Strategy Acquisition for Games Based on the Simplified Reinforcement Learning Using Strategy Network

Masaaki KANAKUBO* and Masafumi HAGIWARA**

*Tokyo University of Technology, 1404-1 katakura-cho, Hachioji-city, Tokyo 192-0982, Japan

**Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama 223-8522, Japan .

email:kanakubo@cs.teu.ac.jp, hagiwara@soft.ics.keio.ac.jp

Abstract: In this paper, we propose a simplified reinforcement learning (RL) for game strategy acquisition using strategy network. RL has been applied to games such as backgammon, checker and others. However, RL applications to Othello or Shogi, which have a huge state space, are considered to be more difficult, because they take very long time to play. The proposed strategy network is composed of N lines from N nodes on the game board to only one evaluation node as 2-layers perceptron. These nodes denote all possible states of every square on the game board. It can easily represent the evaluation function. Moreover, these nodes can also denote imaginary states like pieces that may exist at the next step, or denote every position relation of arbitrary two pieces or other various board phases. After several thousands of games had been played, the strategy network quickly acquired a better evaluation function than the one using normalized Gaussian network. The computer player employing the strategy network becomes to beat a heuristic-based player that evaluates the values of pieces or places on the game boards. In addition, as for 4×4 Othello after co-evolutionary training, the player employing the strategy network acquired winning strategy in our Othello task.

Ⅰ. INTRODUCTION

Min-max method is the most popular among the search techniques for game tree. If all of the states in a game are known, an optimal move can be selected by the min-max method. Since the number of game states is huge, it is extremely difficult to find the optimal move based on an exhaustive search. However, if an arbitrary state can be evaluated correctly, the selected move is equivalent to the optimal one based on min-max search. In this case, the best move could be obtained without search. In each case, evaluation accuracy of a game state determines the strength of the game programming[1).

However, it is difficult to find the good evaluation function. Therefore, automatical acquisition of a good function by machine learning is desirable[2-4). Samuel made a famous checker program based on analogy of evolution as a pioneering attempt[5). Tesauro's backgammon program based on the reinforcement learning is also effective[6). These programs might be stronger than human for the games such as checkers and backgammon.

The numbers of possible states in such games are presumed to be roughly 10^{20} and 10^{30}, respectively. However, reinforcement learning is considered to be inappropriate for the games having larger number of states such as Othello, chess, and Shogi (Japanese chess)[7). The reason is that the convergence of reinforcement learning requires extremely long time. In such large-scale games, several years might be necessary for learning[3),8).

In applications of the conventional reinforcement learning to games, limited features of the board state are used as input elements to the networks; in the research on Shogi[9),10), for instance, values of each piece called *koma*, values of each possession *koma*, safety of a king, and so on, can be used. However, each value of *koma* changes according to the relation with many other *komas*. The safety of a king is evaluated by not only the states of the squares near the king but also the states of the squares that are apart from the king.

In this paper, we propose a strategy network with pseudo-reinforcement learning. It can be applied to various kinds of games. Fundamental strategy network is composed of N lines that connect many nodes on the board and one special node. The many nodes show all of the possible states in each square on the board. The special node has an evaluation value of the game state. The construction can be changed according to the input elements to the network. For instance, new nodes that correspond to *kiki*, which means legal move of each *koma* at the next step, may be put on each square. In addition, important elements can be estimated by using simple reinforcement learning.

Construction of a strategy network is like a perceptron having two layers. Our experimental results have shown that a player using strategy network is stronger than a player using heuristic strategy and a player using current reinforcement learning[11-13] with higher winning rate. Since the algorithm is very simple, it has an advantage with respect to efficiency of strategy acquisition. Moreover, as for Othello, 4×4 it has shown that strategy network acquired winning strategy in our Othello task.

Ⅱ. OUTLINE OF STRATEGY NETWORK

A. Basic structure for Othello

A lot of nodes on the board correspond to all possible states of each square. The proposed strategy network is composed of the lines that connect these nodes with one evaluation node. Fig.1 shows a basic structure for 8×8 Othello. There are 188 nodes that correspond to white, black and empty. Four discs are put at the center of the board beforehand according to Othello's rule. The strategy network is trained by updating the weights of the 188 lines as network parameters. A certain game state is evaluated by simple summation of the weights of lines that connect nodes denoting existing discs and the evaluation node.

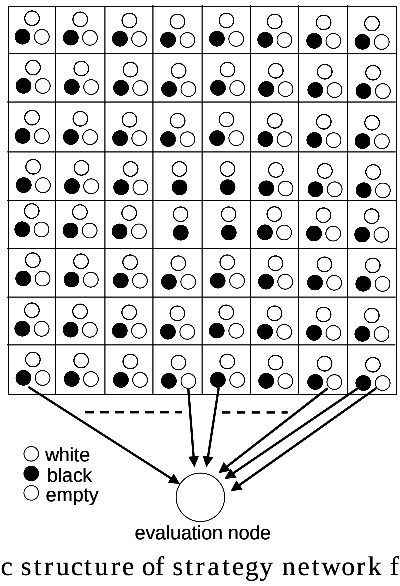


Fig.1 Basic structure of strategy network for Othello.

B. A variation for Othello

Fig.2 shows one variation of strategy network for Othello. Positional relations of arbitrary two pieces are also important to acquire stronger strategy. Therefore, we introduce new nodes corresponding to arbitrary combination of two nodes. In this case, strategy network is composed of lines connecting these new nodes and one evaluation node. On the upper Othello board in Fig.2, there are $_{188}C_2$ lines corresponding to arbitrary combination of two nodes, and these lines are connected to the nodes on the lower Othello board in Fig.2.

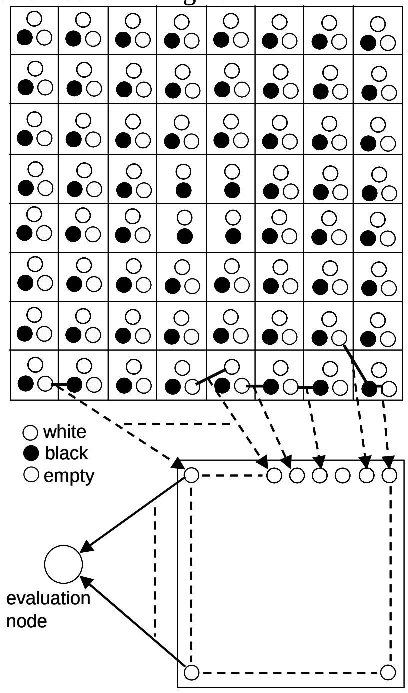


Fig.2 A variation for Othello.

C. Strategy network for Shogi

Fig.3 shows basic structure of strategy network for Shogi (Japanese chess). In Shogi game, 14 kinds of pieces are used. And there are many *nari-komas* which mean upgraded pieces by passing the enemy's line. Considering pieces of ally and enemy, there are 29 kinds of pieces in total. On the Shogi board there are 81 squares. Consequently, there are 81×29 nodes on the Shogi board. The basic structure of strategy network for Shogi is composed of lines connecting these nodes and one evaluation node.

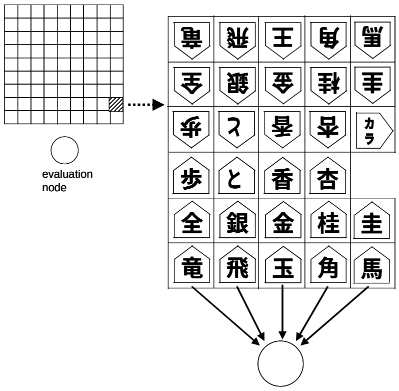


Fig.3 Basic structure of strategy network for Shogi.

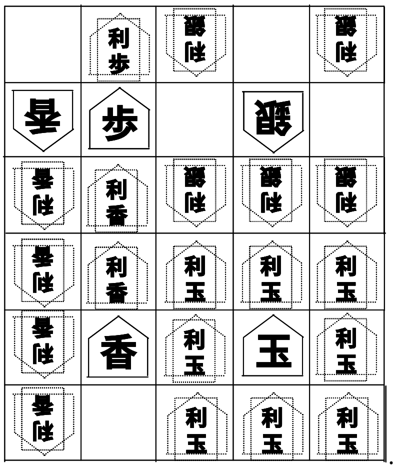


Fig.4 Examples of *kiki-koma*.

To evaluate a certain game state, *kiki*, which means the legal moves of pieces at the next step, is important. Then, we devised the following variation of strategy network. We introduced imaginary pieces called *kiki-koma. Kiki-komas* correspond to the legal moves of pieces. There are many *kiki-komas* generated by one piece. Fig.4 shows examples of *kiki-koma*. In this case, only four kinds of pieces exist actually on the board. Many *kiki-komas* exist in squares close to the actual pieces. Two or more *kiki-koma* might exist in one square. According to the rule, some of these *kiki-komas* will become invalid by the next move. In this case, there are many nodes corresponding to actual pieces and *kiki-komas*, and the strategy network is composed of lines connecting these nodes and one evaluation node.

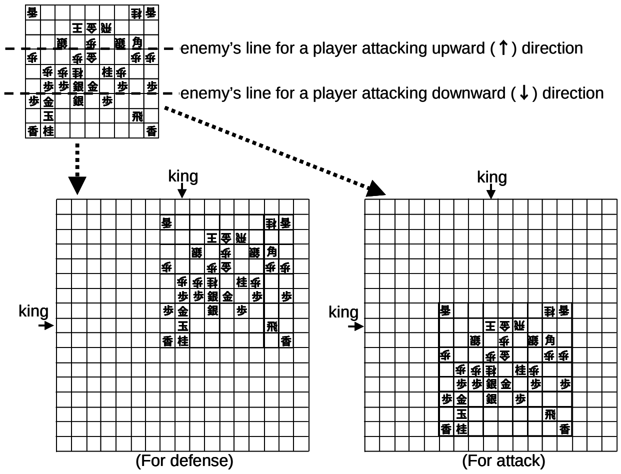

enemy's line for a player attacking upward (↑) direction
enemy's line for a player attacking downward (↓) direction
king
king
king
king
(For defense)
(For attack)

Fig.5 Strategy board for Shogi.

If a king is captured, then Shogi game ends. We devised a strategy board: it is generated by shifting Shogi board. Fig.5 shows an example. A strategy board has 17×17 squares and ally's king or enemy's king always exists at the center of the board. On the strategy board for attack, enemy's king always exists at the center. On the strategy board for defense, ally's king always exists at the center. The reinforcement learning on king's surrounding can be continued regardless of the king's position, and can acquire high generalization ability. In this case, there are 17×17×29=8,381 nodes corresponding to various pieces in each square of the strategy board.

D．Training scheme

The training scheme of simplified reinforcement learning based on strategy network is as follows: During a game, every appeared state is preserved. If the training player wins the game, the values of every line of the strategy network linking the nodes used in the preserved states are increased by one point. If the training player loses the game, the values are decreased. This decreased value is current total number of wins divided by current total number of defeats. If the total number of defeats is zero, this decreased value is current total number. The reason is that if the training player is beaten by a strong opponent, the weights of the network used in the game should be lowered relatively. If the game is a tie, it doesn't change the weights. At the beginning of training, all weights of lines are set to 0.

III. EXPERIMENT

In the experiments, first, we compared strategy network with the current reinforcement learning in Othello task. Next, we tested the effectiveness of variations of strategy network in Shogi task. In both tests, the enemies had easy heuristic strategies such as greedy strategy or the strategy using numbers of each square. A player played 10,000 games in one experiment. All of the following results are mean value of ten experiments. In order to avoid player's deterministic move, if plural best moves having the highest ratings existed, players chose one move at random. Moreover, we considered a strategy network as one chromosome, and tried co-evolution by playing games between chromosomes.

A．Othello's experiments

Using variation of strategy network mentioned in Sec.2, simplified reinforcement learning in 8×8 Othello was carried out. We used two kinds of enemies. Enemies had heuristic strategies and were often used in current researches[11-13]. The first one had the following strategy.

- After the number of empty squares decreases to 6, the exhaustive search is used.
- Before then, a greedy strategy is used, that is, to select a move that maximizes the number of ally discs. However, if a disc can be put at a corner square, this move is selected with priority.

The enemy having the above strategy was denoted by HP1 (heuristic player 1).

The second enemy had a strategy which selects the next move according to the table shown in Fig.6. If there was an ally in one square, then the square's number was added to the evaluation value. If there was an enemy, the square's number was subtracted from the evaluation value. If empty, the evaluation value was not changed. For Othello game, getting corners is the most advantageous strategy. Therefore, four corners had the highest number. If there is an ally in the squares next to corner, the enemy easily gets the corner. So, these squares had the lowest number. Such evaluation method based on square's numbers was often used in the current Othello-game programming[14]. This enemy having this strategy was denoted by HP2.

100	-25	10	5	5	10	-25	100
-25	-25	2	2	2	2	-25	-25
10	2	5	1	1	5	2	10
5	2	1	2	2	1	2	5
5	2	1	2	2	1	2	5
10	2	5	1	1	5	2	10
-25	-25	2	2	2	2	-25	-25
100	-25	10	5	5	10	-25	100

Fig.6 Numbers of each squares of Othello game board.

B. Results of experiments

In Fig.7, the abscissa and the ordinate correspond to the game number and winning rates against HP1, respectively. After one thousand of games had been played, the winning rate increased to 74.0 %(using black disc as the first mover) or 77.1 %(using white disc as the second mover). After ten thousands of games had been played, the winning rate increased to 81.4 %(black) or 83.0 %(white). In the current reinforcement learning[11-13], after ten thousands of games, the winning rate remained in about 60~70%.

Fig.8 shows the winning rates against HP2. After ten thousands of games had been played, the winning rate increased to 86.6 %(black) or 64.7 %(white). In the current reinforcement learning[11-13], after ten thousands of games, the winning rate remained in about 40%.

C. Shogi's experiments

The experiments were carried out to evaluate the ability of strategy network to select the suitable input elements. We used the following four strategy networks.

- Based on normal Shogi board and usually kinds of pieces (SN1)
- Based on normal Shogi board and usually pieces and *kiki-komas* (SN2)

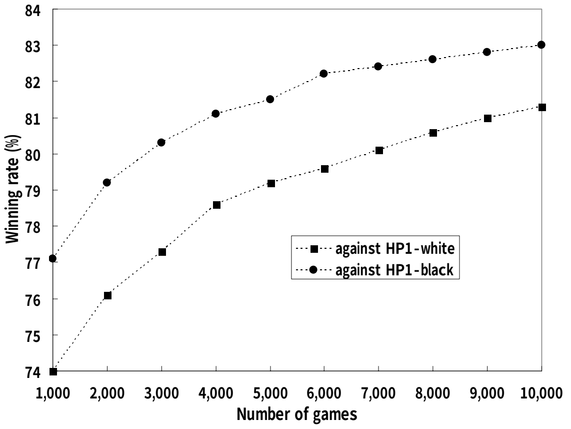


Fig.7 Average winning rates of players using strategy network against HP1.

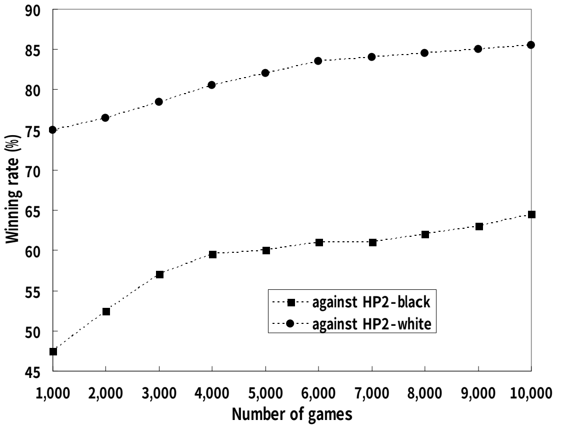


Fig.8 Average winning rates of players using strategy network against HP2.

- Based on strategy board and normal pieces (SN3)
- Based on strategy board and normal pieces and *kiki-komas* (SN4)

Each strategy network played games against the enemies having heuristic strategy. In the current research, the strategy evaluating board states by total value of pieces on the board was often used[10),15-17]. In our experiments, the enemies had this strategy. Table 1 shows typical values of pieces used in our experiments[18].

Pieces	Points	Pieces	Points
飛 (hisha)	15	龍 (narihisha)	17
角 (kaku)	13	馬 (narikaku)	15
金 (kin)	9		
銀 (gin)	8	全 (narigin)	9
桂 (keima)	6	圭 (narikei)	10
香 (yari)	5	杏 (nariyari)	10
歩 (fu)	1	と (narifu)	12

Tab.1 Values of pieces of Shogi.

D. Results of experiments

Fig.9 shows the winning rates against the enemies based on SN1, SN3 and SN4, respectively. After one thousand of games had been played, the winning rate increased to 90.0% or more in every case. The reason is that in Shogi since the board state tends to change more gradually than Othello, reinforcement learning is more effective. The player based on strategy network played as the first mover(*sente*) and the second mover(*koute*) in turn, in every case, the order of winning rates was SN1<SN3<SN4.

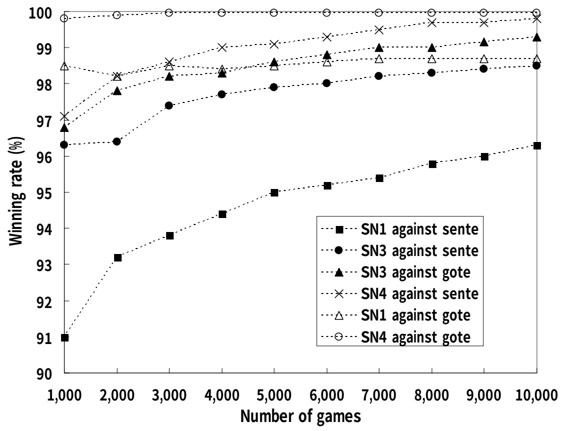


Fig.9 Average winning rates of players using SN1 or SN3 or SN4.

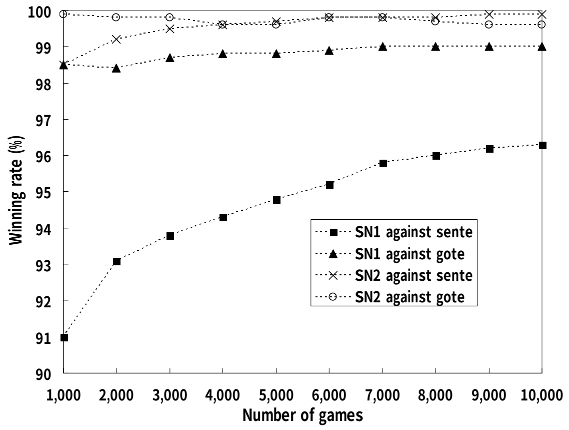


Fig.10 Average winning rates of players using SN1 or SN2.

Fig.10 shows the winning rates against the enemies based on SN1 and SN2, respectively. It has shown that *kiki-komas* and strategy board are both useful; especially the later is extremely effective. It has also shown that the strategy network is effective to select features on the board to construct input layer of neural network.

E. *Co-evolution*

It has shown that in 4×4 Othello the second mover has the winning strategy by using exhaustive search[2]. In our experiments, there were 100 first movers and 100 second movers. Every player had strategy network. We considered a set of weights of lines in one strategy network as one chromosome of Genetic Algorithms (GA). Every player played games against every enemy. According to the winning rate, chromosomes were selected and mutated. At each generation, the best strategy network fought against two enemies, one of them had an easy heuristic strategy, and another had an exhaustive search strategy.

In the experiments, the initial weights were set at the numbers generated at random in the following either range, that is, 0~10, 0~100, 0~1,000 and 0~10,000. These generated numbers were used also for the mutation. As a result, the evolution of strategy network became more dynamic, and the premature of evolution could be avoided. The fitness value of each chromosome was its own winning rate. The method of selection was roulette wheel selection, the mutation rate was 5.0% and elitism strategy was used.

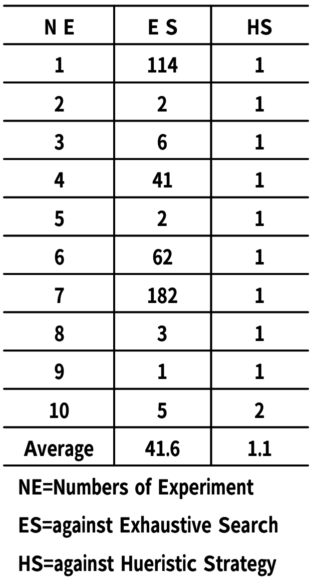
4	-1	-1	4
-1	-1	-1	-1
-1	-1	-1	-1
4	-1	-1	4

Fig.11 Numbers of each square of 4×4 Othello board.

The easy heuristic strategy evaluated a board state according to total value of each square's numbers shown in Fig.11. If there was an ally in one square, then the square's number was added to the evaluation value. If there was an enemy, the square's number was subtracted from evaluation value. If empty, the evaluation value was not changed. Table 2 shows the generation numbers at which strategy network won the enemies for the first time. Against the enemy having heuristic strategy, the enemy was beaten by the strongest strategy network almost at the first generation. Against the enemy having exhaustive search strategy, on the average, at the 40.6 generations the enemy was beaten. It has shown that the co-evolution of strategy network can acquire a suitable winning strategy.

Ⅳ. CONCLUSION

In this paper, we proposed simplified reinforcement learning for game strategy acquisition using strategy network. Strategy network is simple: it is composed of N lines from N nodes on the game board to only one evaluation node. These nodes correspond to all possible states of every square on the game board. Moreover, the nodes of strategy network also can correspond to various features on the game board, for instance, imaginary pieces corresponding to all of the legal move of pieces or positional relation of arbitrary two pieces. Since the strategy network can test easily various input elements to the network, it acquires efficiency, and give useful input elements to other complex neural networks as function approximaters.



N E	E S	HS
1	114	1
2	2	1
3	6	1
4	41	1
5	2	1
6	62	1
7	182	1
8	3	1
9	1	1
10	5	2
Average	41.6	1.1

NE=Numbers of Experiment
ES=against Exhaustive Search
HS=against Hueristic Strategy

Tab.2 Numbers of generation at which the strategy network won the enemies for the first time.

In the experiments in Othello, after ten thousands of games had been played, the winning rates of the player using strategy network against the player using heuristic strategy increased to 64.7%-86.6%. In the current reinforcement learning, on the same condition, the winning rates of the player using normalized Gaussian network remained in about 40.0-70.0%. In the experiments in Shogi, after ten thousands of games had been played, the same winning rates increased to over 95.0%, and the usefulness of various features on the game board has cleared by using strategy network. In addition, after co-evolutionary training in 4×4 Othello, the player employing strategy network acquired the winning starategy. RL applications to Othello or Shogi, which have a huge state space, are considered to be more difficult, because they take extremely long time. The time will be shortened extremely by using strategy network. Since it can easily test various RL conditions, it will increase the possibility of RL for games.

REFERENCES

[1] H. Matsubara, “Why Can Computers Play Complex Games so Well?”, Journal of the Japan Society of Mechanical Engineers, Vol.100, No.949, pp.1246-1247 (1997) (in Japanese).

[2] H. Matsubara, “Shogi and Computer”, Kyoritsu Press., (1994) (in Japanese).

[3] H. Matsubara, “Recent Progresses on Game Programming Researches”, Journal of Japanese Society for Artificial Intelligence, Vol.10, No.6, pp.835-845 (1995) (in Japanese).

[4] T. Kaneko, K. Yamaguchi and S. Kawai, “Automatic Construction of Pattern-based Evaluation Functions for Game Programming”, Transactions of Information Processing Society of Japan, Vol. 43, No.10, pp.3040-3047 (2002) (in Japanese).

[5] Samuel, A. L., “Some Studies in Machine Learning using the Game of Checkers”, IBM J. Res. Dev., Vol.3, pp.210-229 (1959).

[6] Tesauro, G., “Practical Issues in Temporal Difference Learning”, Machine Learning, Vol.8, pp.257-277 (1992).

[7] H. Matsubara, “What Can AI Researchers Learn from Deep Blue’s Victory?”, Journal of Japanese Society for Artificial Intelligence, Vol.12, No.5, pp.698-703 (1997) (in Japanese).

[8] H. Matsubara and T. Takizawa, “How Shogi Programs Become Such Strong As Amateur 4-dan”, Journal of Japanese Society for Artificial Intelligence, Vol.16, No.3, pp.379-384 (2001) (in Japanese).

[9] D. F. Beal and M. C. Smith, “First Results from using Temporal Difference Learning in Shogi”, Computers and Games, pp.113-125 (1998).

[10] K. Usui, T. Suzuki and Y. Kotani, “Parameter Learning using Temporal Differences in Shogi”, Proc. of Symposium of Information Processing Society of Japan, Vol.99, No.14, pp.31-38 (1999) (in Japanese).

[11] T. Yoshioka, S. Ishii and M. Ito, “Strategy Acquisition for the Game “Othello” Based on Reinforcement Learning”, IEICE Trans. Inf. & Syst., Vol.E82-D, No.12, pp.1618-1626 (1999).

[12] T. Yoshioka, S. Ishii and M. Ito, “Strategy Acquisition for the “Othello” game based on the reinforcement learning”, Proc. of Foundations of Artificial Intelligence, SIG-FAI-9703-17, pp.115-120 (1998) (in Japanese).

[13] T. Yoshioka, S. Ishii, “Learning of an evaluation function of the game Othello by EM algorithm”, IEICE Technical Report, NC98-41 (1998) (in Japanese).

[14] M. Buro, “The strongest program for Othello based on machine learning, logistello”, bit, Vol.32, No.7, pp.46-50 (2000) (in Japanese).

[15] N. Sasaki and H. Iida, “The Study of Evolutionary Change of Shogi”, Transactions of Information Processing Society of Japan, Vol. 43, No.10, pp.2990-2997 (2002) (in Japanese).

[16] N. Sasaki, N. Takeshita, T. Hashimoto and H. Iida, “Decision-Complexity Estimate in Evolutionary Changes of Games”, Proc. of Symposium of Information Processing Society of Japan, Vol.2001, No.14, pp.140-147 (2001) (in Japanese).

[17] M. Taketoshi, T. Hashimoto, M. Sakuta and H. Iida, “Search Efficiency by Move Ordering Techniques in Computer Shogi”, Transactions of Information Processing Society of Japan, Vol. 43, No.10, pp.3074-3077 (2002) (in Japanese).

[18] K. Tanigawa, “The thinking way for playing Shogi game”, Ikeda-Shoten, (1982).