Abstract. In fields such as Go and Shogi, the usefulness of artificial intelligence has been recognized. However, it is impossible to apply the knowledge gained there to real-world problems. If artificial intelligence can be applied to real-world problems, the possibilities for artificial intelligence will expand even further. Therefore, in order to set up a problem close to the real world, we used the "RoboCup Soccer Simulation 2D League" as the subject of our experiment and aimed to improve the strength of the team using artificial intelligence. Experiments were conducted using fuzzy inference methods and genetic algorithms. In this paper, we will design an evaluation function to improve the strength of teams, and examine the usefulness of artificial intelligence-based teams based on the competitive performance of the designed teams.

Keywords: RoboCup2D, Artificial intelligence, Genetic algorithm, Numeric GA, Fuzzy reasoning method, simplified inference method.

1. Introduction

Go and Shogi (Japanese chess) needs enough computation time and game states can be represented discretely. Artificial intelligence has won against professional players and was shown usefulness in such games. However, it is impossible to apply the findings obtained in Go and Shogi to real-world problems. This is because real problems have limitations such as noise of information and time limit of decision making. The purpose of this paper is to apply artificial intelligence to a real-world problem. In order to conduct experiments in real-world problem settings, the RoboCup Soccer Simulation 2D League (RoboCup2D) was chosen as the research object. The purpose of this experiment is to improve the strength of a soccer team. RoboCup2D is a soccer game played between agents on a virtual field in a computer [1]. In addition, RoboCup2D provides a sample team (agent2d), which guarantees a certain degree of soccer-like movements. agent2d has implemented an action chain search framework [2]. We propose a method of designing the evaluation function using the real-valued genetic algorithm (numeric GA) and the fuzzy inference method. The design method of the evaluation function is explained, and its usefulness is evaluated from the match results by playing a match with the default agent2d as an opponent.

2. Agent2d overview

2.1. About agent2d

Agent2d is a sample agent that is a simplified version of team HELIOS. Team HELIOS is the winner of RoboCup2010, the RoboCup2D competition held in 2010. The team used in that tournament is the prototype. In agent2d, an action chain search framework has been introduced. This is used to determine the next action of the player holding the ball. When a player kicks the ball, the player decides the next cycle’s action by representing the action plan in a tree structure and generating a search tree. The decision of the action is made by selecting and executing the action with the highest evaluation value calculated by the evaluation function. The content of the evaluation is the assessment of each predictive state created by the search tree.

2.2. Differences between the research content of this paper and past research

From Section 2.1, we can see that optimizing the evaluation function leads to an improvement in the strength of the team. The evaluation function implemented in agent2d is a simple one. It is based only on the distance between the ball position and the opponent’s goal position. Figure ?? shows the coordinate system of RoboCup2D. In the past, many studies on evaluation functions have been conducted and their usefulness has been recognized [3–5]. We designed the evaluation function for this study. The design method was based on numeric GA and fuzzy inference method. The reason for using such a method is that an algorithm with fast processing time is required. In RoboCup 2D, the agent must decide within 0.1 seconds. After this time, the agent is carried over to the next cycle and remains stationary for one cycle. Therefore, it is not possible to apply processes with heavy computational load. Various findings can be obtained from this experiment. One is that we can determine the usefulness of learning methods using unsupervised learning such as real-valued GA. We have also ap-
plied the fuzzy inference method to a soccer field. From this, we can search for the most likely position on the field to score a goal.

Fig. 1. Soccer field position coordinates

3. Proposal of evaluation function using genetic algorithm and fuzzy inference method

3.1. Match conditions
We aimed to improve the strength of the team by designing an evaluation function. Use artificial intelligence to design the evaluation function. In this experiment, we aim to improve the scoring ability. The game mode during the experiment was assumed to be performed from free kicks. This is to ensure that the attack is carried out efficiently. Every 200 cycles, we started in free-kick mode and played 3000 cycles of the game. The condition is that the position of the ball for the free kick is random. If the ball crosses the half line, the game is stopped until 200 cycles have elapsed. The condition is that the position of the ball for the free kick is random. And if the ball crosses the half line, the game is stopped until 200 cycles have elapsed. Figure 2 shows an overview of the experiment.

3.2. Design of evaluation functions using fuzzy inference methods
In the calculation part of the evaluation function, we used a simplified inference method. This is a kind of fuzzy inference method. The simplified inference method was proposed by Maeda et al [6]. The fuzzy inference rule is shown in Equation 1. The input is the value of the x-axis and y-axis in the soccer field and the output is r. \( x_i \), \( y_j \) are the membership functions of the x- and y-axes, respectively, and \( r^*_k \) is the real value of the posterior. Figure 3 and Figure 4 show the membership functions for the x- and y-axes. As can be seen from the figure, in this experiment, we set \( n = 6 \) and \( m = 5 \). The total number of rules was set to be 30. Using r, we derived the evaluation value using Equation 2. U is the evaluation value and x is the positional coordinate of the ball after prediction.

\[
\text{R}_k: \text{If } x \text{ is } x_i \text{ and } y \text{ is } y_j \quad \text{Then } r = r^*_k \\
i = 1, \ldots, n \quad j = 1, \ldots, m \quad k = 1, \ldots, l
\]

(1)

\[
U = x + (40 - r)
\]

(2)

3.3. Learning with numeric Genetic Algorithm
The real value \( r^*_k \), which is the posterior of the simplified inference method, is the value needed to determine the value of r. We tuned the parameters of this \( r^*_k \) by using the numeric GA. The number of generations was set
Design and validation of an evaluation function using the GA for the Fuzzy inference of the action chain search in RoboCup2D

to 100 and the number of individuals to 30. In this experiment, crossover and mutation are performed in sequence. The reason for the mutation is that we are experimenting with a smaller number of individuals than we should. For crossover, we use BLX-α crossover, and for mutation, we use Gaussian mutation. For the selection method, tournament selection was used, and experiments were conducted with tournament sizes of 3 and 4, respectively. Equation 3 was used as the fitness function to evaluate the individuals. \( point \) is the number of goals scored, \( shoot \) is the number of shots taken, and \( penalty \) is the number of entries into the opponent’s penalty area. This fitness function is intended to give higher ratings to individuals with higher scoring ability. One match was played per individual and the match was evaluated. The opponent for the game is agent2d, who has not been changed. The number of generational changes is set to a small number. The reason for this is that it takes about a minute to evaluate one individual. Thus, we are solving the fitness maximum search problem.

\[
\text{fitness} = 2point + shoot + 0.3penalty
\]

4. Experimental result

4.1. Learning results

Figure 5 and Figure 6 shows the experimental results. Figure 5 shows the training results for tournament size 3, and Figure 6 shows the training for tournament size 4. The horizontal axis is the number of generations, and the vertical axis is the maximum evaluated value of an individual in each generation. From the figure, we can see that the evaluation value increases as the generation is updated. Therefore, the success of the study was confirmed. Comparing the two figures, we can see the difference in the increase in the maximum evaluation value. In the case of tournament size 3, the maximum evaluation value increases as the generation progresses. In the case of tournament size 4, the increase stops around generation 20. There was also a difference in the final maximum evaluation value. The maximum rating for a tournament size of 3 was 27.9, while it was 23.6 for a tournament size of 4.

4.2. Results against agent2d

Played 100 games with the default agent2d as the opponent. The purpose is to see if there is any change in the strength of the team compared to the unmodified agent2d. The conditions of the game were as shown in Figure 2. As shown in Section 3.1, the match conditions are the same as those used in the experiment. Table 1 shows the results of the 100 games played by each team. In the point section, it was observed that the tournament size of 3 had the highest scoring power. However, the difference between the other two teams were not large, and all teams had similar scoring power. In the shoot section, we found that the default agent2d had the most opportunities to shoot. In the penalty section, it was highest in the case of tournament size 3. Comparing teams with tournament size 3 and 4, we found that the team with tournament size 4 was inferior in all respects.

5. Consideration

The maximum evaluation value of individuals performed under the experimental conditions of tournament size 4 is lower than that of tournament size 3. This can be confirmed from Figures 5 and 6. In the case of tournament size 3, we found that the increase stopped after about 20 generations. This suggests that learning may have stopped at the locally optimal solution. We calculated the correlation between the \( r^*_k \) obtained by each training and obtained a result of 0.858. We can say that it is a high positive correlation, but we cannot deny that it falls into a local optimum solution. As can be seen in Table 1, the performance of the teams with tournament size 4 was inferior in all respects compared to the teams with tournament size 3. This could be due to a problem with the algorithm of the real-valued GA. We thought it was necessary to revise the algorithm. Therefore, I’m going to run the experiment several times under the same conditions to see the difference in results.

From Table 1, we can see the advantages and disadvantages of the tournament size 3 compared to the default agent2d. In terms of scoring, the team with tournament size 3 had a higher score than agent2d. We can consider tournament size 3 to be superior to this in terms of team strength. However, a t-test was conducted at a signifi-
Table 1. Match results of 100 games by each team

<table>
<thead>
<tr>
<th>Evaluation material</th>
<th>Shoot</th>
<th>Penalty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tournament size 3</td>
<td>1.87</td>
<td>3.51</td>
</tr>
<tr>
<td></td>
<td>1.22</td>
<td>1.77</td>
</tr>
<tr>
<td></td>
<td>10.42</td>
<td>2.66</td>
</tr>
<tr>
<td>Tournament size 4</td>
<td>1.36</td>
<td>2.86</td>
</tr>
<tr>
<td></td>
<td>1.20</td>
<td>1.63</td>
</tr>
<tr>
<td></td>
<td>8.58</td>
<td>2.57</td>
</tr>
<tr>
<td>agent2d</td>
<td>1.64</td>
<td>4.16</td>
</tr>
<tr>
<td></td>
<td>1.26</td>
<td>1.80</td>
</tr>
<tr>
<td></td>
<td>9.9</td>
<td>2.35</td>
</tr>
</tbody>
</table>

cance level of 0.05, and the result was 0.21, indicating that the null hypothesis was not rejected. It was confirmed that the scoring ability of the two groups was similar. One possible reason is that the number of repeated generations is small. In our experiments, we used a small number of repeated generations. The reason for this is that it takes time to evaluate a single individual. Figure 5 shows that the improvement was also observed around the 90th generation. This suggests that the improvement can be expected by increasing the number of iterative generations. In addition, in this experiment, the coordinates of the y-axis were set to absolute values when calculating r using the simplified inference method. Therefore, we use the same evaluation function for the left and right sides. It may be possible to expand the scope of the attack by making the side different rather than absolute.

The evaluation function was designed by using fuzzy inference method with static parameters. Therefore, the same evaluation function is used in all situations to evaluate the state and determine the action. However, in soccer, the position of the enemy agent or the position of a teammate near the ball can change depending on the situation. Therefore, we thought it would be difficult to evaluate all the situations with static parameters. When we checked the file containing the results of each game obtained for the calculation of Table 1, we found that some games scored 7 points per game, while others scored no points at all. The standard deviation is 1.22, but the results are expected to vary depending on the situation. This suggests that it is necessary to consider the design of evaluation functions with dynamic parameters that change depending on the situation.

Using RoboCup 2D as an experiment simulation, we discussed the improvement of team tactics. First, we designed the evaluation function using the fuzzy inference method and tuned the real value of the posterior with the numeric GA. Next, we assessed the usefulness of the team we designed. The results are comparable to the default ones. However, we were also able to analyze areas for improvement. Therefore, in the future, we plan to review the improvements we have considered and conduct experiments to improve the strength of our team. We also discussed a new design method for the evaluation function, which we plan to design at the same time.

References: