

Computer shogi

Hiroyuki Iida^{a,*}, Makoto Sakuta^a, Jeff Rollason^b

^a *Department of Computer Science, Shizuoka University, 3-5-1, Johoku, Hamamatsu, Shizuoka 432-8011, Japan*

^b *OxfordSoftworks, Oxford, England, UK*

Abstract

This paper describes the current state of the art in computer shogi. Shogi (Japanese chess) promises to be a good vehicle for future research into game-playing programs that are based on tree-searching paradigms. This paper shows where chess and shogi are similar, and details the important areas that make shogi programming of particular interest. A crucial difference is the game-tree complexity, which is significantly higher in shogi than in chess. Three important differences are the “drop” rule, the diverging character of the game, and the slow build-up of forces. They make it difficult to have effective opening and endgame procedures.

After a short summary of the rules of shogi and an outline of the main areas of current work in computer shogi, we provide an overview of the history of computer shogi, in which computer-shogi activities both in human tournaments and in exhibition events are given. We conjecture that by the year 2010 a computer will be comparable in strength to the best human players. The most important techniques used in computer shogi are described. We focus on issues such as opening play, selective search, quiescence search, solving tactical exchanges without tree searching, position evaluation and endgame play. At the end the key challenges in computer shogi are enumerated, and finally, concluding remarks are given. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Shogi; Computer shogi; Alpha-beta search; Selective search; Quiescence search; Evaluation function

1. Introduction

In the early days of artificial-intelligence research, priority was given to the games of chess [57] and checkers [55]. These games are Western games and constituted excellent challenges to the modern computers of the 1950s. In Japan computers entered society in the late 1950s. The start of the IPSJ (Information Processing Society of Japan) was in

* Corresponding author.

E-mail addresses: iida@cs.inf.shizuoka.ac.jp (H. Iida), sakuta@cs.inf.shizuoka.ac.jp (M. Sakuta), jeffrollason@cs.com (J. Rollason).

Table 1
The complexities of chess, shogi and Go

	Chess	Shogi	Go
State-space complexity	10^{43}	10^{71}	10^{172}
Game-tree complexity	10^{123}	10^{226}	10^{360}

1960. Twelve years later the GPCC (Games and Puzzles Competitions on Computers) was introduced as a part of the annual IPSJ symposium. Then, Japanese researchers looked for challenges comparable to the checkers and chess programs described in the literature [51]. Their obvious choice was the game of shogi. Would it be possible to program a computer to play shogi? The discussion on this question went along with a discussion on the differences with the other two games, since shogi has a well-known “drop” rule. Whatever the case, Japanese researchers started AI research in the domain of shogi programming. They adopted many techniques from the computer-chess world, but were obliged to find their own way in many cases, mainly as a consequence of the differences in the rules.

Shogi, or Japanese chess, is a two-person zero-sum game with perfect information. It is similar to chess; the goal of the game is also equivalent (capturing the opponent’s King). The most important difference is shogi’s drop rule, which allows captured pieces to be re-used by “dropping” them back on the board. Shogi is very popular in Japan, with an estimated 15 million or more players. The state-space complexity and the game-tree complexity make shogi an interesting target for game programming. It is harder than chess but less so than Go. Table 1 shows the state-space complexity and game-tree complexity of shogi, chess and Go. The game-tree complexity of shogi is based on a branching factor of 80 and an average game length of 115 plies. The assumptions for chess are 35 and 80, and for Go 250 and 150, respectively [40].

This paper describes the current state of the art in computer shogi. In Section 2 we describe the key differences between chess and shogi. Section 3 shows the increase of playing strength of computer-shogi programs. Section 4 outlines the general approaches of computer shogi, and focuses on important issues that have to be considered when making a shogi program. Section 5 itemizes the main challenges that need to be addressed in computer shogi. Section 6 describes relevant studies. Finally, concluding remarks are given in Section 7.

2. Key differences between shogi and chess

In this section we look at the key differences between shogi and chess. Section 2.1 contains the basic rules of shogi. In Section 2.2 we highlight the main rule differences between shogi and chess. Section 2.3 shows the implications of these differences and how they affect the development of a shogi program.

2.1. The basic rules of shogi

Shogi is played by two players, *Black* and *White*, on a board of 9×9 squares. The players move alternately. The goal is to capture the opponent’s King. The starting position is shown

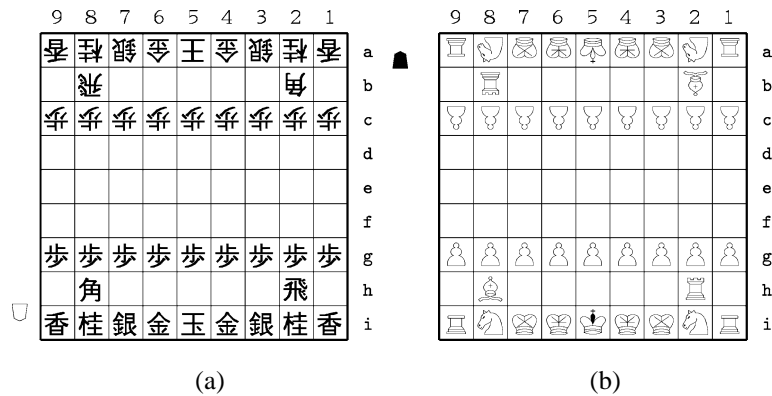


Fig. 1. The initial position for a game of shogi. (a) Using a set of kanji pieces. (b) Using a set of chess-like pieces.

in Fig. 1. Each player has 20 pieces. Black starts the game and plays with the pieces on the bottom three ranks. White's pieces are located on the top three ranks. In Fig. 1, Black's pieces are being displayed normally, while White's pieces are displayed upside down, i.e., the pieces are not distinguishable by colors.

The captured pieces are shown next to the board. Black's pieces are on the right and White's pieces on the left, as shown in Fig. 2. Since the traditional kanji pieces shown in Fig. 1 may not be familiar to most readers, for the rest of the paper we use chess pieces instead. Moreover, most modern Japanese shogi literature uses a symbol to indicate each move and the captured pieces (pieces-in-hand) as either Black's (♣) or White's (♠).

Each square of the board is represented using an algebraic notation similar to chess but with a different starting point. For example, the White King in Fig. 1 is on square 5a. If it were to move to 4b, this move would be represented by ♔5a–4b or ♔4b in short notation. As can be seen in Fig. 1 Black's camp is in the bottom three ranks (i to g), which is also White's promotion zone. Correspondingly, White's camp is in the top three ranks (a to c), which is Black's promotion zone. Most pieces in shogi can promote, but promotion is not obligatory as it is in chess.

In shogi there are eight different kinds of pieces. In the initial position each player has a King (玉 or ♔ in a chess-like symbol set), a Rook (飛 or ♖), a Bishop (角 or ♗), two Golds (金 or ♘), two Silvers (銀 or ♙), two Knights (桂 or ♞), two Lances (香 or ♞), and nine Pawns (歩 or ♟). The following explains how each of these pieces, and their promoted forms, move on the board. A King moves like a King in chess, so it can move one square in all directions (horizontally, vertically and diagonally). A King cannot promote. A Rook moves like a Rook in chess, any number of squares horizontally or vertically but without the ability to make the castling move. The promoted Rook (竜 or ♙) keeps its original movement, but with the added ability to move one square diagonally in all directions. A Bishop also moves like a Bishop in chess, i.e., any number of squares diagonally. The promoted Bishop (馬 or ♘) keeps its original movement, with the added ability to move one square horizontally or vertically in all directions.

A Gold moves like a King, except that it cannot move to either of the two squares diagonally backward. A Gold can thus move to six squares. A Gold does not promote.

A Silver moves like a King, except that it cannot move directly left, right or backwards. A Silver can thus move to five squares. A promoted Silver (銀 or 銀) moves like a Gold. A Knight moves like its chess counterpart, but in this case only one square forward followed by one square diagonally forward. For example, a Black Knight on 5f can only jump to 6d or 4d. A promoted Knight (馬 or 馬) moves like a Gold. A Lance can vertically move any number of squares, but only forward. A promoted Lance (龍 or 龍) moves like a Gold. A Pawn can only move forward, both capturing and moving in the same direction, with no diagonal or double moves. A promoted Pawn (兵 or 兵) moves like a Gold.

Elaborate introductions to shogi have been written by Leggett [37], Fairbairn [9] and Hosking [19], with other shogi texts available on the Internet (see [15]).

2.2. Differences in rules

Below we provide the main differences between shogi and chess by five items.

- (1) Chess is played on an 8×8 board, while shogi uses 9×9 .
- (2) In chess there are 6 different types of pieces; shogi has 8. In chess the players have 32 pieces in total (16 pieces each), while in shogi the players have a total of 40 pieces (20 pieces each).
- (3) In chess only the Pawn is allowed to promote. In shogi promotion is allowed for 6 different kinds of pieces. Also, in chess promotion is only allowed on the last rank (8th rank for White; 1st for Black). In shogi promotion is possible on the back three ranks of the enemy camp, that is for Black the ranks 7 to 9 (for White they are ranks 1 to 3). Promotion can also occur by moving in, out or within the promotion zone (three cases). Promotion is optional except in a few minor cases.
- (4) The most significant difference between shogi and chess is the possibility of reusing pieces in shogi. When a piece is captured, this piece does not disappear from the game (like in chess), but is held by the capturing side. If it is a player's turn, the player can either choose to play a move with a piece on the board or take one of the pieces previously captured and put it on a vacant square on the board (this type of move is called a *drop*). There are almost no limitations to where a piece can be dropped; even giving mate by putting a captured piece back on the board is allowed.
- (5) A draw by agreement or a draw because of the fifty move rule is not possible in shogi. Stalemate is theoretically possible, but because of the possibility of dropping pieces on the board this has never happened in a normal game. However, there is the possibility of *impasse (jishogi)*, where both Kings enter into the enemy camp and can no longer be mated. On average, only 2 out of every 1,000 grandmaster games end in jishogi.

2.3. Differences in programming

Shogi differs from chess in some essential properties. We mention (1) the drop rule, (2) the diverging character of the game due to the drop rule, and (3) the slow motion of the pieces resulting in a prolonged start before the opposing forces meet each other. Therefore, the shogi programmer has to be careful when adopting computer-chess techniques. Below we discuss the three points mentioned. For point (2) we distinguish two separate cases.

(1) A consequence of the reuse rule is that the number of tactical threats in shogi tends to be much larger than in chess. This has enormous implications for the evaluation function, since a position that is statically assessed as being quiescent may be anything but quiescent; a drop move can significantly change the assessment. It implies that the criteria for quiescence must be revisited and this may have ramifications for the way the horizon effect is handled. This problem can be partially alleviated by deeper search and by domain knowledge. Not surprisingly, even human experts have difficulty in accurately evaluating positions with captured pieces.

(2a) Chess is a converging game in which the number of possible moves generally decreases in the later stages of the game, while shogi is a diverging game where the number of possible moves basically increases from the middle game to the endgame because of the drop move. The results of experiments performed with two different computer programs show that the average number of experiments performed with two different computer programs show that the average number of possible moves in a game is 106 [53], while the value 80 was obtained based on the analysis of many game scores played by grandmasters [40]. Fig. 2 shows a position from the second match game for the prestigious professional *Meijin* title between the strong players Y. Sato and T. Maruyama. Here, White (upper side) resigned. In this position, White has 262 possible moves, while Black has 150. These are common values in the endgame of shogi.

(2b) The diverging nature of shogi also affects the construction of endgame databases. In a converging game such as chess, the use of endgame databases enables a computer to play optimally in a particular case. However, in shogi they are of no benefit.

(3) Many pieces in shogi have limited movement. This leads to a slow build-up of forces and influences the average game length and the opening database. There are a large number of books written on opening theory in shogi, but these generally only consider patterns rather than precise opening move sequences. New and interesting opening patterns are still being developed. Therefore, building a good opening database is more difficult in shogi than in chess.

We have seen that shogi and chess are in some aspects similar and in other aspects very different. From a programming point of view the differences are important. Apparently, a crucial difference is the game-tree complexity, which is significantly higher in shogi than in chess. Moreover, the tactical threats by the drop rule, the diverging nature of the game,

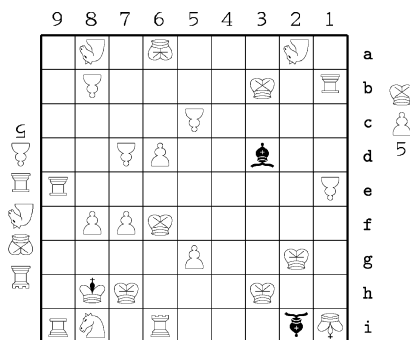


Fig. 2. An example of a position in an actual game played by two top players.

and the shogi-specific features such as a slow build-up of forces constitute obstacles for developing a strong shogi program.

3. Playing strength of computer-shogi programs

The history of computer shogi only covers some thirty years. Since the game is more complex than chess, we may not expect that computer performances are at the same level. For chess it took some fifty years to develop a world-championship-caliber program. Although shogi programmers benefit from the computer-chess techniques developed so far, there still are more obstacles. In this section we deal with the progress in playing strength of computer-shogi programs. We start in Section 3.1 with a brief overview, the definition of playing strength, and the foundation of the Computer Shogi Association (CSA). In Section 3.2, we focus on the increase of playing strength by analyzing the eleven CSA tournaments played so far. Section 3.3 contains a comparison of computer strength with the playing strength of human beings.

3.1. Early history of computer shogi

The first working computer-shogi program was written in 1974 by a research group supervised by Takenobu Takizawa [59]. The program did not cope well with the game-tree complexity, so it could only play at the level of a beginner, analogous to the development of chess programs. In the 1980s and 1990s, substantial improvements in hardware speed and software techniques increased the strength of chess programs to grandmaster level. Although computer shogi benefited from the chess experience published worldwide, it was, as yet, unable to raise the playing strength to grandmaster level. At the moment there are many commercial shogi programs on the (Japanese) market, the strongest of which act at the level of a relatively strong amateur. In the Japanese grading system this roughly corresponds to a level of 3-dan or 4-dan, being the second or third grade of the strong amateur class. (For an overview of the playing strength we refer to Table 3.)

A strong impetus to computer shogi was given by Yoshiyuki Kotani and Takenobu Takizawa. They established a society for the study of computer shogi, the Computer Shogi Association (CSA), in Japan in 1986 [8]. The CSA has motivated the leading researchers in computer shogi to describe their programming techniques in books [41], and to exchange ideas at academic symposiums such as the series of Game Programming Workshops.

3.2. The CSA computer-shogi championship

An annual computer shogi tournament has been organized by the CSA since 1990. Table 2 shows the results of these tournaments. There have been two other major computer-shogi tournaments: the Grand Prix in 1999 jointly won by KAKINOKI and YSS [59] and the 2000 Mind Sports Olympiad won by YSS [29].

The CSA computer-shogi tournaments are growing, as the number of participants has grown, from 6 in the first tournament in 1990 to 55 in the 11th tournament in 2001. Since 1992 KIWAME (later renamed as KANAZAWA, authored by Shinichiro Kanazawa) has

Table 2
Results of CSA tournaments 1990–2001

	Date	Winner	2nd	3rd
1st CSA	Dec, 1990	EISEI MEIJIN	KAKINOKI	MORITA
2nd CSA	Dec, 1991	MORITA	KIWAME	EISEI MEIJIN
3rd CSA	Dec, 1992	KIWAME	KAKINOKI	MORITA
4th CSA	Dec, 1993	KIWAME	KAKINOKI	MORITA
5th CSA	Dec, 1994	KIWAME	MORITA	YSS
6th CSA	Jan, 1996	KANAZAWA	KAKINOKI	MORITA
7th CSA	Feb, 1997	YSS	KANAZAWA	KAKINOKI
8th CSA	Feb, 1998	IS	KANAZAWA	SHOTEST
9th CSA	Mar, 1999	KANAZAWA	YSS	SHOTEST
10th CSA	Mar, 2000	IS	YSS	KANAZAWA
11th CSA	Mar, 2001	IS	KANAZAWA	KCC

won the CSA tournament five times. Other past winners are MORITA (Kazuro Morita), YSS (Hiroshi Yamashita), and IS (Yasushi Tanase, Norifumi Goto, Akihiro Kishimoto, and Ayumu Nagai). IS is the reigning champion and has won the CSA championship three times. There have also been some non-Japanese entries. The best results have been achieved by SHOTEST (Jeff Rollason, Great Britain), which entered for the first time in 1997 and finished third in 1998 and 1999. Other non-Japanese entries have been SHOCKY (Pauli Misikangas, Finland), SPEAR (Reijer Grimbergen, The Netherlands), and KCC (a team with main programmer An Kyong Nam, North Korea), which finished third in the 2001 tournament (see also [14,54]).

The latest CSA tournament in 2001 was held over three days. On the first day, 36 programs competed in the first preliminary stage of seven rounds according to the accelerated Swiss pairing system. Eight programs advanced to the second preliminary stage of 24 programs (8 programs and 16 qualified programs) on the second day. In the second preliminary stage a nine-round accelerated Swiss tournament was played with the top seven programs qualifying for the next stage. On the final day the three best programs from the previous year (IS SHOGI, YSS and KANAZAWA SHOGI) and the seven programs that qualified from the second preliminary stage played a round-robin to decide the winner of the tournament. The time limits for each game were 25 minutes (minimum 1 second per move) for all tournament stages.

In shogi, the system used to measure playing strength has been traditionally different from the Western way where chess players are familiar with the Elo-rating system. The shogi grading scale of 'kyu' and 'dan' is in use for human players as well as for computers. Table 3 shows a rough correspondence between the Japanese grading scale and the Elo-rating system. The year of performance indicates the increase of the playing strength of the best shogi program in the CSA computer tournaments [36,60]. Based on this data,

Table 3

A rough correspondence between the Japanese grading scale and the Elo-rating system

Elo rating	Japanese grading scale	Year of performance
1200	Amateur 4-kyu	1990
1400	Amateur 2-kyu	1995
1600	Amateur 1-dan	1996
1800	Amateur 2-dan	1997
2000	Amateur 3-dan	1999
2200	Amateur 4-dan	2001
2300	Amateur 5-dan	
2400	Amateur 6-dan	
2500	Professional 4 or 5-dan	
2600	Professional 6 or 7-dan	
2700	Professional 8 or 9-dan	
2800	Title holders	
2900	Champion	

Iida predicted that shogi programs will be able to compete with the strongest humans (i.e., Champion ranking) by 2010 [23].

3.3. Computer versus human

So far, only on a few occasions have computers been allowed to participate in human shogi tournaments, excluding tournaments on the Internet. In addition, there have been a few exhibition matches between human experts and computers. Below we give a brief account of two tournaments with program participants and four exhibition matches.

In May 1997 YSS and KANAZAWA (CSA championship winners in that year) participated in the B-class (from 1-dan to 3-dan) of an open rating competition (Asakusa, Japan) [65]. The time limits for each game were 30 minutes for the human players and 40 minutes for the computers (in consideration of the time lost by operating the program). Once the time had expired, a player had to move within 30 seconds. KANAZAWA won the tournament with a score of 3–0, while YSS finished with a score of 1–3. In 2000, KAKINOKI was invited to take place in the B-class (from 1-dan to 3-dan) of an amateur human tournament (Hokkaido, Japan) [33]. KAKINOKI won all its games and finished in first place. Here, the time limit for each game was 20 minutes per player, with 30 seconds per move thereafter.

In September 1996, as a side event of the Game Programming Workshop (Hakone, Japan) there was an exhibition match between Hiroyuki Iida (professional 6-dan) and two computer programs KAKINOKI and MORITA. The games were played simultaneously with the grandmaster giving each program a 6-piece handicap (Rook, Bishop, two Knights and

two Lances). KAKINOKI won the first game very quickly and therefore a second game was started. The latter game was won by Iida [22]. Meanwhile MORITA won by virtue of its excellent endgame play [24]. In May 1998, a young girl (2-dan amateur) played simultaneously with three computer-shogi programs: KAKINOKI, YSS, and MORITA. She lost all the games. The exhibition was shown on television [25]. In September 1998, there was another exhibition match between Iida and three computer programs, as a side event of the National Conference of the Japan Society for Industrial and Applied Mathematics (Tokyo, Japan). Iida played simultaneously against IS, YSS, and KAKINOKI while giving each a 4-piece handicap (Rook, Bishop and two Lances). All three programs lost [24]. In July 1999, Grandmaster Akira Shima (8-dan) played KAKINOKI without a handicap. The computer lost [26]; the grandmaster was too strong. Again the latter exhibition match was on television, this time under the title: ‘Can a computer beat a human?’

Nowadays shogi programs have many opportunities to play with human players on the Internet shogi forum. The programs are ranked, and KANAZAWA has obtained an Elo score of 2030 on the shogidojo.com site (Shogi Club 24). The highest score on that site is around 2100.

4. Techniques used in computer shogi

Since shogi is similar to chess, many techniques proven to be effective in computer chess have been adapted for use in shogi programs. Hence, strong shogi programs have a structure similar to chess programs. Basically, a shogi program builds minimax game trees explored by an iterative α - β search. Shogi programs also make use of a variety of refinements, such as PVS-search, transposition tables, quiescence search, move ordering, aspiration search, null-move pruning, history heuristic and killer heuristic. However, to deal with the large number of legal moves in shogi, new techniques and domain-specific methods had to be developed. Moreover, shogi-specific features, such as the diverging nature of the game, made it necessary to explore other methods. In this section we examine the characteristics of the game that require a different solution than seen in chess programs. The shogi-specific elements are opening play, selective search, quiescence search, solving tactical positions, position evaluation and endgame play.

4.1. Opening

The opening phase is generally defined as the stage from the start of a game until a player is ready to attack. Using this definition, the opening phase generally lasts 20 moves on average. An opening database, such as is used for chess, has not yet been established for shogi.

Kakinoki was the first to describe and implement an idea for building a shogi opening book [32]. The opening data consists of move groups represented by a graph structure. There are two types of moves stored in the book: good moves that are recommended to be chosen, and bad moves after which a potentially good attack can be stored. The drawback of the idea is that the program cannot deal well with the case where an out-of-book position transposes back into the opening database. In shogi such cases happen very often since

a strict move sequence in the opening phase is usually not required. It looks like a matter of transposition, but there are differences due to the representation of move groups.

In chess playing an out-of-book move in the opening is risky; in shogi even a grandmaster sometimes plays such a move, especially against an opening expert. Most shogi games do not follow book moves for very long. However, games played by grandmasters show that a poor formation or poor development can rapidly lead to a disadvantageous position. Making a good formation in shogi is very important for obtaining a positional advantage or for keeping the positional balance even. Most shogi programs have an opening database. KAKINOKI has stored roughly 1,000,000 positions [32]. Using the database, programs can build a grandmaster-like opening formation if both sides follow exactly the same opening line.

To deal with the re-entering problem, several suggestions have been proposed. The main idea is to continue the building of a good formation that would have arisen from the opening line played so far, even when the opponent does not exactly follow the normal opening sequence [48,49]. This requires consideration of potentially tactical plays (i.e., the playing of one move of a move group) that might upset the creation of a previously-planned good formation. In practice it is not so easy to realize this idea, i.e., identifying whether the opponent's out-of-book moves require the choice of a different target formation. So far only a few programs address this problem, and consequently it is common to see strange positions arising from the opening (from the grandmaster's point of view) in games played by computer programs.

To avoid re-entering problems, Yamashita uses an opening database in his program YSS only when both players exactly follow an opening line in the database. As soon as an out-of-book position is reached, his program stops using the database [67]. YSS will continue to develop pieces using the so-called *otoshiana method*, i.e., constructing a castle using hill climbing. YSS has piece-square tables for this purpose for nine popular opening lines, including Yagura, Ibiana and Furibisha. The opening line to be followed is determined at random or is selected depending on the position of the opponent's Rook. According to this method, a piece formation will be made based on the difference between the scores given to a piece occupying a square and given to the pieces on the neighboring squares of the square to which the piece would move. The bigger the difference, the sooner the piece will want to move to that square. With this method a good formation will be made after a reasonable number of moves. This idea makes it possible to make a good formation even when the position is out-of-book. The method is also followed by KANAZAWA [34] and SHOTEST [53].

Fig. 3 illustrates a pitfall for opening book programs. In a game between KANAZAWA and YSS taken from the finals of the 9th CSA Championship, the game continued with 8. ♞3f ♜4a–5a 9. ♞3g ♞2d, after which YSS won easily. 8. ♞3f and 9. ♞3g are very weak moves. Since the move sequence in this opening was not normal, KANAZAWA did not understand the opening line of YSS (it is again a re-entering problem). Such weak opening play was not apparent in the 2001 CSA tournament. The opening databases are slightly improved, but there still remains a considerable gap between grandmasters and computers due to the re-entering problem (or the representation in move groups). Proper opening play remains a serious impediment to the development of strong shogi programs.

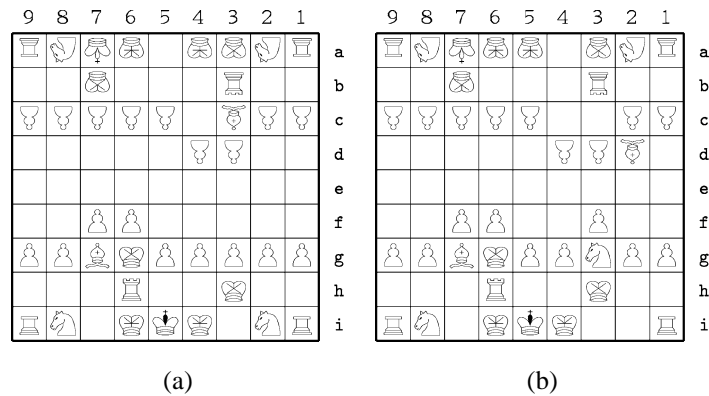


Fig. 3. An example opening played by the best programs. (a) After 7. ... ♔6b–7a. (b) After 8. ♗3f ♘4a–5a 9. ♙3g ♚2d.

In the openings, the current strength of the strongest programs is somewhere in the region of amateur 1-dan or less.

4.2. Selective search—plausible move generation

Brute-force search has been very successful in chess. However, it has not always been the main approach used. In the early days of chess research, most programs did a selective search based on generating only “reasonable” moves. A selective move generator would identify a sub-set of plausible moves using domain-specific knowledge [6,10,50]; the remaining moves would be discarded. For example, Bernstein’s chess program [6] generated only 7 plausible moves in any position. Such selectivity is very risky as good moves may be discarded without any search at all. In chess it was found that the probability of error in selective search was too high. This observation led to the discovery that brute-force search (considering all moves) proved more effective than their selective-search counterparts. In recent years, the trend has reversed as improved algorithms and hardware have resulted in chess programs with highly selective searches.

In shogi the need to be selective about which moves to search is much greater than in chess. It is a consequence of the larger move lists that have to be considered. Several shogi programs, such as EISEI MEIJIN, have experimented with the brute-force search approach, but most shogi programs use selective search based on plausible move generation. The following considers the main approaches adopted by the strongest shogi programs.

IS SHOGI uses ten plausible move generators for different categories of moves [61]. The plausible move generators are considered in the following order: (1) the best move of the previous iteration, (2) capture the opponent piece that just moved, (3) prevent the opponent attack caused by the previous move, (4) a killer move [1], (5) the null move [4], (6) attack moves (capture, attack, promotion, aiming at promotion, and so on), (7) discovered attacks, (8) defend pieces and squares that were attacked on the previous move, (9) defense moves, and (10) other moves that do not look interesting but should be examined (e.g., the dropping

of Pawn just behind the own Rook). If a move in a category leads to an alpha-beta cut-off or has a sufficiently high evaluation, no other moves in the following categories will be generated. This approach has the advantage of saving time, but risks missing moves that might show up as being better after a deeper lookahead.

YSS uses 30 move categories [68]. The plausible move generation is related to the search depth. Moves are generated only if the remaining search depth is sufficient 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 the deepest level since there is insufficient remaining search depth to evaluate properly the consequences of the threat. At depth 1 or higher YSS generates candidates in five categories: (1) capture the opponent piece that just moved, (2) capture an undefended piece, (3) promote a piece, (4) give a check that does not sacrifice material, and (5) move an attacked piece with highest value. At depth 2 or higher we find: (6) defend against a strong threat, (7) attack material, (8) discovered check, (9) attack King from the front, (10) discovered attack, (11) attack pinned pieces, and (12) drops of Bishop and Rook in the camp of the opponent.

SHOTEST makes extensive use of plausibility analysis to select moves to search [53]. The most dominant input comes from SUPER-SOMA, described later, which is capable of predicting complex attacking and defensive moves. In addition, SHOTEST avoids selecting moves that are “similar” to ones that have already been tried, e.g., making too many moves that hit other pieces or too many defensive moves. The current line is also analyzed so that moves near previously played moves are given priority. Each move gets a net value for a combination of plausibility credits. A move may therefore be selected for having many low-weighted features in preference to another which has a single high-weighted feature. SHOTEST recognizes about 100 move types in its plausibility analysis.

Since the large number of candidate moves makes forward pruning in shogi unavoidable, evaluating the effectiveness of plausible move generation is very important. The decision-making process as to which plausible moves to consider is therefore quite sensitive, as there are many more opportunities to overlook moves than in chess. An evaluation which throws up many plausible moves may create a move list that is too large. The evaluation needs to be selective in choosing moves that would be normally considered as strong candidates, if there are too many similar moves. The plausibility analysis needs to be able to choose moves that not only look interesting, but become so when too many similar stronger moves have already been considered.

With one exception, the computer-shogi literature is devoid of a detailed analysis of plausible move generation. Recently, Grimbergen has proposed a plausible move generator using move merit analysis; each move is assigned a score based on the move’s influence on capturing, promotion, attack, defense, and some kinds of threats [13]. Experiments performed using his program SPEAR show that the method includes the best move (as judged by human experts) in the plausible move list in about 93 to 99 percent of the cases, while pruning 46 to 68 percent of the total number of legal moves. SPEAR obtained the 13th rank in the second preliminary stage of the 11th CSA tournament.

Finally, we would like to mention that plausible move generation can be combined with move ordering. For instance, KANAZAWA first generates plausible moves and then orders these moves according to five cases [34]. They are: (1) take a piece with the highest value, (2) defend an attacked piece with the highest value, (3) take move(s)

from the transposition table, (4) play killer moves, and (5) play the null move. If none of these moves are sufficiently good to cause a cutoff, then all remaining moves are generated.

4.3. Quiescence search—selective deepening

In shogi it is more difficult than in chess to assess whether a leaf position is sufficiently stable (“quiescent”) so that the lookahead search can be stopped. The difficulty resides in the threats caused by the drop moves. A solution to the evaluation of these unstable positions is to extend the search until a seemingly stable position is reached. Below we describe three different approaches to selective deepening.

KANAZAWA SHOGI [34] uses the selective-deepening powers of six search-extension heuristics: (1) sacrifice extensions, (2) recapture extensions, (3) check extensions [58], (4) castle-attack extensions (for the best response against capturing a crucial piece that belongs to the castle), (5) deep mate-threat extensions, and (6) single-reply extensions [38]. The sacrifice extensions add two plies to the search depth, if the sacrifice is generated at an internal node of the game tree and the evaluation of the node falls within the alpha-beta window. Due to time constraints, in practice such an extended move is only started in the upper part of the tree. To prevent explosive growth of the search tree caused by excessive deep-search extensions, KANAZAWA SHOGI restricts the use of the search extensions to the early plies of the search by using a possibly-adaptive depth threshold.

IS SHOGI is able to deal with a sequence of moves as if it were a single move (or macro move), resulting in an implicit search extension. Fig. 4 illustrates this idea. If Black plays ♖6f×6e, White would respond with ♙2b×7g+, which will be followed by ♘8ix7g and ♙8e–8f. In this case, the move sequence ♙2b×7g+ ♘8ix7g and ♙8e–8f is recognized as one move, resulting in a two-ply search extension.

YSS uses Levy and Broughton’s SEX (Search EXtension) algorithm [38]. The performance of this algorithm is encouraging, since YSS has performed well in the CSA tournaments [66,68].

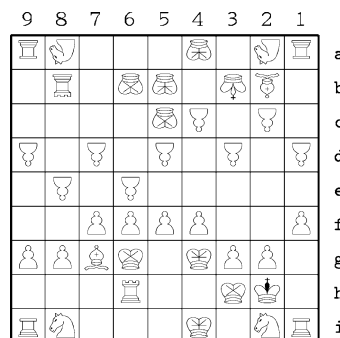


Fig. 4. An example of move-sequence tactics used in IS SHOGI (Black to move).

4.4. Solving tactical positions without search: SUPER-SOMA

Some chess programs determine the material balance of a position by a static analysis of all possible capture-move sequences. These routines are often referred to as SOMA (“Swapping Off Material Analyzer”) [39,43]. In the domain of computer shogi, Jeff Rollason proposed an algorithm called SUPER-SOMA [53], an enhanced SOMA algorithm with shogi-specific features. The general operation of SUPER-SOMA is to calculate a table of all single-square threats and then iteratively cycle through this table, ordering the priority of each tactical move, choosing a move and then updating the table. The idea is best explained by considering captures around one single square. Cycling through the table produces all moves that are involved in the capture. These moves are usually ordered by their material value. Since pins may be involved, the table should be updated continuously in order to produce legal moves. One further step is to take into account features such as ‘overloading’. If the first capturing piece is an overloaded Knight and the second piece is a free Rook, then the capture sequence may be re-ordered. The SUPER-SOMA algorithm allows separate exchanges on the board to be linked and prioritized. In complex positions, with many interrelated exchanges on several squares, the cycle is repeated until neither side can play a beneficial move.

In Fig. 5 SUPER-SOMA considers 6 plausible moves (three for each side) as shown in Table 4. At the first iteration of SUPER-SOMA, the move ♖6dx4d seems to be worse because of its negative value (−560). However, the cross-referencing mechanism of SUPER-SOMA allows it to find that ♖6dx4d is the move to play. This is possible as SUPER-SOMA detects six linkage features in the position which are resolved to give the highest weighting to this move. This process iterates to give the complete sequence: ♖6dx4d, ♘3fx4d and ♙4cx8g+. This is a very simple example. A more complex and detailed example appears in the full paper [53]. We mention that SUPER-SOMA can deal with many other characteristics, such as multi-square captures, pins, ties, discovered attacks, promotions, defensive play, mate threats and even positional moves.

SUPER-SOMA has been incorporated in SHOTEST. Analyzing SHOTEST’s performance in the CSA tournaments indicates that it performs badly in the opening (lack of shogi knowledge) and endgame stages (no mating search). Despite these shortcomings,

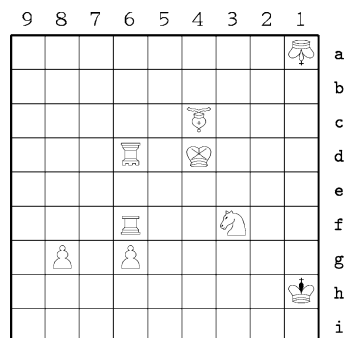








Fig. 5. An example position analyzed by SUPER-SOMA (White to move).

Table 4

Six moves are generated after the whole board is analyzed by SUPER-SOMA

Move	Value	Type of move
 6fx6d	960	Black's Lance is capturing White's Rook on 6d.
 4dx4c+	740	Black's Silver is capturing White's Bishop on 4c and promote.
 6dx4d	−560	White's Rook is capturing Black's Silver on 4d.
 6dx6f	−720	White's Rook is capturing Black's Lance on 6f.
 4cx8g+	152	White's Bishop is capturing Black's Pawn on 8g and promote.
 4d5c+	20	Black's Silver is moving to square 5c and promote.

SHOTEST finished third in the 1998 and 1999 CSA tournaments. It thus performs adequately in the middle-game stage where positions can become very complex. Even though SHOTEST examines roughly 50 times fewer nodes than other strong programs, it seems to play well in this stage of the game.

4.5. Position evaluation

Every shogi program contains an evaluation function to assess the merit of the positions that it encounters during its search for a move. There are many features employed in these evaluation functions. From the grandmaster's point of view, the balances between material and development in the opening phase, material and positional terms in the middle game, and king safety and initiative in the endgame, are the most important parts of the evaluation function [34,61].

The first step in designing an evaluation function is to determine the relative values of the pieces. Experiments to determine piece values have been performed using temporal difference learning, in the same manner as has been done for Western chess pieces [5]. If one normalizes the value of a Rook to 5.0, then the shogi pieces (and their promoted counterpart) score as follows: Pawn 0.53 (3.11), Lance 1.04 (1.51), Knight 1.18 (2.35), Silver 2.61 (2.55), Gold 3.02, Bishop 3.58 (6.34) and Rook 5.0 (8.14). These values are recognized as reasonable by human expert players.

The learned value for promoted Pawns requires additional discussion as it is consistently greater than the learned values for promoted Lances, Knights, and Silvers. First, promoting a Pawn has the additional benefit of making all empty squares in that file available for the subsequent dropping of a captured Pawn. In passing we note that the rule called *double Pawn* (*Nifu* in Japanese) prohibits dropping a Pawn on a file that already contains a friendly un-promoted Pawn. Second, the value of promoted pieces will not be maintained; when the opponent captures a piece it will not keep its promoted value. For example, a captured promoted Pawn reverts to being a natural Pawn for the opponent.

Grimbergen discusses a set of features that can be vital for an evaluation function [11]. The features are similar to those that have been used in the strongest programs such as KAKINOKI, KANAZAWA, YSS and IS. The main features are: (1) material balance, (2) piece placement, (3) shape, development, and castling, (4) king safety, and (5) initiative. KANAZAWA also considers the mobility of the King and major pieces (Rook and

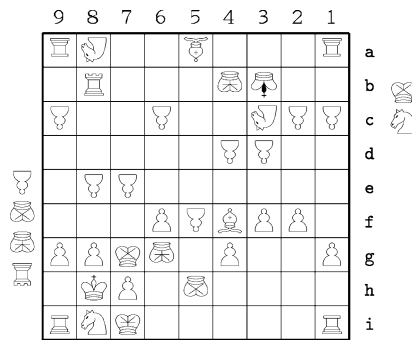


Fig. 6. An example of the drawbacks in position evaluation. After 48. ... ♖*42 (dropping), Black to move.

Bishop) [34]. Since evaluating a position is relatively expensive, each program uses some performance enhancements. For example, IS ignores non-material components when the material balance is greater (or less) than beta (or alpha) in an α - β search [61].

A typical evaluation function is a linear function of weighted features. The difficult point is that the weight of each feature greatly varies depending on the stage of a game. For example, in the opening, material balance is the most important feature, but in the endgame it is of minor importance compared to the king safety and initiative features. Moreover, the weight is further complicated by the unusual property that a game can regress from a middle-game position to an opening position, or from an endgame position to a middle-game position, a consequence of the drop rule. This makes context-dependent feature weighting critical, as the relative values of pieces and features can radically change during the game.

Fig. 6 shows an example of the obvious drawbacks in position evaluation which occurred in the game between KANAZAWA and KCC that was played in the final of the 11th CSA tournament. In this position, KANAZAWA (Black) could win by selecting 49. ♖*5d (dropping), ♜*5c 50. ♜5dx4b+ ♜5ax4b 51. ♜*5b. KANAZAWA instead made a wrong choice by playing 49. ♜8bx8a+ (taking a material gain) and led the game into an unclear position. This error was caused by the program overestimating the value of material in the endgame stage and not correctly weighting the initiative and king safety.

4.6. Endgame—mating search and brinkmate search

Given that endgame positions can revert back into middle-game positions, it is difficult to provide a clear definition of when the endgame occurs in shogi. We assume that the endgame begins as soon as a player starts to attack the opponent's King. In shogi the average branching factor in the endgame is much larger than in chess, rendering perfect endgame play practically impossible. Therefore, we need sophisticated search algorithms to play a master-level endgame.

The endgame usually results in a mating race, where the speed of the attack has the highest priority. It implies that a player would like to speed up their own mating attack and to slow down the opponent's attack. Losing a strong piece such as a Rook often leads to a disaster in the opening and middle game, but it can be completely insignificant in the

endgame. In the early endgame stage many tactics involve sacrifices which are offered to accelerate the King attack. In the next stage, it is usually a race between the two players to see who mates the other first. Therefore, the main task is to find the correct sacrifice(s) that lead to a mating attack. There are two search approaches addressing simplified versions of endgame play: tsume-shogi and brinkmate. We remark that in tsume-shogi problems and brinkmate problems, usually the attacker's King is not on the board.

4.6.1. Mating search (*tsume-shogi*)

The first simplification of the endgame search is a mating search called *tsume* or *tsume-shogi* in Japanese. In a mating search, the attacker is only allowed to play checking moves, until either the checks are exhausted or the defender is mated. Research in specialized mating-problem solvers has been performed concurrently and in close conjunction with the development of shogi programs.

There are two types of mating search: (1) fixed-depth full-width search, and (2) variable-depth search. Conventionally, the first type of mating search, which is based on iterative-deepening depth-first search, is incorporated in most shogi programs. It can find very short mating sequences (e.g., within 7 or 9 plies), but fails in positions requiring a deep search. Yet, this type of mating search is used in many programs for at least three reasons: (1) it is easy to implement, (2) it adequately handles time control, and (3) a lookahead of 7 to 9 plies suffices in most cases.

Variants of the proof-number search algorithm [2], such as PN*, can be used for variable-depth mating searches [12,56]. PN* is a depth-first iterative-deepening algorithm for AND/OR trees. A description of PN* is given in [56] where a tsume-shogi-solving program based on PN* is described in detail. It solved most problems of a difficult set of human-composed problems, including the famous “Microcosmos” position with a solution sequence of 1525 plies. Subsequently, PN* was refined to use both proof- and disproof-numbers, and it is now called PDS [44,45]. PDS (Proof-number and Disproof-number Search) behaves asymptotically the same as proof-number search. Even more recently, another new algorithm, df-pn search (meaning the depth-first proof-number search) [46], has been proposed. Its behavior is the same as the proof-number search behavior in the sense that a most-proving node is expanded. The advantage of df-pn search is that it is a depth-first type of search whereas proof-number search is a best-first search. Finally we remark that Nagai's program based on both PN* and df-pn has solved nearly all the existing hard tsume-shogi problems, including several problems that Seo's program based on PN* alone was unable to solve [47].

The ideas of AND/OR tree search mentioned above have been investigated in the domain of tsume-shogi, without the imposition of real-time constraints. The critical issue is how these mating search algorithms can be effectively used in a tournament shogi program.

4.6.2. Brinkmate search

The second simplification is a “brinkmate” search called *hisshi* in Japanese. The definition of a brinkmate uses notions as threatmate and threat sequence [21]. A threatmate is a move by the attacker leading to a position such that the attacker can mate the defender's King on the next move. A threat sequence is a move or sequence of moves which may contain threatmates or checks by the attacker or responses by the defender to threatmates

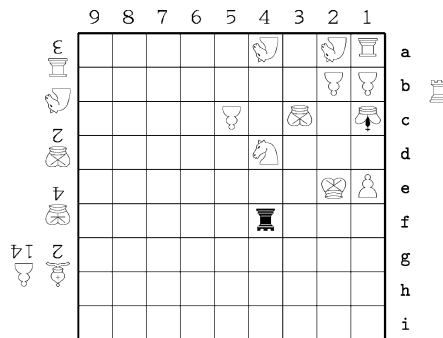


Fig. 7. Naito's brinkmate problem with a solution of 17 plies.

or checks. A brinkmate is then defined as a threat sequence leading to a position such that the attacker can mate the defender on the next move against any of the defender's responses except a check. It is obvious that a brinkmate search is more general and therefore more important than a mating search, but it is also more expensive to compute.

Fig. 7 shows an example of a brinkmate problem. The position was created by Kunio Naito (shogi grandmaster 9-dan) in 1999 with a comment: "This is a hard brinkmate problem that a computer will never solve". It was too good a challenge to pass up, and the brinkmate solver TACO-H successfully found the solution [16]. The program uses several heuristics to generate efficiently a threatmate and its response(s) in a PDS search. There are only a few studies on brinkmate search, even though this is an important component of a competitive shogi program [3,21]. In a brinkmate search, a mating search may have to be called many times. Hence, brinkmate searching can be very expensive and it must be used judiciously.

Brinkmate search is used in the program KFEND, which finished on 6th place in the 10th Computer Shogi Championship. However, the contribution of the algorithm to the program's overall performance is not so clear. Yamashita, the programmer of YSS, commented critically on the use of brinkmate search in an actual endgame [67]. One reason is that a program cannot completely rely on the outcome of the brinkmate search (in the sense of a brinkmate solver) since the attacker's King is also on the board, but this is not so in brinkmate problems. Another reason is that it is quite hard to deal with the counterattack of the defender during the brinkmate sequence. So, the process is complicated and on top of that it takes much time.

4.6.3. Mating search in the actual endgame

As stated above, in most shogi programs, iterative-deepening depth-first full-width search is used to find a mate. The mating search is only tried at the root node of a game tree. Predictably, short mating sequences are found and longer ones are out of reach. Recently, however, the variable-depth ideas (PN*, PDS and df-pn) have been tried under tournament time constraints, resulting in some long mating sequences (over 20 plies). Consequently, mating searches are now also possible at interior nodes close to the root.

Fig. 8 shows an example of endgame play from a game between IS SHOGI and YSS (final of the 10th CSA championship). In the position of Fig. 8(a), Black played

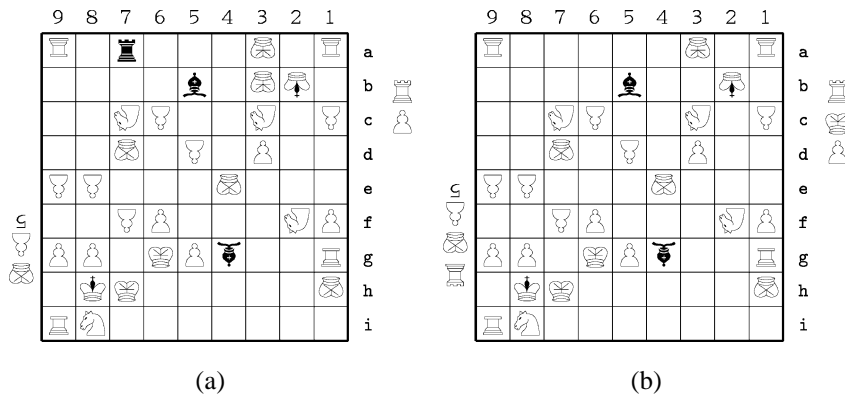


Fig. 8. An endgame position of a game between IS and YSS. (a) After 56. ... ♜3gx4g. Black to move. (b) After 57. ♜7ax3a ♕3bx3a. Black to move.

57. ♜7ax3a in 38 seconds, which was followed by ♕3bx3a in 30 seconds (see Fig. 8(b)). Black then checkmated White's King after 58. ♜3dx3c+. All of Black's checking moves in the main 23-ply mating sequence were played instantly. We note that the position of Fig. 8(a) can be disproved by an analogous mating search since White's reply 57. ♕3bx3a is wrong. Better is 57. ♔2bx3a. Black found the mating possibility by performing the mating searches in the internal nodes.

An important enhancement to mating-search efficiency is that some of the results of a search can be re-used in subsequent searches. IS SHOGI uses such a technique to determine quickly whether a defender's move affects the threatmate possibility [61] (it is similar to a method proposed by Kawano [35]). If White's King is assumed to be threatened with mate with White to move, a null move is played and a mating search for Black is performed. The mating sequence (if found) is stored in a transposition table. Now when White's moves are searched, the mating sequence previously found can be used to eliminate quickly many useless moves that do not defend against the mate threat.

4.7. New search algorithms

Besides selective search, as discussed in Section 4.2, there are other interesting approaches that have yet to gain a strong following. One such approach is the ABC game-tree search, proposed by McAllester and Yuret [42]. This algorithm is used in KFEND but its impact is unclear. A second new approach is a depth-first search based on using the probability with which a position will be realized from the root position of a game tree as the search threshold [7]. This idea has been incorporated in GEKISASHI, which reached the final of the 11th CSA tournament. To determine the probability that a position will be realized from the root of a game tree, statistics of various moves in actual games have been analyzed. For this task 600 game scores played by Habu, the best human shogi player, have been taken. The statistics suggest that 'a move that captures (with material gain) the opponent piece that just moved' has been played in more than 50 percent of the cases, and

‘a check that captures a piece with material gain’ has been played in about 43 percent of all check moves. Using such statistical information as a threshold for best-first-like search, GEKISASHI tries to approximate a “human-like” search.

5. Challenges for computer-shogi research

Below we formulate four main challenges for building a computer-shogi program capable of beating the human champion.

- *Master-level opening.* A prerequisite for a champion program is proper opening play. So far it is a serious impediment to achieving high performance. Hence this topic needs considerably more research attention.
- *Perfect play in the endgame.* In a position where one starts to attack the opponent’s King (the early stage of the endgame), a champion program should exactly (or almost exactly) know who is to win. New algorithms and faster hardware are needed to build such a program. In cases where the outcome of the analysis is not a win, the program should adopt a speculative strategy to anticipate possible mistakes of the opponent. Speculative play in the endgame is an important research area that has received only limited attention in the literature, e.g., [30,31].
- *Shogi hardware for deep lookahead.* As in chess, computer-shogi researchers are looking for specialized hardware to assist the search. Hori et al., propose using reconfigurable field-programmable gate arrays [18]. The processor takes care of the piece relations. The hardware is still in an early stage of development, although preliminary analysis suggests that it will operate at a speed of 62-times faster than a current single processor.
- *Organization of a title match* (computer versus best human). A strong motivation for future developments is the organization a man-machine title match. The strongest program should be given the opportunity to compete against the World Champion. Although computers are currently given only 25 minutes for a complete game (in the CSA tournaments), shogi grandmasters are given 5 hours in official tournaments and 9 hours in title matches. It will be interesting to see how the programs perform when given additional computing time.

6. Relevant studies

There are shogi-related research topics that do not directly contribute to building a strong program. Below we provide a brief overview of such investigations.

- *Composition of mating problems:* Recently, three attempts have been made to develop computer programs for composing shogi mating problems with artistically appealing features [17,52,63]. In the so-called *algorithmic generation* method [52], a starting position is first obtained by placing at random some pieces on the board for both sides. Then the mate finding procedure starts. First a mate position is derived by transforming (reducing) the original position into another simpler position. The transformation process follows a partial ordering starting in the original position.

Another method, called *reverse method* [17], is similar to the retrograde search as used in building chess endgame databases. However, a completeness check for the specific rules of shogi mating problems is necessary using the constraints for decreasing the search space. These two ideas have been combined in [63].

- *Speculative play*: One of the factors deciding an outcome of a game is strategic surprises. We may assume that mistakes cause a change in the theoretical outcome of the game, even in games played between shogi grandmasters. The drop rule in shogi provides a player with a high number of possible surprises. Strategic surprises have motivated studies on opponent modeling [64]. The point is to obtain a better result than the one dictated by the minimax strategy. Opponent modeling is a generic idea. It can be applied in any domain of games and in particular with emphasis on speculative play [20,62].
- *Evolutionary changes of shogi game*: The evolutionary changes when going from Heian shogi (an old version of shogi) to modern shogi have been investigated using computer analysis [27]. The analysis suggest the reason why the initial 8×8 board of Heian shogi was changed to the 9×9 board used today. The analysis is largely based on the outcome of King and Gold vs. King endgames. As a consequence of this research, a new measure of decision complexity in the domain of shogi (and also chess-like games) has been proposed [28]. This may give some insight into the evolutionary history of chess-like games. The novelty of such an approach is combining computer analysis with an historical study when investigating the evolutionary history of games.

7. Concluding remarks

Shogi programs have been improved significantly in the past few years. In a speed game it is possible for a computer to win against the strongest amateur players (roughly 2300 Elo rating). Since it is still rare for computers to be allowed to participate in human tournaments, it is not clear how strong the best shogi programs actually are. Given that computer games are played at a pace that is at least 10 times faster than human games, the question of calibrating the playing strength is even more difficult. Yet, we conjecture that by 2010, or shortly thereafter, a computer will be as strong as the best human player.

Currently, the strongest shogi programs still have some clear weaknesses, in particular in the opening and the middle game. In the endgame stage the mating search works well, and a computer can almost outperform the best human players. The first attempt to build shogi hardware has been made. Although the hardware is still in its infancy, it promises to raise the level of computer play significantly.

Finally, one of the significant advances is the development of AND/OR tree-search algorithms for the domain of mating searches. Moreover the strongest programs are now able to deal well with the tactical threats which often occur in shogi due to the drop moves. Since the last two developments are still not perfect, shogi remains a promising domain for future research into game-playing programs that are based on tree-searching paradigms.

Acknowledgements

The authors thankfully acknowledge the help of two anonymous referees. Their comments and suggestions on earlier versions of this paper are gratefully incorporated. Apart from them we are pleased to thank Jaap van den Herik and Jonathan Schaeffer for their constructive remarks.

References

- [1] S.G. Akl, M.M. Newborn, The principal continuation and the killer heuristic, in: *Proceedings, ACM '77 National Conference*, ACM Press, New York, 1977, pp. 446–473.
- [2] L.V. Allis, M. van der Meulen, H.J. van den Herik, Proof-number search, *Artificial Intelligence* 66 (1994) 91–124.
- [3] M. Arioka, Brinkmate search in KFEnd, http://plaza9.mbn.or.jp/~k fend/inside_k fend/hissi.html, 2000 (in Japanese).
- [4] D.F. Beal, A generalised quiescence search algorithm, *Artificial Intelligence* 43 (1) (1990) 85–98.
- [5] D.F. Beal, M.C. Smith, First results from using temporal difference learning in shogi, in: H.J. van den Herik, H. Iida (Eds.), *Proceedings of First International Conference on Computers and Games, CG'98, Lecture Notes in Computer Science*, Vol. 1558, Springer, Heidelberg, 1999, pp. 113–125.
- [6] A. Bernstein, M. de V. Roberts, Computer vs chess-player, *Scientific American* 198 (1958) 96–105.
- [7] Chikayama-Taura Lab, Search algorithm for shogi program GEKISASHI on the Internet Web page <http://www.logos.t.u-tokyo.ac.jp/~gekisashi/>, 2001 (in Japanese).
- [8] Computer Shogi Association, <http://www.computer-shogi.org/>.
- [9] J. Fairbairn, *Shogi for Beginners*, Shogi Association Ltd., London, 1984.
- [10] R. Greenblatt, D. Eastlake III, S. Crocker, The Greenblatt chess program, in: *Proceedings of the Fall Joint Computer Conference*, 1967, pp. 801–810.
- [11] R. Grimbergen, An evaluation function for shogi, in: H. Matsubara (Ed.), *Proceedings of Game Programming Workshop in Japan '97, Hakone, Japan*, 1997, pp. 159–168.
- [12] R. Grimbergen, A survey of Tsume-Shogi programs using variable-depth search, in: H.J. van den Herik, H. Iida (Eds.), *Proceedings of First International Conference on Computers and Games, CG'98, Lecture Notes in Computer Science*, Vol. 1558, Springer, Heidelberg, 1999, pp. 300–317.
- [13] R. Grimbergen, Plausible move generation using move merit analysis with cut-off thresholds in Shogi, in: I. Frank, T.A. Marsland (Eds.), *Proceedings of Second International Conference on Computers and Games, CG2000, Lecture Notes in Computer Science*, Vol. 2063, Springer, Heidelberg, 2001, pp. 331–349.
- [14] R. Grimbergen, H. Iida, Report on the 8th CSA world Computer-Shogi Championship, *ICCA J.* 21 (2) (1998) 135–138.
- [15] R. Hare, Shogi—Japanese chess, <http://www.edinburgh.ac.uk/~rjhare/shogi/intro.html>.
- [16] T. Hashimoto, On-going work on brinkmate solver, *J. Computer Shogi Assoc.* 13 (2000) 44 (in Japanese).
- [17] M. Hirose, H. Matsubara, T. Itoh, The composition of Tsume-Shogi problems, in: H.J. van den Herik, J.W.H.M. Uiterwijk (Eds.), *Advances in Computer Chess*, Vol. 8, Universiteit Maastricht, The Netherlands, 1997, pp. 299–318.
- [18] Y. Hori, M. Seki, R. Grimbergen, T. Maruyama, T. Hoshino, A Shogi processor with a field programmable gate array, in: I. Frank, T.A. Marsland (Eds.), *Proceedings of Second International Conference on Computers and Games CG2000, Lecture Notes in Computer Science*, Vol. 2063, Springer, Heidelberg, 2001, pp. 311–330.
- [19] L.V. Hosking, *The Art of Shogi*, The Shogi Foundation, England, 1997.
- [20] H. Iida, J.W.H.M. Uiterwijk, H.J. van den Herik, I.S. Herschberg, Potential applications of opponent-model search, Part 1. The domain of applicability, *ICCA J.* 16 (4) (1993) 201–208.
- [21] H. Iida, F. Abe, Brinkmate search, in: *Proceedings of Game Programming Workshop in Japan '96, Hakone, Japan*, 1996, pp. 160–169.
- [22] H. Iida, Grandmaster's exhibition handicap games with shogi programs at GPW'96, *J. Computer Shogi Assoc.* 10 (1997) 8–9 (in Japanese).

- [23] H. Iida, Computer shogi will come of age: Toward challenging human supremacy in shogi, *IPSI Magazine* 39 (9) (1998) 872–876 (in Japanese).
- [24] H. Iida, Computer vs grandmaster, *J. Computer Shogi Assoc.* 11 (1998) 6–9 (in Japanese).
- [25] H. Iida (Ed.), *J. Computer Shogi Assoc.* 11 (1998).
- [26] H. Iida (Ed.), *J. Computer Shogi Assoc.* 12 (1999).
- [27] H. Iida, J. Yoshimura, K. Morita, J.W.H.M. Uiterwijk, Retrograde analysis of the KGK endgame in shogi: Its implications for ancient Heian shogi, in: H.J. van den Herik, H. Iida (Eds.), *Proceedings of First International Conference on Computers and Games, CG'98, Lecture Notes in Computer Science, Vol. 1558, Springer, Heidelberg, 1999*, pp. 318–340.
- [28] H. Iida, N. Sasaki, T. Hashimoto, J.W.H.M. Uiterwijk, H.J. van den Herik, Towards a classification of games using computer analyses, in: N. Neuwahl (Ed.), *Proceedings of the International Colloquium "Board Games in Academia III", Florence, 1999*, pp. 139–152.
- [29] H. Iida, YSS wins shogi tournament, *ICCA J.* 23 (3) (2000) 184–185.
- [30] P.J. Jansen, Problematic positions and speculative play, in: T.A. Marsland, J. Schaeffer (Eds.), *Computers, Chess, and Cognition, Springer, New York, 1990*, pp. 169–182.
- [31] P.J. Jansen, KQKR: Speculatively thwarting a human opponent, *ICCA J.* 16 (1) (1993) 3–17.
- [32] Y. Kakinoki, The search algorithms in the Shogi program K3.0, in: H. Matsubara (Ed.), *Advances in Computer Shogi, Kyoritsu Shuppan, Tokyo, 1996. Chapter 1*, pp. 1–23 (in Japanese).
- [33] Y. Kakinoki, KAKINOKI won the Sapporo Meijin (B-class) title, *J. Computer Shogi Assoc.* 13 (2000) 22–27 (in Japanese).
- [34] S. Kanazawa, The algorithms in Kanazawa Shogi, in: H. Matsubara (Ed.), *Advances in Computer Shogi, Vol. 3, Kyoritsu Shuppan, Tokyo, 2000. Chapter 2*, pp. 15–26 (in Japanese).
- [35] Y. Kawano, Using similar positions to search game trees, in: R.J. Nowakowski (Ed.), *Games of No Chance, MSRI Publications, Vol. 29, Cambridge University Press, Cambridge, 1996*, pp. 193–202.
- [36] Y. Kotani, Which computer shogi is strong?, in: H. Matsubara (Ed.), *Advances in Computer Shogi, Vol. 3, Kyoritsu Shuppan, Tokyo, 2000*, pp. 111–120 (in Japanese).
- [37] T. Leggett, *Shogi: Japan's Game of Strategy*, Charles E. Tuttle Company, England, 1966.
- [38] D.N. Levy, D. Broughton, The SEX algorithm in computer chess, *ICCA J.* 12 (1) (1989) 10–21.
- [39] D.N. Levy, M. Newborn, *How Computers Play Chess*, Computer Science Press, England, 1991.
- [40] H. Matsubara, H. Iida, R. Grimbergen, Natural Developments in game research, *ICCA J.* 19 (2) (1996) 103–112.
- [41] H. Matsubara (Ed.), *Advances in Computer Shogi, Vols. 1–3, Kyoritsu Shuppan, Tokyo, 1996, 1998, 2000* (in Japanese).
- [42] D.A. McAllester, D. Yuret, Alpha-beta-conspiracy search, <http://www.research.att.com/~dmac/>.
- [43] D. Michie, Game playing and game learning automata, in: L. Fox (Ed.), *Advances in Programming and Non-Numerical Computation, Pergamon, Oxford, 1966*, pp. 183–200.
- [44] A. Nagai, A new AND/OR tree search algorithm using proof number and disproof number, in: *Proceedings of Complex Games Lab Workshop, ETL, Tsukuba, Japan, 1998*, pp. 40–45.
- [45] A. Nagai, A new depth-first-search algorithm for AND/OR trees, M.Sc. Thesis, Department of Information Science, University of Tokyo, Japan, 1999.
- [46] A. Nagai, H. Imai, Application of df-pn+ to Othello endgames, in: *Proceedings of Game Programming Workshop in Japan '99, Hakone, Japan, 1999*, pp. 16–23.
- [47] A. Nagai, The recent achievements of computer Tsume-Shogi: Challenges in solving problems with extremely long steps, *J. Computer Shogi Assoc.* 13 (2000) 34–42 (in Japanese).
- [48] H. Nakaie, Y. Kotani, H. Iida, A method of applying opening book data to non-recorded positions, in: *Proceedings of Game Programming Workshop in Japan '96, Hakone, Japan, 1996*, pp. 218–227 (in Japanese).
- [49] H. Nakaie, Y. Kotani, A method of applying opening book data by partial matching, in: *Proceedings of Game Programming Workshop in Japan '97, Hakone, Japan, 1997*, pp. 106–113 (in Japanese).
- [50] A. Newell, C. Shaw, H.A. Simon, Chess playing programs and the problem of complexity, *IBM J. Res. Develop.* 2 (1958) 320–335.
- [51] A. Newell, H.A. Simon, *Human Problem Solving*, Prentice-Hall, Englewood Cliffs, NJ, 1972.
- [52] K. Noshita, A note on algorithmic generation of tsume-shogi problems, in: H. Matsubara (Ed.), *Proceedings of Game Programming Workshop in Japan '99, Hakone, Japan, 1996*, pp. 27–33.

- [53] J. Rollason, SUPER-SOMA—Solving tactical exchanges in Shogi without tree searching, in: I. Frank, T.A. Marsland (Eds.), *Proceedings of Second International Conference on Computers and Games, CG2000*, Lecture Notes in Computer Science, Vol. 2063, Springer, Heidelberg, 2001, pp. 291–310.
- [54] M. Sakuta, H. Iida, Report on the 9th CSA World Computer-Shogi Championship, *ICCA J.* 22 (3) (1999) 180–182.
- [55] A. Samuel, Some studies in machine learning using the game of checkers, *IBM J. Res. Develop.* 3 (1959) 210–229.
- [56] M. Seo, H. Iida, J.W.H.M. Uiterwijk, The PN*-search algorithm: Application to Tsume-Shogi, *Artificial Intelligence* 129 (1–2) (2001) 253–277.
- [57] C.E. Shannon, Programming a computer for playing chess, *Philos. Magazine* 41 (1950) 256–275.
- [58] D.J. Slate, L.R. Atkin, CHESS 4.5—The Northwestern University chess program, in: P.W. Frey (Ed.), *Chess Skill in Man and Machine*, Springer, Berlin, 1977, pp. 82–118.
- [59] T. Takizawa, Report on Computer-Shogi Grand Prix in 1999, *J. Computer Shogi Assoc.* 12 (1999) 39–41 (in Japanese).
- [60] T. Takizawa, R. Grimbergen, Review: Computer Shogi through 2000, in: I. Frank, T.A. Marsland (Eds.), *Proceedings of Second International Conference on Computers and Games, CG2000*, Lecture Notes in Computer Science, Vol. 2063, Springer, Heidelberg, 2001, pp. 456–466.
- [61] Y. Tanase, The algorithms in IS Shogi, in: H. Matsubara (Ed.), *Advances in Computer Shogi*, Vol. 3, Kyoritsu Shuppan, Tokyo, 2000, Chapter 1, pp. 1–14 (in Japanese).
- [62] J.W.H.M. Uiterwijk, H.J. van den Herik, Speculative play in computer chess, in: H.J. van den Herik, I.S. Herschberg, J.W.H.M. Uiterwijk (Eds.), *Advances in Computer Chess*, Vol. 7, University of Limburg, Maastricht, 1994, pp. 79–90.
- [63] H. Watanabe, H. Iida, J.W.H.M. Uiterwijk, Automatic composition of Shogi mating problems, in: H.J. van den Herik, H. Iida (Eds.), *Games in AI Research*, Van Spijk, Venlo, Netherlands, 2000, pp. 109–123.
- [64] G. Xinbo, H. Iida, J.W.H.M. Uiterwijk, H.J. van den Herik, Strategies anticipating a difference in search depth using opponent-model search, *Theoret. Comput. Sci.* 252 (1–2) (2001) 83–104.
- [65] H. Yamashita, A report on two shogi programs participating in an open rating competition, *J. Computer Shogi Assoc.* 10 (1997) 3–7 (in Japanese).
- [66] H. Yamashita, Half-extension algorithm, in: H. Matsubara (Ed.), *Proceedings of Game Programming Workshop in Japan '97*, Hakone, Japan, 1997, pp. 46–54.
- [67] H. Yamashita, YSS—About the data structures and the algorithms, in: H. Matsubara (Ed.), *Advances in Computer Shogi*, Vol. 2, Kyoritsu Shuppan, Tokyo, 1998, Chapter 6, pp. 112–142 (in Japanese).
- [68] H. Yamashita, YSS 7.0—About the data structures and the algorithm, http://plaza15.mbn.or.jp/~yss/book_e.html.