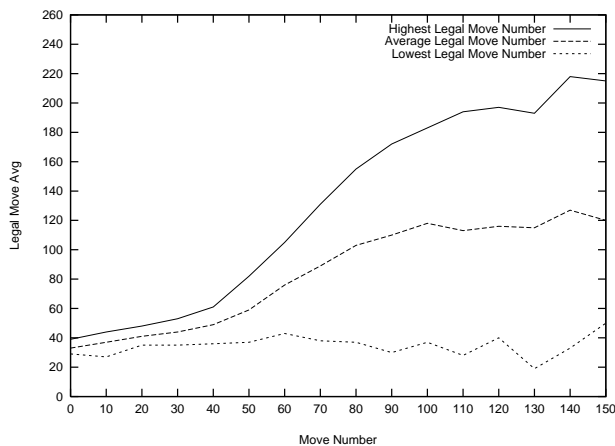


Fig. 1 Highest number of legal moves, average number of legal moves and lowest number of legal moves by move number in 100 test games.

gresses. 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 legal moves 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 [JSF 99], 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. Therefore, plausible move generation is a promising method to help improve the strength of shogi playing programs.

3.2 A Set of PMGs for Shogi

As pointed out, for each game the five basic PMGs of Chapter 2 have to be refined to fit the application domain best, as goals and threats are different in different application domains. For shogi, we have refined the five PMG categories 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.

4. Position Evaluation and PMG

Generating plausible moves is closely related to the evaluation of the current position. In Figure 2 the relations between the different features of the evaluation function and the plausible move generation are given. Plausible move generation also needs other information: *Position Data* to know which piece is on which square and *Piece Cover Data* to know which piece is covering which square. All plausible move generators need this information to generate moves. The connection between the position evaluation and the different plausible move generators is more complex. For example, a PMG to generate moves for attack and defence of the king needs information about the strength of the attack. This information is collected by the evaluation function. In the figure it can be seen that not all elements of the evaluation function and the plausible move generation are connected. The evaluation of material is simply counting the pieces on both sides and has no relation to the plausible move generation. In contrast, no position evaluation is used to generate checks, check defences, defence of undefended pieces and pinning of pieces. A further refinement of the method might make a difference between pins of pieces of different value. A pin of a rook would then be given a higher value than a pin of a pawn. In this case, there will be a relation between the position evaluation and the PMG *Pin piece*. This remains a future work.

The central evaluation feature is the *Static Exchange Evaluator*, which is used by most PMGs. The static exchange evaluator calculates the outcome of a piece exchange on each square without any search. In its simple form, an exchange evaluation for square S orders the pieces covering S for both sides in ascending piece value order. These two sets are then compared, where each side has the option of stopping the exchange at his turn. Depending on features of the position, the exchange evaluation can become quite

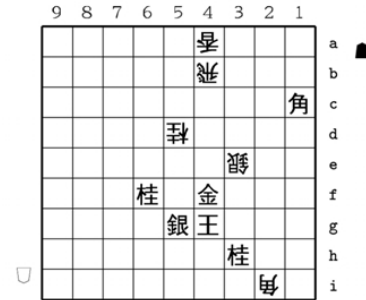


Fig. 3 Example of a complicated exchange situation.

complex. This is illustrated in Figure 3.

In Figure 3, the problem is to calculate the exchange value on 4f. In a normal ordering, the knight on 3h would be the first of black pieces. However, this knight is pinned by the bishop on 2i and can not move as long as bishop and king do not move. As for the white pieces, in the simple ordering the lance on 4a would be ordered higher than the rook on 4b, since a lance is of lower piece value than a rook. However, in this case the lance is blocked by the rook and should be ordered lower than the rook. A different type of block is the piece formation of the silver on 3e and the bishop on 1c. The bishop on 1c is not covering the square 4f, but as soon as the silver on 3e moves, the bishop will enter the exchange. Finally, black has a problem involving piece ties. The silver on 5g is tied to the defence of the knight on 6f as well. If the silver captures on 4f, the knight on 6f will be lost. Therefore, an exchange can have consequences for the exchange evaluation on other parts of the board as well. This example illustrates the importance of position evaluation for PMG and shows the potential complexity of this relation. A detailed description of the implementation of the static exchange evaluator is beyond the scope of this paper.

5. Results

We have analysed the behaviour of plausible move generation with three tests: a plausible move generation test, a search comparison test and a self play experiment between full-width search and PMG based search.

5.1 Plausible Move Generation Test

First, we looked at the savings and move generation accuracy of our PMG set. To do this, we used

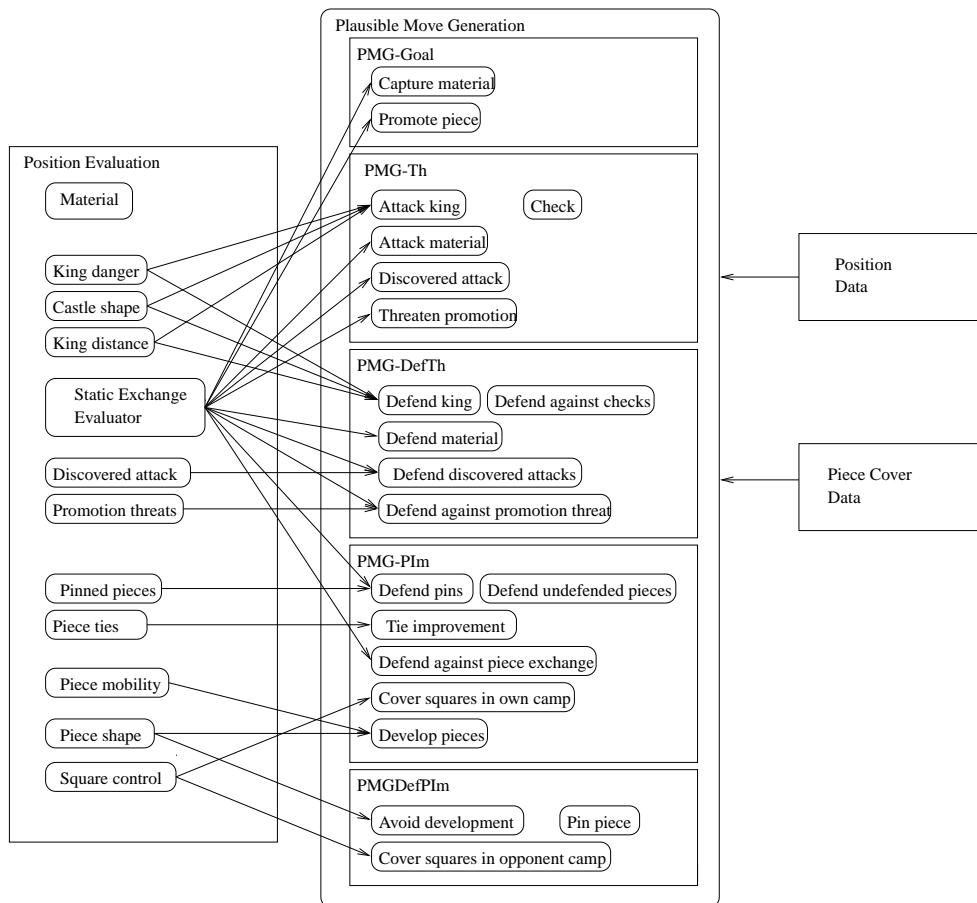


Fig. 2 Position evaluation and plausible move generation.

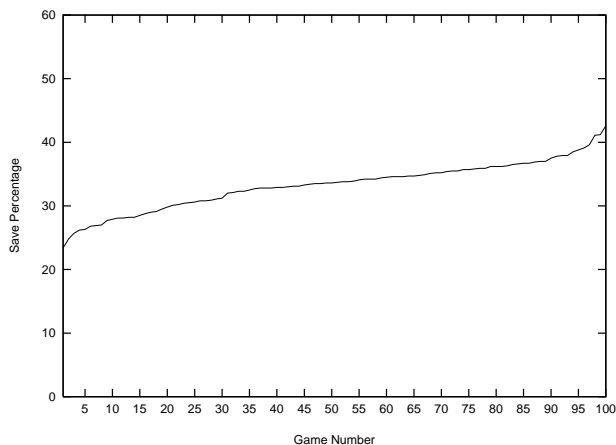


Fig. 4 Save percentages when using plausible move generators in the 100 test games.

the 100 test games described in Chapter 3. These 100 test games have a total of 12097 positions. We tested the accuracy of plausible move generation by checking if the move played by the expert was generated by any of the PMGs. 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 PMGs. The savings for the plausi-

ble move generation are given in Figure 4. Plausible move generation on average saved 33.2% of the total number of moves. The figure shows that there are small areas of good and bad results, but that the majority of the savings are close to the average.

Vital is the balance between the savings of the plausible move generation and the accuracy. In a total of 12097 positions there were 63 moves not generated by the PMGs, giving an accuracy of 99.5%. In Table 1 the accuracy results are summarised. There are 60 games where all expert moves were generated by the PMGs. An additional 26 games had only one move that was not generated. It is rare that the PMGs will miss more than 2 moves in a game. The maximum number of expert moves not generated was 5, which happened in two games. In both cases these moves were an unusual development of pieces in the opening stage of the game. These moves could not be generated by *PMG-Pim*. It is likely that there will be alternative moves available in these positions, so this does not point to a problem in our PMG approach.

5.2 Search Comparison Test

The most important question when using plausible move generation is whether the effort spent on the generation of plausible moves is worthwhile. To answer this question, we compared the search performance of plausible move generation 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. 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 a 34% improvement in the number of problems solved when using plausible move generators instead of full-width search: 62 problems solved instead of 41 problems solved.

5.3 Self Play Experiment

As a final experiment to compare plausible move generation with full-width search, we played two different versions of the same shogi program against each other. One program was using full-width search and one program was using PMGs. In all other respects the programs were identical. The programs played each other 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 match are given in Table 3. Plausible move generation clearly outplayed full-width search, winning

Table 1 Total number of games for each number of moves not generated by the PMGs.

#NG	#Games
0	60
1	26
2	10
3	1
4	1
5	2

Table 2 Results of full width search (FWS) and plausible move generation (PMG) in tactical problems.

Cat	Tot	FWS	PMG
1	50	12	20
2	50	6	14
3	50	10	12
4	50	4	7
5	50	4	4
6	48	5	5
Tot	298	41	62

16 of the 20 games, a winning percentage of 80%.

Table 3 Result of a 20-game match between PMG and FWS.

Version	Games	Score	WinPer(%)
PMG-FWS	20	16-4	80%

6. Related Work

In this section we will present different approaches for plausible move generation in shogi and compare them with the method presented in this paper. We will look at the plausible move generation in the shogi programs IS SHOGI, winner of the 1998 and 2000 CSA computer shogi championships; YSS, winner in 1997; and KAKINOKI SHOGI, winner of the Computer Shogi Grand Prix in 1999.

6.1 IS Shogi

IS SHOGI [Tanase 00] uses the following plausible move generators:

- Best move of the previous iteration.
- Capture opponent piece that just moved.
- Move piece that was attacked on the previous move to a safe square.
- Killer move.
- Null move.
- Attack king or attack material.
- Discovered attacks.
- Defend piece that was attacked on the previous move.
- Defence moves.
- Other special moves.

The categories are strictly ordered. If a move in a category leads to an alpha-beta cut-off or has a sufficiently high evaluation, none of the moves in the categories ordered below it will be generated. It is unclear which moves are generated by *Other special moves* as the description about this PMG is very short. There is only one example given of a special defence move to shut out pieces from attack.

The advantage of this plausible move generation method compared to our method is that no time is spent on generating moves that do not influence the search. Disadvantage is that the ordering of plausible move generators is defined statically and is unable to deal with the special features of the current position. With our method, we think it is possible to make better use of the evaluation of the current position for plausible move generation. This should lead to a better move ordering, which aids alpha-beta search.

6.2 YSS

Yamashita's YSS [Yamashita 98] uses 30 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. Also, some moves are only generated at the start of the search or based on the stage of the game (opening, middle game or endgame). Here are YSS's move categories in detail:

- **Remaining search depth is at least 1:**
 - Capture opponent piece that just moved.
 - Capture undefended piece.
 - Promote piece.
 - Checks that do not sacrifice material.
 - Move attacked piece with highest value.
- **Remaining search depth is at least 2:**
 - Defend against strong threat.
 - Attack material.
 - Discovered check.
 - Attack king from the front.
 - Discovered attack.
 - Attack pinned pieces.
 - Drops of bishop and rook in opponent camp.
- **Remaining search depth is 3 or higher:**
 - Attack pieces around the opponent king.
 - Attack tied defending pieces.
 - Capture material higher than pawn.
- **Moves only generated at the first ply of search:**
 - Develop useless pieces.
 - Sacrifices with check.
 - Pawn drops far from the promotion zone.
- **Moves only generated at the first two ply of search:**
 - Pawn pushes in front of rook and lance.

- Material sacrifices that lead to a fork.
- Pawn promotion sacrifice.
- Dangling pawn.
- Blocking opponent rook or bishop.
- Move gold sideways in opponent camp.
- Attack opponent piece that just moved with pawn drop.
- Attack opponent piece that just moved.
- Move king.
- **Moves only generated at the first two ply of search in the opening:**
 - Attack pawn with piece.
 - Drop pawn to make an attacking base (covering the opponent camp).
 - Develop useless pieces.

YSS uses more plausible move generators than our method (30 PMGs instead of 21 PMGs), which might improve the quality of the plausible move generation, but also takes more time. Also, control of the search seems difficult with a PMG method that depends heavily on the search depth. However, it is not difficult to use our plausible move generation method in the same way. Only minor modifications are needed to relate plausible move generation to the search depth. Further research is needed to investigate if this improves the performance of our method.

6.3 Kakinoki Shogi

Kakinoki Shogi uses 8 basic move categories [Kakinoki 96]:

- Capture material.
- Defend material.
 - Move away from attack.
 - Cover attacked piece.
 - Take attacking piece.
 - Interpose piece between attacker and attacked piece.
- Promote piece.
- Defend against promotion threat.
- Attack king.
- Defend king.
- Other attacks.
- Other defences.

Most of Kakinoki's move categories correspond to the PMGs in Section 3.2. Although not clear from his description, we assume that moves like *Attack material* and *Pin piece* fit in the category *Other attacks*, while moves like *Defend undefended pieces* are part of the category *Other defences*. Absent from Kakinoki's move category description are non-tactical moves for

piece development. Search in KAKINOKI SHOGI is a tactical search only and moves for piece development are handled differently. These moves become plans that can be played if there are no tactical problems detected by the search.

7. 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 general plausible move generation method for two-player complete information games and applied this method to shogi. The 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.

Results in shogi show that it is possible to get important savings in the search candidates without compromising accuracy. Savings of 33.2% can be achieved, losing only 0.5% of moves chosen by expert players.

In tactical shogi problems, plausible move generation performs better than full-width search. Finally, a shogi program based on plausible move generation outplayed a full-width search based program, winning 16 out of 20 test games. The tactical improvement and the considerable improvement in playing strength show that plausible move generation method is an important area of research in two-player, complete information games.

◇ References ◇

- [Anantharaman 90] T. Anantharaman, M.S. Campbell, and F. Hsu. Singular Extensions: Adding Selectivity to Brute-Force Searching. *Artificial Intelligence*, 43:99–109, 1990.
- [Beal 90] D. Beal. A Generalised Quiescence Search Algorithm. *Artificial Intelligence*, 43:85–98, 1990.
- [Bernstein 58] A. Bernstein and M. de V. Roberts. Computer v Chess-Player. *Scientific American*, 198:96–105, 1958.
- [Buro 97] M. Buro. The Othello Match of the Year: Takeshi Murakami vs. Logistello. *ICCA Journal*, 20(3):189–193, September 1997.
- [De Groot 65] A.D. De Groot. *Thought and Choice in Chess*. The Hague, The Netherlands: Mouton & Co, 1965.
- [Greenblatt 67] R. Greenblatt, D. Eastlake III, and S. Crocker. The Greenblatt Chess Program. In *Proceedings of the Fall Joint Computer Conference*, pages 801–810, 1967.
- [Heinz 98a] E. Heinz. Extended Futility Pruning. *ICCA Journal*, 21(2):75–83, June 1998.
- [Heinz 98b] E.A. Heinz. DARKTHOUGHT Goes Deep. *ICCA Journal*, 21(4):228–229, December 1998.
- [JSF 99] Japanese Shogi Federation. *Heisei 10 Shogi Nenkan*. Nihon Shogi Renmei, 1999.
- [Junghanns 99] A. Junghanns and J. Schaeffer. Domain-Dependent Single-Agent Search Enhancements. *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 570–575, 1999.
- [Kakinoki 96] 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).
- [Korf 97] R.E. Korf. Does Deep Blue Use Artificial Intelligence?. *ICCA Journal*, 20(4):243–245, December 1997.
- [Matsubara 94] H. Matsubara and K. Handa. Some Properties of Shogi as a Game. *Proceedings of Artificial Intelligence*, 96(3):21–30, 1994. (In Japanese).
- [Matsubara 96] 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.
- [Newell 58] 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.
- [Pell 96] B. Pell. A Strategic Metagame Player for General Chess-like Games. *Computational Intelligence* 12(2):177–198, 1996.
- [Schaeffer 97a] J. Schaeffer. *One Jump Ahead: Challenging Human Supremacy in Checkers*. Springer-Verlag New York, Inc., 1997. ISBN 0-387-94930-5.
- [Schaeffer 97b] J. Schaeffer and A. Plaat. Kasparov Versus Deep Blue: The Rematch. *ICCA Journal*, 20(2):95–101, June 1997.
- [Slate 77] 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.
- [Tanase 00] Y. Tanase. The IS Shogi Algorithm. In H. Matsubara, editor, *Computer Shogi Progress 3*, pages 1–14. Tokyo: Kyoritsu Shuppan Co, 2000. ISBN 4-320-02956-9. (In Japanese).
- [Thompson 96] K. Thompson. 6-Piece Endgames. *ICCA Journal*, 19(4):215–226, December 1996.
- [Yamashita 98] 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).

〔担当委員：新田克己〕

Received February 3, 2000.

Author's Profile

**Grimbergen, Reijer**

1989: MSc. in Computer Science from the University of Nijmegen (Holland). 1994: Ph.D. in Cognitive Science from the Nijmegen Institute of Cognitive Science and Information Technology (NICI). Since 1995 working as a researcher at the Machine Understanding Division of the Electrotechnical Laboratory, AIST, Ministry of Industrial Trade and Industry, JAPAN. Research

interests: search, game playing and cognitive modelling of human problem solving.

**Matsubara, Hitoshi (Member)**

Hitoshi Matsubara is a professor at department of Media Architecture, School of Systems Information Science, Future University - Hakodate, JAPAN. His current research interest include game programming, cognitive modelling of human thinking, multi-agent system, and machine learning. He was a visiting researcher at Stanford University from 1993 to 1994. Matsubara received his

Ph.D from Tokyo University in 1986.