Emerging Collective Intelligence in Othello Players Evolved by Differential Evolution

Tetsuyuki Takahama
Department of Intelligent Systems
Hiroshima City University
Asaminami-ku, Hiroshima, 731-3194 Japan
Email: takahama@info.hiroshima-cu.ac.jp

Setsuko Sakai
Faculty of Commercial Sciences
Hiroshima Shudo University
Asaminami-ku, Hiroshima, 731-3195 Japan
Email: setuko@shudo-u.ac.jp

Abstract—The evaluation function for game playing is very important. However, it is difficult to make a good evaluation function. In this study, we propose to play Othello using collective intelligence of players. The evaluation functions of the players are learned or optimized by Differential Evolution. The objective value is defined based on the total score of the games with a standard Othello player. In order to generate different types of players, the objective value is slightly changed by introducing the stability of each player. Each player can select a next move using the learned evaluation function. The collective intelligence player selects a move based on majority vote where the move voted by many players is selected. It is shown that the collective intelligence is effective to game players through computer simulation.

Keywords—collective intelligence; differential evolution; Othello; two-player game

I. INTRODUCTION

In zero-sum, two-player games with perfect information such as checkers, chess and Othello, successful computer-based approaches often utilize the following three elements:

- Opening book: An opening book is used to choose good moves in opening stage.
- Evaluation function: A next move is decided by using minimax method or αβ-pruning in middle stage. A minimax game tree for legal moves is constructed up to a certain depth. Leaf nodes of the tree are evaluated with the evaluation function. A move which has the best evaluation value is selected as the next move.
- Complete search: The best move in endgame stage is decided by searching all legal moves until the end of the game. Victory or defeat of the game can be determined if players take the best moves always.

When it is difficult to make an evaluation function for the game like Go, Monte Carlo tree search [1] can be adopted. Many playouts, in which both players select legal moves randomly or stochastically until the end of the game, are performed. Since the next move is selected based on the winning percentage of the playouts, the evaluation function is not used.

The evaluation function is very important element for playing game. However, it is difficult to make a good evaluation function, because the special knowledge for the game is necessary to make it and much effort is necessary to improve it through trial and error. In order to solve the difficulty, there are studies to learn the evaluation function or the game strategy automatically or by unsupervised learning. For example, reinforcement learning can be adopted to learn the strategy using the victory or defeat of games as a reward or a penalty. Also, optimization methods such as evolutionary computation can be adopted to learn the evaluation function using the score of a game as an objective value or a fitness value.

In this study, we propose to play Othello using collective intelligence of Othello players in order to investigate whether collective intelligence is effective to two-player games or not. Collective intelligence [2] is group intelligence that emerges from the cooperation or competition of multiple individuals. The intelligence appears in decision making by the individuals such as bacteria, animals, human beings and computer agents. For example, Kasparov versus the World [3] was a game of chess played in 1999 over the Internet. Kasparov faced with the World Team of over 50,000 people where moves are decided by plurality vote. Although Kasparov won, he admitted that he had never expended as much effort on any other game in his life. Also, ensemble learning in machine learning is a method where multiple models (weak learners), which have relatively low performance, are generated and combined to make a better model (strong learner), which have high performance. It is thought that the ensemble learning is an example of collective intelligence.

In this study, collective intelligence is introduced to Othello-learning players. The evaluation functions of the learning players are optimized by Differential Evolution, which is an evolutionary algorithm. The objective value of each player for optimization is defined based on the total score of the games with a standard Othello player. In order to generate different types of players, the objective value is slightly changed by introducing the stability of each player. Each player can select a next move using the learned evaluation function. The next move using collective intelligence of the players is decided by majority vote. That is, the collective intelligence player chooses the move which is voted by many players. It is shown that the collective intelligence is effective to game players through computer simulation.

In Section II, related works are briefly reviewed. Othello and the evaluation function are explained in Section III. An algorithm for evolving Othello players using Differential Evolution is proposed in Section IV. In Section V, collective
intelligence of the players is investigated and experimental results are shown. Finally, conclusions are described in Section VI.

II. RELATED WORKS

The representative methods for unsupervised learning of game strategy are reinforcement learning approaches such as temporal difference learning (TDL) [4] and evolutionary learning approaches [5]. A famous reinforcement learning approach was Tesauro’s TD-Gammon [6], [7] which played the backgammon. The game strategy of backgammon was expressed by the neural network which was learned by TDL. TD-Gammon was very strong, nearly as strong as the human world champion. Also, the strategies of chess [8] and Othello [9] were learned by TDL. Evolutionary learning approaches have been applied to backgammon [10], checker [11], [12] and so on. Also, comparative studies on TDL and evolutionary approach [13] was performed and hybridization of both approaches [14], [15] was proposed. But players generated by unsupervised learning in chess, Othello, shogi or Go are not comparable to strong human players.

Recently, collective intelligence in games has been paid attention to. Takahama and Sakai [16], [17] proposed to learn multiple players who play tic-tac-toe using neural networks learned by TDL in artificial game society. A consultation system to play Shogi was proposed in [18]. One of the moves that are sent by a set of players is selected as the actual move. It was shown that the system composed of three famous Shogi programs played better than each software individually. Marcolino and Matsubara [19] proposed a multi-agent version of UCT Monte Carlo Go. During the Monte Carlo simulations, one agent is randomly selected in multiple agents (agent database) and the agent will decide the move. The agents are generated by changing the order of the default heuristics of Fuego [20]. It was shown that a group of good agents selected from the database could significantly overcome Fuego.

It is thought that introducing collective intelligence to artificial game players is one of promising approaches in games. There are the following problems:

- Generating different types of players, where each player selects somewhat different moves, is key issue for emerging collective intelligence. Although different types of players can be prepared using some heuristics, the special knowledge for the game is necessary. Also, if a different initial solution or a different initial population is used in TDL or evolutionary approaches, different types of players might be generated. But the types of players tend to become similar when same objective value is used and the optimization process finds nearly optimal solution.

- The number of players is important to emerge collective intelligence. If a complex evaluation function is adopted, execute time for playing games becomes large and the number of players must be reduced.

In this study, stability of players in addition to the fitness of players is considered to generate different types of players using an evolutionary approach. A learning player plays with a standard player in plural games. The fitness value of the learning player in the games is defined by the results of the games. The plural games are repeated and the average and the standard deviation of the fitness values are obtained. Since a player who has small standard deviation is a stable player, it is thought that a player with larger fitness value and smaller standard deviation is a good player. In order to generate different types of players, the weight between the average fitness value and the standard deviation is changed appropriately. Also, the weighted piece counter which is the simplest evaluation function for Othello is adopted and 101 players are evolved to investigate whether collective intelligence can be emerged or not.

III. OTHELLO

Othello is a two-player board game with perfect information and is played on an 8 × 8 board. There are 64 identical pieces called discs which are white on one side and black on the other side. First move player uses black side and second move player uses white side.

A. Rules

The game starts with two white and two black discs where same-colored discs are on a diagonal with each other in the center on the board. The players take turns putting own colored disc on the board. A legal move consists in placing a disc on an empty position so that it will bound a horizontal, vertical, or diagonal line of opponent player’s discs by 2 current player’s discs on each end. The sandwiched opponent’s discs are reversed and the color of the discs is changed to current player’s color after the disc is placed. A player passes if and only if there is no legal move. The game ends when both players passed consecutively. Then, the player who has more discs with their color wins. If both players have the same number of discs with their color, the game is tied.

Fig. 1 shows the initial Othello board and legal moves of black as stars in the left and legal moves of white in the right.

Fig. 1. Initial state of Othello and legal moves.

B. The Othello League

A good reference for different Othello player models and their estimated performance is provided by the Othello Position Evaluation Function League [21]. The goal is not to design state-of-the-art Othello players, but to evaluate position evaluation functions. Therefore, a 1- ply setup is adopted: given the current state of the board, a player generates all legal moves and applies the evaluation function to the next states.
which are obtained by the legal moves. The most desirable evaluation value is selected to determine the next move. Ties are resolved at random. Player’s rank in the league is based on the score obtained in games played at 1-ply against the standard heuristic player who uses the standard weighted piece counter to be described later.

Since Othello game and the players are deterministic (with an exception when at least two positions have the same evaluation value and a position is randomly selected), there can be only two games with changing the color. In order to provide better performance measure, Othello League introduces some random factors to Othello. Both players are forced to make random moves with the probability of $\varepsilon = 0.1$. In other words, $\varepsilon$ greedy strategy is adopted by both players, where 10% of moves are selected randomly.

C. Weighted piece counter

The weighted piece counter (WPC) is the simplest position evaluation function which assigns a weight to each position in Othello board. Let a board position be denoted by $(x, y)$ which corresponds the $x$-th column and the $y$-th row where $x = 1, 2, \ldots, 8$, $y = 1, 2, \ldots, 8$. Let a weight of the position $(x, y)$ be denoted by $w_{xy}$. The evaluation function $f_w(\cdot)$ maps current board state $b = (b_{xy})$ into the evaluation value and is defined as follows:

$$
f_w(b) = \sum_{y=1}^{8} \sum_{x=1}^{8} w_{xy} b_{xy} \quad (1)
$$

$$
b_{xy} = \begin{cases} 
1 & \text{a black disc is placed} \\
0 & \text{empty} \\
-1 & \text{a white disc is placed}
\end{cases} \quad (2)
$$

The first move (black) selects a move that maximizes the evaluation value, but the second move (white) selects a move that minimizes the evaluation value.

Fig. 2 shows the standard heuristic WPC whose weights are hand-coded by Yoshioka et al [22]. This standard WPC is often used in Othello researches as an heuristic expert opponent.

![Fig. 2. The standard heuristic Weighted Piece Counter.](image)

IV. LEARNING WPC BY DIFFERENTIAL EVOLUTION

Multiple players learns their own game strategy represented by WPC using Differential Evolution to emerge collective intelligence.

A. Learning WPC

The number of variables for WPC is $8 \times 8 = 64$. It is possible to reduce the number of variables by taking advantage of symmetries of Othello board. As shown in Fig. 3, WPC can be represented by 10 variables from $a$ to $j$ [23].

![Fig. 3. The 10 variables a to j for weighted piece counter.](image)

The 10 sets of 100 games are played using WPC, which is defined by a decision vector $x$, against the 0.1-greedy standard WPC player with changing first move and second move. Thus, the problem of learning WPC can be defined as follows:

$$
\text{maximize} \quad f(x)
$$

$$
x = (a, b, c, d, e, f, g, h, i, j)
$$

$$
f(x) = \text{score}(x) + \alpha \text{stability}(x)
$$

$$
score_i(x) = (W_i + 0.5D_i)/N, i = 1, 2, \ldots, S
$$

$$
\text{score}(x) = \frac{1}{S} \sum_{i=1}^{S} \text{score}_i(x)
$$

$$
\text{stability}(x) = -\sqrt{\frac{1}{S} \sum_{i=1}^{S} (\text{score}_i(x) - \text{score}(x))^2}
$$

where $\alpha$ is the weight between score and stability, $W_i$ is the number of winning games, $D_i$ is the number of draw games in the $i$-th set, $N$ is the total number of games in one set where $N = 100$. This kind of score is used in real Othello tournament. $S$ is the number of the sets.

In (3), the stability is defined by the negative value of the standard deviation and a player with larger score and larger stability, or smaller standard deviation, can be obtained. In order to generate 101 players, $\alpha$ is changed as $\alpha = 0, 0.02, 0.04, \ldots, 2$.

B. Differential Evolution

Differential evolution (DE) is proposed by Storn and Price [24]. DE is a stochastic direct search method using a population or multiple search points. DE has been successfully
applied to optimization problems including non-linear, non-differentiable, non-convex and multimodal functions [25], [26]. It has been shown that DE is fast and robust to these functions.

In DE, initial individuals are randomly generated within given search space and form an initial population. Each individual contains $D$ genes as decision variables. At each generation or iteration, all individuals are selected as parents. Each parent is processed as follows: The mutation operation begins by choosing several individuals from the population except for the parent in the processing. The first individual is a base vector. All subsequent individuals are paired to create difference vectors. The difference vectors are scaled by a scaling factor $F$ and added to the base vector. The resulting vector, or a mutant vector, is then recombined with the parent. The probability of recombination at an element is controlled by a crossover rate $CR$. This crossover operation produces a trial vector. Finally, for survivor selection, the trial vector is accepted for the next generation if the trial vector is better than the parent.

There are some variants of DE that have been proposed. The variants are classified using the notation DE/base/num/cross such as DE/rand/1/bin and DE/rand/1/exp. “base” specifies a way of selecting an individual that will form the base vector. For example, DE/rand selects an individual for the base vector at random from the population. DE/best selects the best individual in the population. “num” specifies the number of difference vectors used to perturb the base vector. In case of DE/rand/1, for example, for each parent $x^i$, three individuals $x^{i_1}$, $x^{i_2}$ and $x^{i_3}$ are chosen randomly from the population without overlapping $x^i$ and each other. A new vector, or a mutant vector $m$ is generated by the base vector $x^i$ and the difference vector $x^{i_2} - x^{i_3}$, where $F$ is the scaling factor.

$$m = x^{i_1} + F(x^{i_2} - x^{i_3}) \tag{4}$$

“cross” specifies the type of crossover that is used to create a child. For example, “bin” indicates that the crossover is controlled by the binomial crossover using a constant crossover rate, and ‘exp’ indicates that the crossover is controlled by a kind of two-point crossover using exponentially decreasing the crossover rate. Fig. 4 shows the binomial and exponential crossover. A new child $x^\text{child}$ is generated from the parent $x^i$ and the mutant vector $m$, where $CR$ is a crossover rate.

C. The Algorithm of Differential Evolution

The algorithm of DE is as follows:

**Step 0** Population size $NP$, a scaling factor $F$ and a crossover rate $CR$ are given.

**Step1** Initialization of a population. Initial $NP$ individuals $P = \{x^i | i = 1, 2, \ldots, NP\}$ are generated randomly in search space and form an initial population.

**Step2** Termination condition. If the number of generations reaches the maximum number of generations $T_{max}$, the algorithm is terminated.

**Step3** DE operations. Each individual $x^i$ is selected as a target vector (parent). If all individuals are selected, go to Step4. A mutant vector $m$ is generated according to Eq. (4). A trial vector (child)

is generated from the parent $x^i$ and the mutant vector $m$ using a crossover operation shown in Fig. 4. In this study, the binomial crossover is adopted. If the child is better than the parent, or the DE operation is succeeded, the child survives. Otherwise the parent survives. Go back to Step3 and the next individual is selected as a parent.

**Step4** Survivor selection. The population $P$ is formed by the survivors. Go back to Step2.

![Fig. 4.](image-url)

**DE/rand/1()**

```java
// Initialize a population
N = N individuals generated randomly in S1;
for (i = 1; i <= N; i++) {
    // DE operation
    x^i = Randomly selected from P (p1 \neq i);
    x^i_2 = Randomly selected from P (p2 \neq (i, p1));
    m = x^i_2 + F(x^i_2 - x^i_1);
    x^i_\text{child} = trial vector is generated from x^i and m by the binomial crossover operation;
    // Survivor selection
    if (f(x^i_\text{child}) > f(x^i))
        z^i = x^i_\text{child};
    else
        z^i = x^i;
    P = {z^i | i = 1, 2, \ldots, N};
}
```

![Fig. 5.](image-url)

Fig. 5. The pseudo-code of DE, $T_{max}$ is the maximum number of generations.

V. EXPERIMENTS FOR COLLECTIVE INTELLIGENCE

A. Experiment 1 for Generating 101 Players

DE/rand/1/bin is executed 101 times to generate 101 different players with changing the weight $\alpha$ as 0, 0.02, 0.04, \ldots, 2. The parameters for DE are shown in Table I.
WPC defined by 10 variables is (53.34, 0.0591, and the average of the standard deviations is 0.5385, and the average score over all players is 0.573. The most stable player is player 86 whose standard deviation is 0.5385, and the average score over all players is 0.573. The reference. The highest scored player is player 37 whose score is 0.658 and votes. As for the best scored player 37, the score is 0.658 and the standard deviation is second lowest value of 0.0129. The WPC defined by 10 variables is (53.34, -11.30, 20.97, -3.76, -21.61, -2.57, 0.05, 0.50, -0.26, -0.16) which weights the corner position very strongly and is different from the standard WPC. As for the best scored player 37, the score is 0.6055, and the standard deviation is 0.0335, and the WPC is (3.47, 5.70, -0.40, 1.89, 26.89, -1.30, 0.00, -1.85, 2.77, 1.25) which is different from the best WPC and its standard deviation is 0.0398. Thus, it is thought that various WPC can be learned.

### Table II. WPC and Results for 101 Players

<table>
<thead>
<tr>
<th>Player</th>
<th>Initial score</th>
<th>Mean score</th>
<th>Standard deviation</th>
<th>Best score</th>
<th>Top 10 players</th>
</tr>
</thead>
<tbody>
<tr>
<td>37</td>
<td>0.658</td>
<td>0.573</td>
<td>0.5385</td>
<td>0.658</td>
<td>37</td>
</tr>
</tbody>
</table>

### B. Experiment 2 for Generalization Ability

Learning 0.1-greedy players against 0.1-greedy standard players is a highly noisy task due to stochastic fluctuation. Also, 1,000 games are only a small fraction of possible Othello games. Thus, the learned players might not have generalization ability to play various games. In order to investigate the generalization ability of the players, the 10 sets of 10,000 games are performed and the results are shown in Table III. The same seed for random numbers in the same set is used for every player.

The best scored player 37 in experiment 1 still attained the best score of 0.6274, although the second best player 19 attained somewhat lower score of 0.5582. It seems that player 37 is very good and stable player. The average score of all players is 0.5302 which is greater than 0.5. Thus, it is thought that players have generalization ability.

### C. Experiment 3 for Collective Intelligence

A collective intelligence player (CIP) is realized by 101 players: Each player decides a move using the player’s WPC. Each player votes the move. The move which has the maximum number of votes is selected as the move of CIP. If the number of votes is tied, a move is randomly selected from tied votes.
TABLE III. THE RESULTS IN 10 SETS OF 10,000 GAMES FOR 101 PLAYERS

<table>
<thead>
<tr>
<th>No</th>
<th>set 1</th>
<th>set 2</th>
<th>set 3</th>
<th>set 4</th>
<th>set 5</th>
<th>set 6</th>
<th>set 7</th>
<th>set 8</th>
<th>set 9</th>
<th>set 10</th>
<th>average</th>
<th>std</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.5294</td>
<td>0.5231</td>
<td>0.5253</td>
<td>0.5225</td>
<td>0.5219</td>
<td>0.5231</td>
<td>0.5226</td>
<td>0.5209</td>
<td>0.5216</td>
<td>0.5222</td>
<td>0.5221</td>
<td>0.0005</td>
</tr>
<tr>
<td>2</td>
<td>0.5289</td>
<td>0.5297</td>
<td>0.5275</td>
<td>0.5250</td>
<td>0.5241</td>
<td>0.5253</td>
<td>0.5244</td>
<td>0.5227</td>
<td>0.5232</td>
<td>0.5227</td>
<td>0.5236</td>
<td>0.0007</td>
</tr>
<tr>
<td>3</td>
<td>0.5269</td>
<td>0.5265</td>
<td>0.5272</td>
<td>0.5240</td>
<td>0.5232</td>
<td>0.5240</td>
<td>0.5230</td>
<td>0.5218</td>
<td>0.5220</td>
<td>0.5216</td>
<td>0.5226</td>
<td>0.0005</td>
</tr>
<tr>
<td>4</td>
<td>0.5382</td>
<td>0.5419</td>
<td>0.5407</td>
<td>0.5446</td>
<td>0.5446</td>
<td>0.5418</td>
<td>0.5424</td>
<td>0.5439</td>
<td>0.5360</td>
<td>0.5452</td>
<td>0.5437</td>
<td>0.0044</td>
</tr>
<tr>
<td>5</td>
<td>0.5383</td>
<td>0.5397</td>
<td>0.5397</td>
<td>0.5384</td>
<td>0.5382</td>
<td>0.5380</td>
<td>0.5383</td>
<td>0.5374</td>
<td>0.5385</td>
<td>0.5382</td>
<td>0.5386</td>
<td>0.0007</td>
</tr>
<tr>
<td>6</td>
<td>0.5309</td>
<td>0.5424</td>
<td>0.5372</td>
<td>0.5322</td>
<td>0.5320</td>
<td>0.5319</td>
<td>0.5329</td>
<td>0.5325</td>
<td>0.5323</td>
<td>0.5325</td>
<td>0.5329</td>
<td>0.0051</td>
</tr>
<tr>
<td>7</td>
<td>0.5555</td>
<td>0.5590</td>
<td>0.5590</td>
<td>0.5546</td>
<td>0.5581</td>
<td>0.5561</td>
<td>0.5541</td>
<td>0.5532</td>
<td>0.5540</td>
<td>0.5537</td>
<td>0.5542</td>
<td>0.0076</td>
</tr>
<tr>
<td>8</td>
<td>0.5344</td>
<td>0.5377</td>
<td>0.5367</td>
<td>0.5343</td>
<td>0.5342</td>
<td>0.5346</td>
<td>0.5338</td>
<td>0.5342</td>
<td>0.5342</td>
<td>0.5342</td>
<td>0.5343</td>
<td>0.0013</td>
</tr>
<tr>
<td>9</td>
<td>0.5345</td>
<td>0.5346</td>
<td>0.5346</td>
<td>0.5346</td>
<td>0.5346</td>
<td>0.5346</td>
<td>0.5346</td>
<td>0.5346</td>
<td>0.5346</td>
<td>0.5346</td>
<td>0.5346</td>
<td>0.0000</td>
</tr>
<tr>
<td>10</td>
<td>0.5345</td>
<td>0.5346</td>
<td>0.5346</td>
<td>0.5346</td>
<td>0.5346</td>
<td>0.5346</td>
<td>0.5346</td>
<td>0.5346</td>
<td>0.5346</td>
<td>0.5346</td>
<td>0.5346</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

Table IV shows the results of CIP (101 players) and the results of CIP without the best player 37 (100 players) under the same condition of experiment 2. The average results of 101 players in Table III is also included for comparison. In every set of games, CIP and CIP without player 37 attained better results compared with the average results. CIP without player 37 outperformed CIP on average although the difference is very small. It is thought that the effect of the best player can be negligible.

Fig. 6 shows the scores of CIP, CIP without player 37 and all players in the 10 sets. A thick line shows the results of CIP and CIP without player 37, a nearly straight line shows the average scores of all players, and the line over 0.5 shows the results of player 37. It is clear that CIP with and without player 37 outperformed many players in 101 players and CIP attained better scores in all sets. Therefore, CIP is very good and stable player although CIP without player 37 outperformed all players including CIP with and without player 37.

It is difficult to know the best player in the learning in advance because the learning task is highly noisy and the best player in the learning in the learning games does not always attain the best result in unrelated games. On the contrary, in this experiment, CIP player is always better than the average player. Thus, it is thought that utilizing collective intelligence of many players is effective compared with a player randomly.

VI. CONCLUSIONS

Collective intelligence is one of very important issues in the study of artificial intelligence, especially in the area of computer vision. The results show that the collective intelligence by major vote of 101 players can defeat player 37 on average. It is thought that CIP is more stable player than player 37 and can defeat not only standard player but also player 37.

Player 37 is a very strong player against standard player. However, it is not sure whether player 37 is the best player in general. Experiments show that the collective intelligence by major vote of 101 players can defeat player 37 in highly noisy task of learning Othello players. In real life, players will learn game
Table IV. The results in 10 sets of 10,000 games for Collective Intelligence Player

<table>
<thead>
<tr>
<th>No</th>
<th>set 1</th>
<th>set 2</th>
<th>set 3</th>
<th>set 4</th>
<th>set 5</th>
<th>set 6</th>
<th>set 7</th>
<th>set 8</th>
<th>set 9</th>
<th>set 10</th>
<th>average</th>
<th>std.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIP</td>
<td>0.5512</td>
<td>0.5423</td>
<td>0.5513</td>
<td>0.5477</td>
<td>0.5410</td>
<td>0.5570</td>
<td>0.5520</td>
<td>0.5410</td>
<td>0.5401</td>
<td>0.5471</td>
<td>0.5471</td>
<td>0.0055</td>
</tr>
<tr>
<td>CIP w/o 37</td>
<td>0.5527</td>
<td>0.5466</td>
<td>0.5476</td>
<td>0.5483</td>
<td>0.5473</td>
<td>0.5524</td>
<td>0.5585</td>
<td>0.5403</td>
<td>0.5556</td>
<td>0.5496</td>
<td>0.0050</td>
<td></td>
</tr>
<tr>
<td>average</td>
<td>0.5304</td>
<td>0.5297</td>
<td>0.5301</td>
<td>0.5298</td>
<td>0.5303</td>
<td>0.5303</td>
<td>0.5309</td>
<td>0.5301</td>
<td>0.5299</td>
<td>0.5306</td>
<td>0.5302</td>
<td>0.0045</td>
</tr>
</tbody>
</table>

Fig. 6. Score of CIP and all players in the 10 sets.

Table V. The results in 10 sets of 10,000 games for Collective Intelligence Player against Player 37

<table>
<thead>
<tr>
<th>No</th>
<th>set 1</th>
<th>set 2</th>
<th>set 3</th>
<th>set 4</th>
<th>set 5</th>
<th>set 6</th>
<th>set 7</th>
<th>set 8</th>
<th>set 9</th>
<th>set 10</th>
<th>average</th>
<th>std.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIP</td>
<td>0.5152</td>
<td>0.5101</td>
<td>0.5148</td>
<td>0.5139</td>
<td>0.5054</td>
<td>0.4998</td>
<td>0.5067</td>
<td>0.5009</td>
<td>0.5083</td>
<td>0.4978</td>
<td>0.5073</td>
<td>0.0057</td>
</tr>
<tr>
<td>CIP w/o 37</td>
<td>0.5029</td>
<td>0.5102</td>
<td>0.5130</td>
<td>0.5145</td>
<td>0.5079</td>
<td>0.5092</td>
<td>0.4988</td>
<td>0.5198</td>
<td>0.5068</td>
<td>0.5148</td>
<td>0.5098</td>
<td>0.0056</td>
</tr>
<tr>
<td>average</td>
<td>0.5091</td>
<td>0.5102</td>
<td>0.5139</td>
<td>0.5142</td>
<td>0.5067</td>
<td>0.5045</td>
<td>0.5028</td>
<td>0.5104</td>
<td>0.5076</td>
<td>0.5063</td>
<td>0.5085</td>
<td>0.0056</td>
</tr>
</tbody>
</table>

Strategies by playing with various types of players in a game society. It is interesting to consider the relation between collective intelligence and an artificial game society.

- All players had the same weight to vote the next move in this study. If proper weights for each player can be learned, the performance of collective intelligence will be improved.

Acknowledgment

This research is supported in part by Grant-in-Aid for Scientific Research (C) (No. 24500177, 26350443) of Japan society for the promotion of science and the 2014 Research Fund (ChosaKenkyu-Hi) of the Center for the Co-creation of Hiroshima’s Future, Hiroshima Shudo University.

References


