WIT2017:Team Description Paper

Ryo Mizushima and Hajime Anada

Tokyo City University, 1-28-1, Tamazutsumi, Setagaya-ku, Tokyo, 158-8557, Japan.

Abstract. RoboCup is an international scientific initiative to advance state-of-the-art intelligent robots. RoboCup features 2D and 3D simulation leagues (RoboCupSoccer) that focus on artificial intelligence and team strategy. In the 2D league, the artificial intelligence of the 11 team agents has been implemented by many models, most of which require the special knowledge of soccer. Here, we construct two algorithms based on a real-coded genetic algorithm for decision making and person-to-person defense training by Q-learning, without consulting the special knowledge of soccer.

Keywords: RoboCup, 2D simulation league,Real-Coded GA, Q-learning

1 Introduction

RoboCup is an international scientific initiative to advance the state-of-the-art design of intelligent robots. RoboCup focuses on artificial intelligence and team strategy through its 2D and 3D simulation leagues (called RoboCupSoccer). In the 2D league, many artificial intelligence models have been proposed for manipulating the eleven team agents, without requiring the special knowledge of soccer. Our WIT2017 team is based on agent 2d (Ver.3.11) [1] by H. Akiyama. We aim to create a strong team without consulting the special knowledge of soccer. Here, we propose two algorithms; one based on real-coded genetic algorithm (GA)[2] for decision making, the other based on Q-learning for person-to-person defense training[3].

2 Decision Making using Real-Coded GA

2.1 Coordinate system of soccer field

The coordinate system of the field in RoboCupSoccer's 2D simulation league is shown in Fig.1.

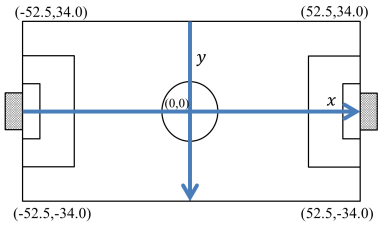

Fig. 1. Coordinate System of the Soccer Field

2.2 Evaluation value for learning Decision Making

The evaluation value, which evaluates each agent as the game unfolds, depends on the actions of the agent (namely, their dribbling, pass, and hold actions). The evaluation value V of an action a is therefore defined as follows:

$$V(a) = \alpha \times Y + \sum_{p=1}^{3}(w_{ap} \times U_{ap}) \tag{1}$$

where α is the weight factor of the common term Y for all actions, U_{ap} is an evaluation term of measure p with action a, and w_{ap} is the weight factor of U_{ap}. The common value Y is defined as follows:

$$Y = \begin{cases} 34.0 - \frac{|y_b|}{34.0} & if\ x_b > th_1 \\ 0.0 & otherwise \end{cases} \tag{2}$$

where x_b and y_b are the x- and y-coordinates of the ball, respectively, after an action, and th_1 is the threshold of x_b. A ball cannot be passed overhead, because the 2D Simulation League excludes the height dimension. Accordingly, all actions from the sides have a disadvantage. Therefore, V is set to higher values around $y = 0$.

The evaluation terms U_{ap} are defined as follows:

$U_{a1} = \frac{(x_b+52.5)}{105.0}$: $degree\ of\ closeness\ to\ the\ opponent\ goal\ line$
$U_{a2} = \frac{max0.0, th_{a2} - dist_{bg}}{th_{a2}}$: $degree\ of\ closeness\ to\ the\ opponent's\ goal$
$U_{a3} = dist_{b\_no}$: $degree\ of\ freedom\ from\ the\ opponent\ agents$

where $dist_{bg}$ is the distance between the ball and the opponent's goal after an action, th_2 is a threshold of $dist_{bg}$, and $dist_{b\_no}$ is the current distance between the ball and the nearest opponent agent. All U_{ap}s are adjusted to lie between 0.0 and 1.0.

The number of parameters (thresholds plus weights) is 14. As each parameter is prepared according to six positions, the total number of parameters is 84.

2.3 Application of Real-Coded GA

The team is evaluated by the next evaluation score G, using the results of 20 games against 4 arranged teams participating in the RoboCup Japan Open 2011.

$$G = \frac{\sum_{i=1}^{4}\sum_{j=1}^{20}(P_{ij} + \frac{D_{ij}}{100})}{4 \times 20} \tag{3}$$

where P_{ij} is the j-th game point against team i, and D_{ij} is the j-th game goal difference against team i. All teams are evaluated by the average game point and average goal difference. The team is evaluated by the average score G of 80 games.

2.4 Procedure of Real-Coded GA

The algorithm based on real-coded GA [2] is implemented as follows.

First, this procedure constructs 15 teams with 84 random parameters and one team that adopts the common knowledge of soccer games. These teams compete against the four arranged teams and the evaluation score G is calculated for each set of games.

Next, the following four steps are implemented for a specified number of iterations.

1 Crossover
From the 16 teams, construct 8 groups consisting of two teams by ranking selection[2]. Then construct 16 teams by crossover of these groups.

2 Mutation
Replace each parameter value with random value in each range shown in sec.4.1 with a probability of 5%.
3 Calculate the evaluation score G
Calculate the evaluation score G of the 16 new teams playing against the 4 arranged teams.
4 Selection
Select 16 teams from the above 32 teams to the next generation.

3 Training of Person to Person Defense

3.1 Training method

Figure 2 shows the exercise regime in a real soccer game. The offense and defense agents are indicated by red and blue circles, respectively. This coordinate system of the playing field which is $15m$ long by $8m$ wide, is shown in Fig.3. The offense agent tries to dribble the ball beyond the red line. The defense agent try to steal the ball or carry it beyond the blue line.

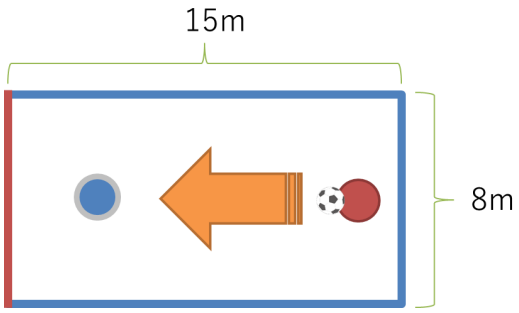

Fig. 2. Field for training of person to person defense

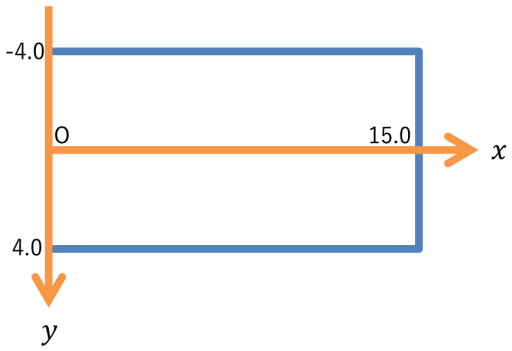

Fig. 3. Coordinate system of the field for training

3.2 Method of applying Q-learning

Q-learning In Q-learning[3], an agent observes current state S_t and selects an action a and receives a consequent reward r. $Q(S_t, a)$ is updated $Q(S_{t+1}, a)$ by the following equation:

$$Q(S_t, a) \leftarrow Q(S_t, a) + \alpha[r + \gamma\, max_p Q(S_{t+1}, a) - Q(S_t, a)] \tag{4}$$

where α is the learning rate, γ is the discount factor, and $max_p Q(S_{t+1}, a)$ is the maximum Q value of the next possible states. In this study the defense agent updates $Q(S_t, a)$ by repeating the observation and the action selection.

Design of reward We must define the reward r, the observable states S_t and the selectable actions. The reward r is based on the degree of attainment. Therefore we set the next preliminary goals of the defense agent as follows.

1 Steal the ball

2 Remain in the field
3 Face the ball
4 Retrude the ball
5 Approach the ball to approximately $1.0m$

As mentioned above, the reward r determines the degree of attainment of the preliminary goals:

$$r = \sum_{i=1}^{5} reward_i \tag{5}$$

where

$$reward_1 = \begin{cases} 10000 & if\ the\ defense\ agent\ take\ the\ ball \\ 0 & otherwise \end{cases}$$
$$reward_2 = \begin{cases} -100 & if\ the\ defense\ agent\ leaves\ the\ field \\ 0 & otherwise \end{cases}$$
$$reward_3 = max\left(\frac{90.0 - |\theta_{sb}|}{60}, 0.0\right)$$
$$reward_4 = max(x_b - prebx_b, 0.0) \times 5.0$$
$$reward_5 = min(15.0 - dist_{sb}, 15.0 - 1.0)$$

where the $reward_i$ is the result of achieving the above preliminary goal i, θ_{sb} and $dist_{sb}$ denote the angle between the x-coordinate and ball direction (viewed from defense agent) and the distance between the defense agent and the ball, respectively (see Fig.4), and $prebx_b$ is the x-coordinate of the ball at previous step.

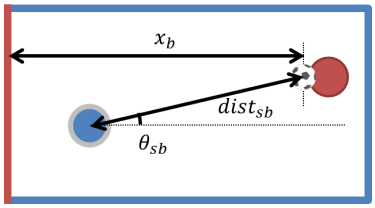

Fig. 4. Positional relation of the agents and the ball

Definition of state The defense agent must select an action depending on its situations. Therefore, to study the person-to-person defense, we define the following states.

1 The area of the ball viewed by the defense agent (identified among eight facing directions)
2 The Y-axis coordinate of the defense agent (plus or minus)
3 The Y-axis coordinate of the ball (plus or minus)
4 The defense agent is within $1.5m$ of the ball (true or false)
5 The defense agent is within $5.0m$ of the ball (true or false)
6 The ball is within $1.5m$ of the blue line (true or false)
7 The ball is within $1.5m$ of the red line (true or false)
8 The ball is within $1.5m$ of the offense agent (true or false)

In total, there are $1024(8 \times 2^7)$ states for the defense agent.

Act selection The agents can walk or run in one of eight directions. Other actions are tackle and stand still. There are 18 actions in total.

The defense agent randomly selects an action with probability ϵ and selects an action yielding the highest Q value with probability $(1-\epsilon)$.

4 Results

4.1 Real-Coded GA

The model was learned by competing against the four teams listed in table1.

Table 1. Names and abstracts of the arranged teams

Team	Abstract
agent2d ver.3.1.1	Developed by Akiyama in 2012
KU_BOST	These teams are the newest
TDUThinkingAnts	publically available teams from the
TOYOSU_GALAXY	RoboCup Japan（RoboCuo 2011）

Table2 summarizes the parameters and their settings in the initial setting of 16 teams.

Table 2. Range and increments of the parameters

Parameter	Range	Increment
α	[0.0,5.0]	0.1
th_1	[-50.0,50.0]	2.0
th_2	[0.0,100.0]	2.0
w_{ap}	[0.0,5.0]	0.1

The simulations were iterated through 30 generations. The best evaluation value G at each generation is plotted in Fig.5. Clearly, G is a rising function of number of generations. We conclude that our algorithm effectively learns decision-making without consulting the special knowledge of soccer.

4.2 Q-learning

In preliminary experiments we determined $\alpha = 0.1, \gamma = 0.9, \epsilon = 0.3$. The initial locations of the defense and offense agents are $(0.5, 0.0)$ and $(14.5, 0.0)$, respectively. Agent2d was employed as both the offense agent and the initial defense agent.

The Q-learning method iterates the following steps until the Q value converges.

1 Place the ball and the agents at their initial positions
2 Repeat exercise until the ball leaves the field or 30 seconds has elapsed

Figure 6 plots the maximum Q value summed over all states versus number of iterations. The sum converges after sufficiently many iterations.

The success probabilities of training in the games of (agent2d $\times$ agent2d) and (learned agent $\times$ agent2d) are summarized in table3.

From these results, we conclude that our algorithm effectively trains the person to person defense strategy.

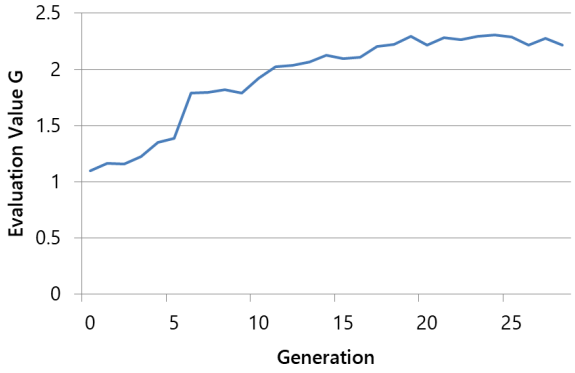

Fig. 5. The best evaluation value G at each generation

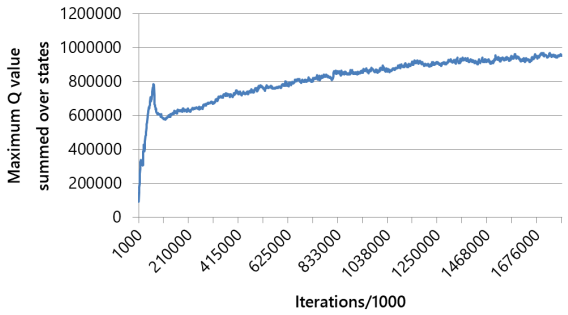

Fig. 6. Maximum Q value summed over states at each 1000 iterations

Table 3. Success probability of the training program

	Defense:agent2d,Ofense:agent2d	Defense:Learned agent,Ofense:agent2d
Success probability	6%	94%

5 Conclusion

Our proposed algorithm (real-coded GA and Q-learning) strengthen the soccer team without consulting special knowledge of soccer. The effectiveness of our algorithms was confirmed in simulation experiments.

In future work, our person to person defense training will consider the surrounding environment.

Acknowledgments

The authors would like to thank Enago (www.enago.jp) for the English language review.

References

1. Akiyama H., Shimora H. and Noda I.: HELIOS2009 Team Description. In Robocup 2009 (2009)
2. Eshelman, L.J. and Schaffer, J.D.:Real-Coded Genetic Algorithms and Interval-Schema, Foundations of Genetic Algorithm 2, pp.187-202 (1993)
3. Watkins C.J.C.H. and Dayan P.: Technical note: Q-learning, Machine Learning, vol.8, pp.279-292 (1992)