

# Evolutionary Othello Players Boosted by Opening Knowledge

Kyung-Joong Kim and Sung-Bae Cho  
Dept. of Computer Science, Yonsei University  
134 Shinchon-dong, Sudaemoon-ku, Seoul 120-749, Korea  
kjkim@cs.yonsei.ac.kr, sbcho@cs.yonsei.ac.kr

**Abstract**— The evolutionary approach for gaming is different from the traditional one that exploits knowledge of the opening, middle, and endgame stages. It is therefore sometimes inefficient to evolve simple heuristics that may be created easily by humans because it is based purely on a bottom-up style of construction. Incorporating domain knowledge into evolutionary computation can improve the performance of evolved strategies and accelerate the speed of evolution by reducing the search space. In this paper, we develop an evolutionary Othello player with the systematic insertion of opening knowledge into the framework of evolution. The probability of opening selection is coming from the expert’s opening list. Preliminary experimental results show that the proposed method is promising for generating better strategies for Othello players.

## I. INTRODUCTION

In Othello, the typical approach uses a computer program to search a game tree to find an optimal move at each play, but there are challenges in overcoming an expert’s experience in the opening, middle, and endgame stages. Sometimes a computer Othello program fails to defeat a human player because it makes a mistake that is not common among human players. Sometimes the fault is discovered by the computer program after searching beyond the predefined depth of the game tree (a so-called “horizon effect”). To defeat the best human players, Logistello (the best computer Othello program) uses an opening book and most importantly the endgame solver [17][16]. Also, Logistello relies on expert knowledge that is captured in an evaluation function, which is used in the middle stage. Logistello’s success is based largely on traditional game theory mechanics (game tree and alpha-beta search) and expert knowledge (opening book, components in evaluation function, and endgame database).

Recently, evolutionary induction of game strategies has gained popularity because of the success reported in [20] using the game of checkers. This approach does not need additional prior knowledge or expert heuristics for evolving strategies and expert-level strategies have evolved from the process of self-play, variation, and selection. In other games such as Othello [4][5], blackjack [21], Go [22], chess [23] and backgammon [10], the evolutionary approach has been applied to discover better strategies either without relying on human experience or with very little reliance. Usually, opening knowledge and endgame databases are not involved

in the evolutionary process because a researcher wants to investigate the possibility of pure evolution [20]. However, it might take a long evolution time to create a world-level champion program without a predefined knowledge-base. It took 6 months (using a PII machine) to evolve the checkers program rated at the expert level by Fogel and Chellapilla [25] and it would take even longer time to evolve the world-level champion program.

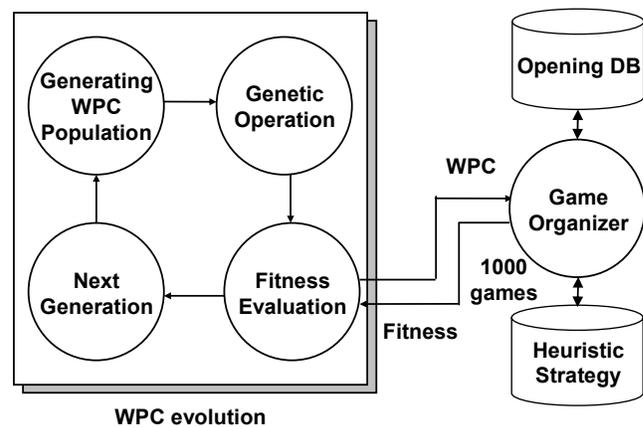


Figure 1: Conceptual diagram of the proposed method. Game organizer decides the usage of the opening DB.

Incorporating a priori knowledge, such as expert knowledge, meta-heuristics, human preferences, and most importantly domain knowledge discovered during evolutionary search, into evolutionary algorithms has gained increasing interest in recent years [26]. Kim *et al.* proposed a method that systematically incorporates domain-specific knowledge into a speciated checkers players [27]. We could confirm the usefulness of domain knowledge in the course of evolutionary approach for game strategies in terms of opening stage.

In this paper, we propose a method for inserting expert knowledge into an evolutionary Othello framework at the opening stage. In the opening stage, openings defined by a number of experts are used. The selection of opening is based on the probability of the each move in the list of openings defined by experts. Figure 1 explains the conceptual framework of the proposed method. Weight Piece Counter (WPC) which represents the weights of each board positions is used as a representation of strategy. Though it is very simple, it is still strong compared to other representations if it is well adjusted. Standard heuristic strategy represented with

the WPC format is used as a static opponent for fitness evaluation.

## II. BACKGROUNDS

### A. Othello

Othello has a very simple rule but it is very difficult to master the complex strategies for human [15]. Othello is traditionally played on an eight-by-eight board (see Figure 2) and there are two players (“black” and “white”). Initially, four discs (two white and two black) are placed in the board. The first move must be done by black player. The rule of game is very simple. Moves that are capable of sandwiching opponent’s discs in one or more horizontal, diagonal, and vertical directions are valid. The sandwiched discs are flipped (usually, Othello disc has two sides (white and black)). In the initial position, there are four choices for black (C4, D3, E6 and F5; it is standard notation of Othello board; each capital letter represents the row and number for column). In the figure, the result is shown after black chooses E6. The sandwiched disc at the E5 is flipped to black disc. After then, the next turn is white.

Because there is 64 squares in the board (4 squares are occupied), each player can moves 30 times if there is no pass, which means that there is no available moves and the turn is automatically changed. If there is no available move for the both players, the game is ended. The number of final discs is counted for each player and the winner is the one with more discs. If the number of disc is the same, the game is draw (32-32).

The complexity of Othello is harder than checkers, yet simpler than chess [13]. Logitello, one of the strongest Othello program in the world, already beat the human World Champion at 1997 [18].

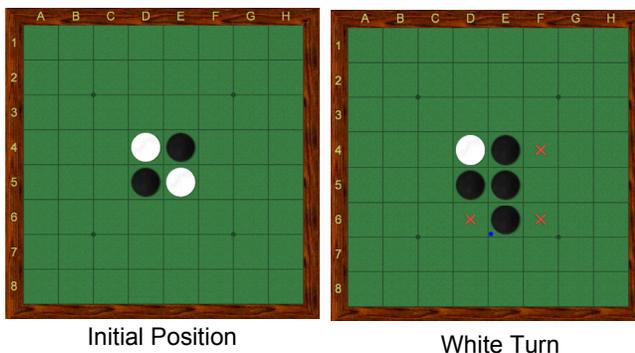


Figure 2: Initial position of Othello board. Black moves first.

### B. Traditional Game Programming

Usually a game can be divided into three general phases: the opening, the middle game, and the endgame. Entering thousands of positions in published books into the program is a way of creating an opening book. A problem with this approach is that the program will follow published play, which is usually familiar to the humans. Without using an

opening book, some programs find many interesting opening moves that stymie a human quickly. However, they can also produce fatal mistakes and enter a losing configuration quickly because a deeper search would have been necessary to avoid the mistake. Humans have an advantage over computers in the opening stage because it is difficult to quantify the relevance of the board configuration at an early stage. To be more competitive, an opening book can be very helpful but a huge opening book can make the program inflexible and without novelty.

One of important parts of game programming is to design the evaluation function for the middle stage of the game. The evaluation function is often a linear combination of features based on human knowledge, such as the number of discs, the number of legal moves, the piece differential between two players, and pattern-based features. Determining components and weighting them requires expert knowledge and a long trial-and-error tuning. Attempts have been made to tune the weights of the evaluation function through automated processes, by using linear equations, neural nets, and evolutionary algorithms and can compete with hand-tuning.

In Othello, the final outcome of most games is usually decided before the endgame and the impact of an endgame solver is particularly significant. In Othello, the results of the game can be calculated in real-time if the number of empty spaces is less than 26.

### C. Othello and Evolution

Because the strategy of Othello is very complex and hard to study even for human, researchers use the game as a platform of AI research. Miikkulainen *et al.* used evolutionary neural network as a static evaluator of the board [3]. Because they used marker-based encoding for neural network, the architecture and weights were coevolving. In their work, they did not use game tree but the evolution generated mobility strategy, one of the important human strategies for the game. The analysis by the world champion Othello player, David Shaman, was attached and the result showed the possibility of evolutionary approach for the game. In other works, they used evolutionary neural network to focus the search of game tree [1][2]. In the work, game tree was used to find the good move with static evaluator. The purpose of the evolution was to find a neural network that decided whether deep search was needed. By reducing the moves for deep search, it was possible to expand the search depth more than previous one.

Chong *et al.* used the same framework used in successful evolutionary checkers to evolve Othello strategies [4][5]. Their work showed that the same architecture could be successful in another game (Othello). They discussed that the reasons of success were spatial preprocessing layer, self-adaptive mutation, and tournament-style selection (coevolving). They evolved only the weights of fixed-architecture neural network and the concept of spatial preprocessing (dividing the board into a number of sub-boards) were used.

Sun *et al.* proposed dynamic fitness function for the evolution of weights of linear static evaluator for Othello [6]. The fitness function was changed according to the performance of the previous stage. They also investigated the expansion of chromosome structures to meet new challenges from the environment in the game of Othello [7]. In other work, they used multiple coaches (they were used for fitness evaluation) selected from local computer Othello tournament before evolution [8]. Sun's work was focused on increasing the diversity and self-adaptation of evolutionary algorithm given linear evaluation function. The purpose of evolution was adjusting the weights of features (position, piece advantage, mobility and stability) by expert.

Alliot *et al.* proposed a method for improving the evolution of game strategies [9]. They evolved weights of linear evaluation function (it was similar to the weight piece counter). The sharing scheme and sophisticated method for fitness evaluation were used. Because Othello's rule was so simple to understand, it was frequently used for educational purpose to teach student about the evolutionary mechanism [10][11]. Smith *et al.* proposed co-evolutionary approach for Othello [12]. The fitness of the member in the population was evaluated by the competition results with other members. There were some works about reinforcement learning or self-teaching neural networks for Othello [13][14]. The discussion about the relationships between reinforcement learning and the evolutionary algorithm can be found at [19].

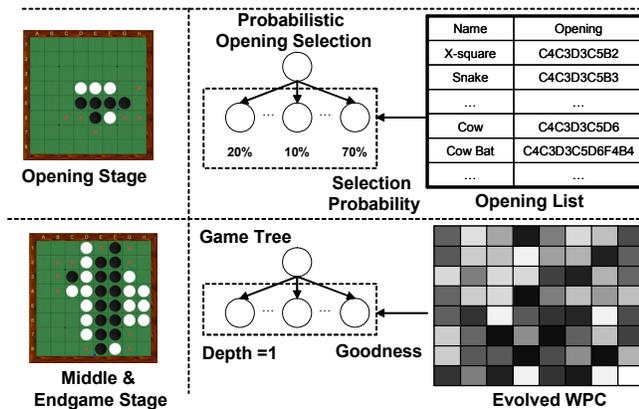


Figure 3: The flow of the game using the proposed method. In each WPC, the darkness represents the strength of the weight.

### III. INCORPORATING KNOWLEDGE INTO EVOLUTIONARY OTHELLO

As mentioned before, we have classified a single Othello game into three stages: opening, middle, and endgame stages. In the opening stage, about 99 previously summarized openings are used to determine the initial moves. In the middle and endgame stage, a weight piece counter is used to search for a good move and an evolutionary algorithm is used to optimize the weights of the counter. Though game tree is useful to find a good move, in this paper we only focus on the goodness of the evaluation function. Figure 3 shows the procedural flow of the proposed method in a game. The

longest opening has 18 moves and it can significantly reduce the difficulty of the evaluation function optimization.

#### A. Opening Stage

Opening is very important for Othello game because the game is extremely short compared to other games and the importance of early moves is huge. In the games between expert players, gaining small advantage in the early games is very important. They attempt to memorize good or tricky moves before tournament. Brian Rose, World Othello Champion at 2002, said that he memorized about 300 lines before the championships. It is critical to prepare and memorize well-defined openings in the Othello game.

In this work, well-defined 99 openings are used (<http://www.othello.nl/content/anim/openings.txt>). The length of opening ranges from 2 to 18. In fact, in the expert player's game, there is a case that all moves are memorized by both of them. The length of opening can be expanded to the last move, but it is not possible to exploit all of them. "Out-of-opening" means that the moves by the player are not in the list of the openings. Because it is carefully analyzed by experts, it can guarantee the equivalence between two players. In some openings, they are a bit beneficial to a specific color but it is not important because in human's game the difference can be recovered in the middle and end game.

The opening is named after their shaped. Tiger, Cat, and Chimney means that the shape of the opening is similar to the objects. Until out-of-opening, the WPC player follows one of the openings in the list. Figure 4 shows an example of opening selection. The current sequence of the game is defined as  $S = \{m_1, m_2, \dots, m_n\}$ .  $m_i$  represents each move.

$n$  is the number of moves played. For each opening, check whether the first  $n$  moves are the same with the sequence. The satisfactory openings are called candidates. Among candidates, one opening is chosen probabilistically. The next move of the selected opening after the first  $n$  moves is decided as the next move of player. Figure 5 shows the category of openings. At the 3<sup>rd</sup> move, there are 3 choices: Diagonal, perpendicular, parallel. For each choice, there are several extended openings. The probability of choice for each line is determined based on the number of openings. For example, the diagonal opening has 59 variations in his line and about 59% probability to be selected. If there are many variations for each choice, it means that the line is the most preferable one for humans. The probability of selection is based on the human's long investigation.

The probability of opening move selection can be determined using another way. For example, the probability of openings in the games of WTHOR database (It contains all public games played between humans) can be used. Or specific book evaluation (automatically learned from self-played games of strong program) values can be exploited. WZEBRA (strong Othello program) provides evaluation value for each opening. For example, the X-square opening is -23 (in black's perspective) and Snake is -4.

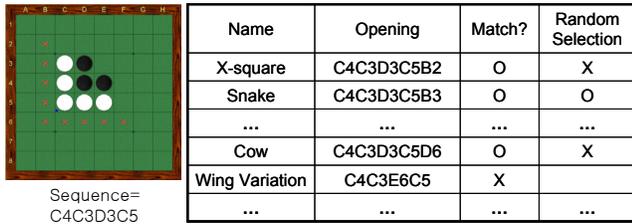


Figure 4: The selection of opening (Snake is selected)

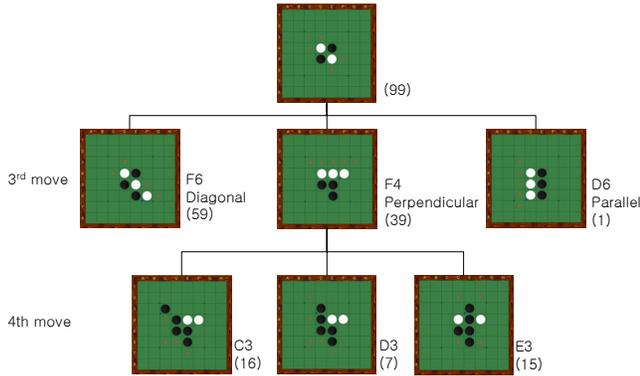


Figure 5: The categorization of openings and the number of openings for each category

B. Evolving Evaluator (WPC)

The board is represented as a vector of length 64. Black disc is represented as 1 and white disc as -1 (Figure 6). Empty space is 0. The relevance of the board is calculated using weighted piece counter. The weighted piece counter is a vector of length 64. Each element is corresponding to one square in the Othello board, which means the weights of the board position. The relevance of the board is calculated using dot product of two vectors. Figure 7 shows an example of move selection using WPC.

The evaluation of each WPC is based on the competition against static player. Standard heuristic WPC is used to evaluate the performance of the individual. Figure 8 shows the standard WPC. Its darkness represents the weight of each position. In Othello, four corners are extremely valuable and the weight of the corners is 1.0. Other places except the ones near the corner has the similar (0.5~0.55) weights. The positions near the corners are very dangerous because it gives a chance to the opponent to capture the corner. Because it is static, it cannot evaluate well the dynamics of the relevance of position but it is still strong compared to other random approaches.

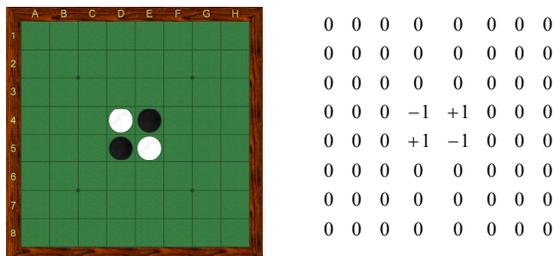


Figure 6: The representation of the board

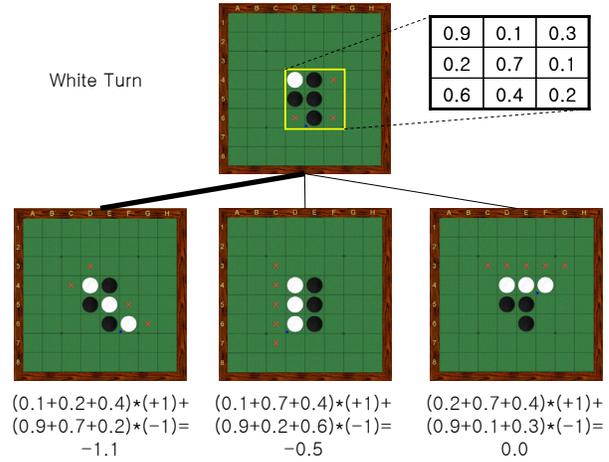


Figure 7: The selection of move using WPC (White will choose the -1.1 move)

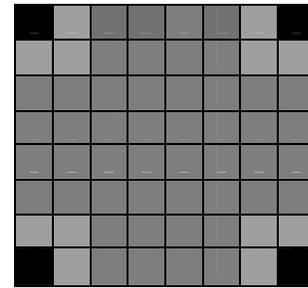


Figure 8: Weights of the heuristic (Darkness represents higher weight)

The fitness of the WPC is evaluated based on the 1000 games between the standard WPCs. Given two deterministic strategies, there can be only two games (changing the color). This makes the fitness evaluation difficult and random moves are used. 10% of moves of both players are decided randomly. It allows 1000 games among them are different but it also makes the fitness of each WPC unstable. To preserve good solutions for the next generation, elitist approach is used.

If the number of wins is  $W$ , the number of losses is  $L$ , and the number of draws is  $D$ , the fitness is defined as follows.

$$fitness = W * 1.0 + D * 0.5$$

The weights are widely used in real Othello tournament.

The weights of each position are initialized as a value between -1 to 1. Until out-of-opening, the WPC in the population uses opening knowledge. After out-of-opening, the WPC evaluates the relevance of board and decides the next move. Among possible moves, the one with the highest WPC is selected. Roulette-wheel selection is used and 1-point crossover is applied to the converted 1-dimensional array of the 8x8 board. Mutation operator changes an element of the vector as a new value ranging from -1 to 1. Figure 9 summarizes the algorithm of the proposed method.

```

1: /* Opening[] : Total 396 openings (99 openings in four
   directions) */
2: /* Sequence[] : The sequence of moves played */
3: /* WPC_H : The standard heuristic */
4: /* POP[]: Population of WPC */
5: /* Board[] : A vector representation of current board */
6: /* Fitness[] : Fitness of each individual */
7: /* Move(WPC): Select a move using the WPC (10%
   randomness) */
8:
9: Initialization(POP); // Population of WPC initialization
10:
11: FOR i=1 to MAX_GEN {
12:   FOR j=1 to POP_SIZE {
13:     FOR k=1 to NUM_OF_GAMES {
14:       Initialization(Sequence);
15:       Initialization(Board);
16:       // The color of disc for each player is randomly chosen.
17:       FOR m=1 to 60 {
18:         IF(MY_TURN && !OUT_OF_OPEN) {
19:           FOR n=1 to 396 {
20:             IF(Match(Sequence, Opening[n])) Count++;
21:           }
22:           Choose(Random(0,Count));
23:           // Choose a opening among candidates
   (matched)
24:         }
25:         ELSE IF (MY_TURN) { Move(POP[j]);}
26:         ELSE {Move(WPC_H);}
27:       }
28:     }
29:     Fitness[j]=WIN+DRAW*0.5;
30:   }
31:   Selection(); // Roulette-Wheel selection
32:   Crossover(); // 1-point crossover
33:   Mutation ();
34:   Elitist();
35: }

```

Figure 9: A pseudo code of the proposed method

#### IV. EXPERIMENTAL RESULTS

The parameters of the evolution are as follows. The maximum generation is 100, population size is 50, crossover rate is 0.8, and mutation rate is 0.01. Before deciding the representation of strategy, we have tested WPC and MLP (fixed architecture, 64 inputs, and 32 hidden neurons in one hidden-layer). The number of parameters for WPC is 64 but 2113 for neural networks. The genetic algorithm for MLP is the same with one of WPCs. Figure 10 shows the change of the maximum and average fitness of the population of WPC. It shows that the fitness is gradually increased until 21 generation and it converges. The average fitness is increasing continuously and performance has been improved. The maximum of fitness is about 450.

In the MLP evolution, the individual with maximum fitness emerges at the 51<sup>st</sup> generation (Figure 11). Both of the evolutions (Figure 10 and Figure 11) show the fluctuation of the maximum fitness because of the random move in the

competition games. It allows sudden decrease of fitness. Lucky individual with high fitness dominates the population and it shows low performance after the generation. Compared to WPC evolution, the evolution of MLP shows low performance. Its best fitness is about 400. It shows that the general genetic algorithm is not suitable for the evolution of MLP. It is better to adopt self-adaptive mutation and spatial preprocessing layer like [4] for evolving the weights of the fixed architecture. The increase of the average fitness for MLP evolution is smaller than the one in the WPC evolution.

Figure 12 shows the evolution of WPC with the population size 100. It shows that the performance is increased and the best fitness is about 500. Increasing the population size can be a solution for such kinds of fitness function with randomness but it can increase the computational cost. Also, it is possible to increase the number of games between the individual and the static WPC but it also decreases the evolution speed. The score 500 is about the 40<sup>th</sup> rank in the CEC Othello evolution competition homepage (<http://algoval.essex.ac.uk:8080/othello/html/Othello.html>). Though it is not the best one, it can be improved by increasing the maximum generation number. Also, incorporating spatial preprocessing layer and self-adaptive mutation can improve the performance. Because the focus of this paper is incorporating knowledge into evolution, they are not considered.

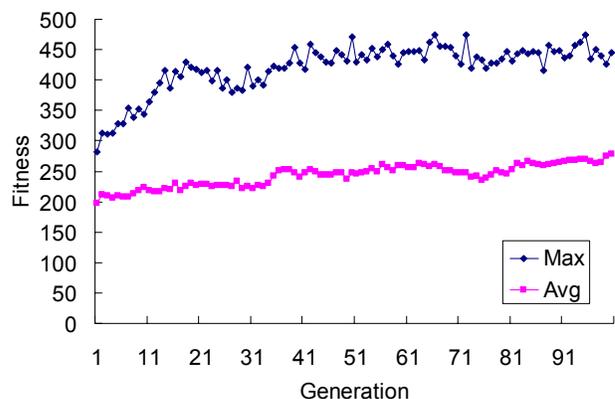


Figure 10: The evolution of weight piece counter (Population size is 50)

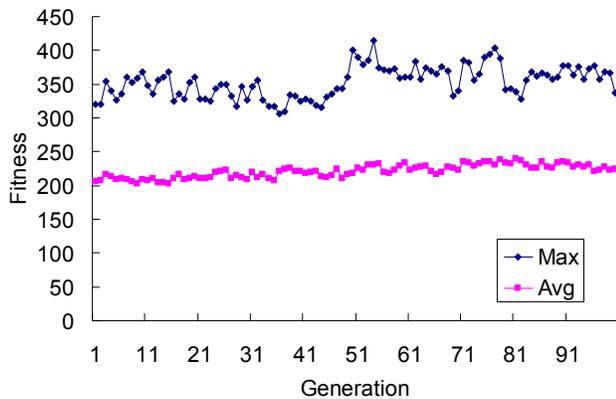


Figure 11: The evolution of fixed-architecture MLP (Population size is 50)

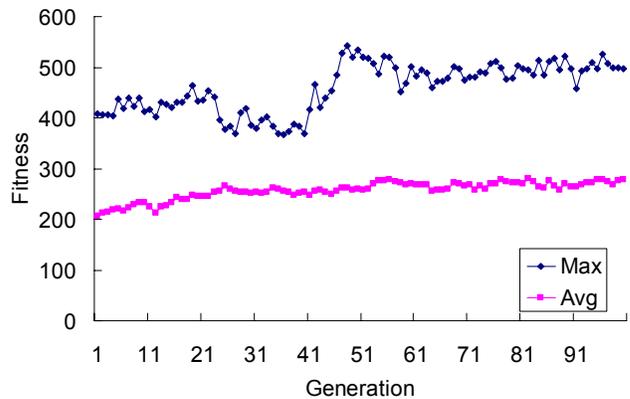


Figure 13: The evolution of weight piece counter with opening knowledge (Population size is 50)

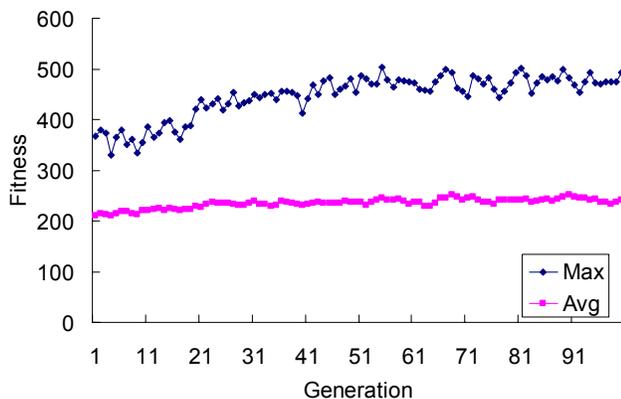


Figure 12: The evolution of weight piece counter (Population size is 100)

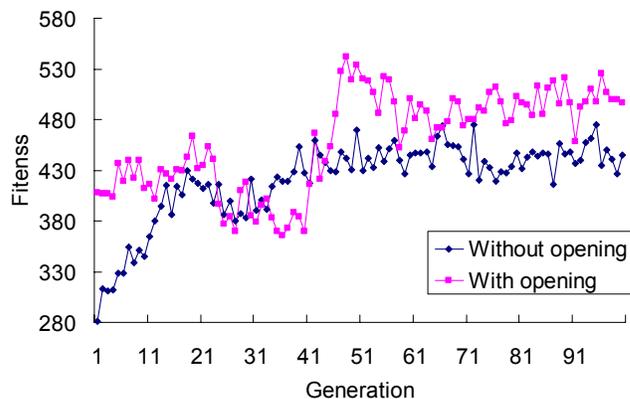


Figure 14: Comparison of the maximum fitness

Figure 13 shows the evolution of WPC with opening knowledge. The best fitness is about 540. After then, it converges to 500 though the population size is 50. The best individuals at the 1<sup>st</sup> generation is 400. It means that the incorporation of the opening knowledge gives some individual the high performance gain. The average fitness at the 1<sup>st</sup> generation is about 200. It is not quite different with the one of evolution without opening. That means the incorporation of opening does not increase the performance of all individuals in the population.

The evolution finds solutions that can do well after the opening and it can significantly reduce the burden for the WPC. Though the generations from 21 to 45 show performance degradation, the best individual that exploits the advantage of opening emerges at 51. Figure 14 shows the comparison of the maximum fitness over the evolution between the one with opening and the one without opening. It shows that the one with opening shows better fitness during evolution. Though the competition dose not support opening and the direct comparison against them is not possible, the 550 is about rank 12 in the CEC competition.

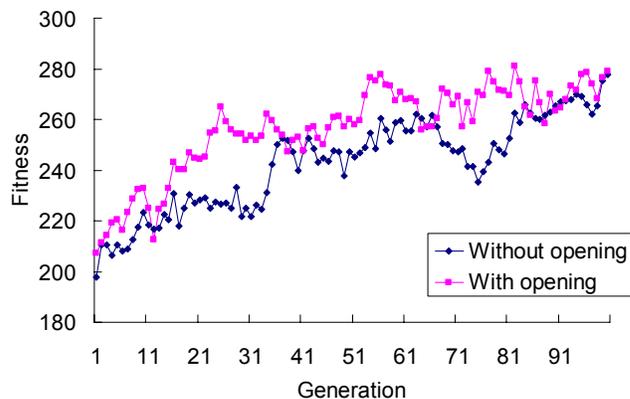


Figure 15: Comparison of the average fitness

Figure 15 shows the comparison of the average fitness between them. It shows that the average fitness of the one with opening is larger than one without opening but it finally converges to the same value. It means the incorporation of opening does not give huge advantage to the population. The average fitness of the 1<sup>st</sup> generation is not different and it

means that the evolution finds more improved individuals that exploit the opening knowledge.

Figure 16 shows the weights of the best WPC in the last generation of the evolution without opening. Compared to the standard heuristic, there is a big confusion in the weights. Figure 17 shows the weights of the best WPC in the last generation of the evolution with opening. It shows relatively clear separation of the dangerous area and the safe area in the near-corners. The relevance of the center is not big because the opening knowledge is used to find the initial moves (most of them are in the center of position). In the case of weight of evolution without opening, the weight of center is big (darkness).

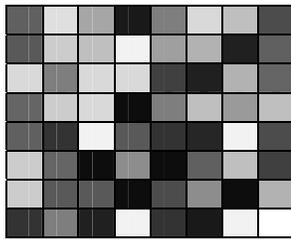


Figure 16: Weights of the best individual in the evolution without opening (white=0.0, black=1.0)

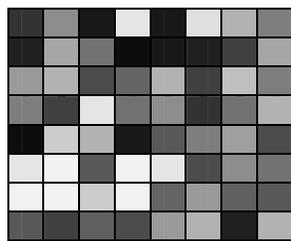


Figure 17: Weights of the best individual in the evolution with opening

## V. CONCLUSIONS

In this paper, we have proposed the incorporation of domain knowledge into the evolutionary Othello player. The incorporation of the domain knowledge can reduce the difficulty of the task and it makes the evolutionary search effective. The experimental results on the WPC evolution show that the opening knowledge can improve the performance of the evolution.

WPC can be a good solution for the position evaluation but has a limitation to be a best evaluator. It seems that MLP is one of the good choices. Though it needs careful configuration of the evolution, it has a possibility of promising results. Applying spatial preprocessing, self-adaptive mutation and co-evolution can improve the performance. Applying domain knowledge (opening) to the spatial MLP is the future work.

In this work, we have used probabilistic opening selection strategy but deterministic approach can be used (select one with the highest value). Investigating the difference between two approaches for the evolution is interesting research topic.

Also, changing the source of probabilistic information is interesting. In this work, we derived the information from the list of frequently used representative openings but it can be derived from a set of games played by human. There are many databases from human-played games. Also, the probability can be derived from the huge database of self-played games of programs (more than 12000 games by Logistello)

## ACKNOWLEDGEMENT

This research was supported by Brain Science and Engineering Research Program sponsored by Korean Ministry of Commerce, Industry and Energy.

## REFERENCES

- [1] D. E. Moriarty, and R. Miikkulainen, "Improving game-tree search with evolutionary neural networks," *IEEE World Congress on Computational Intelligence*, vol. 1, pp. 496-501, 1994.
- [2] D. E. Moriarty and R. Miikkulainen, "Evolving neural networks to focus minimax search," *Proceedings of the 12<sup>th</sup> National Conference on Artificial Intelligence*, pp. 1371-1377, 1994.
- [3] D. E. Moriarty and R. Miikkulainen, "Discovering complex Othello strategies through evolutionary neural networks," *Connection Science*, vol. 7, pp. 195-209, 1995.
- [4] S. Y. Chong, M. K. Tan and J. D. White, "Observing the evolution of neural networks learning to play the game of Othello," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 3, pp. 240-251, 2005.
- [5] S. Y. Chong, D. C. Ku, H. S. Lim, M. K. Tan and J. D. White, "Evolved neural networks learning Othello strategies," *The 2003 Congress on Evolutionary Computation*, vol. 3, pp. 2222-2229, 2003.
- [6] C.-T. Sun, and M.-D. Wu, "Self-adaptive genetic algorithm learning in game playing," *IEEE International Conference on Evolutionary Computation*, vol. 2, pp. 814-818, 1995.
- [7] C.-T. Sun, and M.-D. Wu, "Multi-stage genetic algorithm learning in game playing," *NAFIPS/IFIS/NASA 94*, pp. 223-227, 1994.
- [8] C.-T. Sun, Y.-H. Liao, J.-Y. Lu, and F.-M. Zheng, "Genetic algorithm learning in game playing with multiple coaches," *IEEE World Congress on Computational Intelligence*, vol. 1, pp. 239243, 1994.
- [9] J. Alliot, and N. Durand, "A genetic algorithm to improve an Othello program," *Artificial Evolution*, pp. 307-319, 1995.
- [10] E. Eskin, and E. Siegel, "Genetic programming applied to Othello: Introducing students to machine learning research," *Proceedings of the Thirtieth SIGCSE Technical Symposium on Computer Science Education*, vol. 31, no. 1, pp. 242-24, 1999.
- [11] R. Bateman, "Training a multi-layer feedforward neural network to play Othello using the backpropagation algorithm and reinforcement learning," *Journal of*

- Computing Sciences in Colleges*, vol. 19, no. 5, pp. 296-297, 2004.
- [12] R. E. Smith, and B. Gray, "Co-adaptive genetic algorithms: An example in Othello Strategy," *TCGA Report No. 94002*, The University of Alabama, 1994.
- [13] A. Leuski, and P. E. Utgoff, "What a neural network can learn about Othello," *Technical Report TR96-10*, Department of Computer Science, University of Massachusetts, Amherst, 1996.
- [14] A. Leuski, "Learning of position evaluation in the game of Othello," *Technical Report TR 95-23*, Department of Computer Science, University of Massachusetts, Amherst, 1995.
- [15] B. Rose, *Othello: A Minute to Learn... A Lifetime to Master*, <http://othellogateway.strategicviewpoints.com/rose/book.pdf>.
- [16] M. Buro, "Improving heuristic mini-max search by supervised learning," *Artificial Intelligence*, vol. 134, no. 1-2, pp. 85-99, 2002.
- [17] M. Buro, "How machines have learned to play Othello," *IEEE Intelligent Systems*, vol. 14, no. 6, pp. 12-14, 1999.
- [18] M. Buro, "The Othello match of the year: Takeshi Murakami vs. Logistello," *International Computer Games Association Journal*, vol. 20, no. 3, pp. 189-193, 1997.
- [19] D. E. Moriarty, A. C. Schultz, and J. J. Grefenstette, "Evolutionary algorithms for reinforcement learning," *Journal of Artificial Intelligence Research*, vol. 11, pp. 241-276, 1999.
- [20] D. B. Fogel, *Blondie24: Playing at the Edge of AI*, Morgan Kaufmann, 2001.
- [21] G. Kendall and C. Smith, "The evolution of blackjack strategies," *Proc. of the 2003 Congress on Evolutionary Computation*, vol. 4, pp. 2474-2481, 2003.
- [22] N. Richards, D. Moriarty and R. Miikkulainen, "Evolving neural networks to play go," *Applied Intelligence*, vol. 8, pp. 85-96, 1998.
- [23] D. B. Fogel, T. J. Hays, S. Hahn and J. Quon, "A self-learning evolutionary chess program," *Proc. of the IEEE*, vol. 92, no. 12, pp. 1947-1954, 2004.
- [24] J. B. Pollack and A. D. Blair, "Co-evolution in the successful learning of backgammon strategy," *Machine Learning*, vol. 32, no. 3, pp. 225-240, 1998.
- [25] D. B. Fogel and K. Chellapilla, "Verifying Anaconda's expert rating by competing against Chinook: Experiments in co-evolving a neural checkers player," *Neurocomputing*, vol. 42, no. 1-4, pp. 69-86, 2002.
- [26] Y. Jin, *Knowledge Incorporation in Evolutionary Computation*, Springer, 2004.
- [27] K.-J. Kim and S.-B. Cho, "Systematically incorporating domain-specific knowledge into evolutionary speciated checkers players," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 6, pp. 615-627, 2005.