Minority Game Strategies for Dynamic Multi-Agent Role Assignment

Tingting Wang, Jiming Liu and Xiaolong Jin
Department of Computer Science
Hong Kong Baptist University
Kowloon Tong, Hong Kong
{ttwang,jiming,jxl}@comp.hkbu.edu.hk

Abstract

In different multi-agent systems, agents coordinate with different fashions. In team-based games, agents coordinate to enhance the collaborative behaviors of a team. An attractive problem in team-based games is the role assignment problem. It requires agents to decide which roles they should take based on the real-time feedback from a dynamically changing environment. The Minority Game, widely used in modeling financial marketing problems, has very similar characteristics which meet the fundamental requirements of the role assignment problem. In this paper, we propose a formulation of MG strategies in the role assignment problem in a particular multi-agent system: Simulation Robot Soccer. Through experiments, we demonstrate that MG strategies improve the effectiveness of the role assignment among agents in a real competitive environment. The improvement obeys some characteristics discovered in the theoretical MG model.

1 Introduction

1.1 Motivation

In multi-agent systems (MASs), agents interact with each other or with their environment to achieve coordinative behaviors. MASs can have various coordination fashions. For example, in Cellular Automata(CA), agents, cells in a lattice, coordinate using some *local rules* to update their states[12]. In Sensor Networks [2], which consist of large amount of vast distributed sensors, agents coordinate to optimizing the gathering and routing of massive data.

Team-based agents, involve another coordination manner, which helps agents enhance collaborative behaviors of a team. An attractive problem in team-based agents, especially in the environment of team-based games (e.g., RoboCup), is the *role assignment* problem. Agents in this environment face with dynamic changing situations for limited resources, such as, the communication ability, the energy restriction, etc. Agents taking different roles will face different tasks and consume different resources. The challenge in the role assignment is, the situation is changing dynamically and rapidly, agents need to decide which roles they should take based on the real-time feedback. Therefore, agents involved in the role assignment problem have the following characteristics:

- Situated in a dynamically changing environment;
- Have a real-time decision making process series;
- Use a performance driven payoff function;
- Provide with more than one roles and should select a role or more dynamically.

From the above descriptions, a recently appeared game theory, the Minority Game, come into our sight to solve the role assignment problem. The Minority Game (MG), proposed by Challet and Zhang [4], is one theory for studying cooperation and competition of agents given limited resources. Agents playing MG are making a set of sequential decisions adaptively based on their past experiences without interacting directly. In general, the characteristics of MG are:

- Players are making a sequence of decisions;
- Each decision is based on the history results, i.e., the decision is based on a changing environment;
- Each player will get a reward or punishment according to their performance;
- Each player are faced with two alternatives;

We can observe from the above descriptions that, MG is quite suitable to the requirements of the role assignment problem. Therefore, by assigning the players in MG with different roles, we can formulate MG strategies to the role assignment problem in a multi-agent system. We want to examine whether the formulations work for this problem, and whether the factors such as the memory size in MG affect the performance of the strategies.

To explore the above problems, we need to select a platform where agents with different roles coordinate in a competitive environment. Simulation Robot Soccer (SRS), therefore, come into our consideration. SRS is a good multi-agent testbed. It provides a standard platform for performing soccer games, and do not care about the real-robot designing. In SRS, virtual robots can run on different clients to play a soccer game via a server. Many technologies can be applied and examined in SRS, such as multi-agent collaboration, real-time reasoning and planning, and intelligent robotic design [10]. As introduced in the official web site [1], the goal of SRS is to foster the research in AI and robotics related fields.

Figure 1. A Snapshot of a competition of SRS Simulation League

1.2 Problem Statement

In Section 1.1, we described the fundamental requirements of the role assignment problem, and we found a quite suitable theory, MG, to solve this problem. We also selected an appropriate platform, SRS, to implement the given problem. Here we can detail the motivation that we described in the above section as follows:

1. How can we formulate the MG strategies in the the role assignment of SRS?
2. How can we implement the formulations?
3. How can we test whether the formulations in (1) is effectiveness in the the role assignment problem?
4. Can the important factors in MG such as the memory size affect the performance of MG strategies?

1.3 Organization of the Paper

The rest of the paper is organized as follows. Section 2 introduces the related work about MG and SRS. Section 3 gives the formulation of MG strategies to SRS. Section 4 presents the design of our experiments according to the formulation. Then in Section 5, we analyze the results we have achieved. Section 6 concludes the paper by summarizing the key points of our research.

2 Survey of Related Work

2.1 The Minority Game

2.1.1 Introduction of MG

Self-organization has been studied for a long time. In a self-organizing system, with local interactions among members, some emergent behaviors can appear. There is a particularly interesting system, in which agents are not interacting locally, self-organization is embodied by the feedback from collective behaviors. Minority Game (MG), proposed by Challet and Zhang [4], gives a simple model of this system. It is suitable to various of cases, especially to the competition for limited resources. For instance, in a stock market, everyone wants to sell in a high price and buy in a low price, therefore it is better to stand in a minority group. In a fire disaster, everyone wants to select an exit way chosen by the minority people. In the rush-hour, drivers always want to select a way containing minority traffics [9].

As claimed by Challet and Zhang, MG has three main basic parameters: the memory m, the number of players N, and the number of strategies S, which will be briefly introduced below.

Let us suppose there are N (odd) players, each of them has S strategies. At each cycle, they need to choose between two options A and B. After everyone has chosen an option independently, those who are in the minority side will win the game. All the strategies that have correctly predicted the winner side will get rewards. Players make their decisions according to the past records of results. We can use signals ‘1’ and ‘0’ to represent A is winning or not. Thus the result of the game can be denoted by a sequential binary series.

Signal	Prediction
000	1
001	0
010	1
011	1
100	1
101	0
110	0
111	1

Table 1. An example of MG strategy where $M = 3$.

Assume that players can only memorize the last M results, and make their next choice based on these results. A strategy is defined to be the next choice given a certain M bits memory. Table 1 is an example of one strategy for $M = 3$.

The second column of Table 1 is a distinct strategy. The number of all combinations of game's results is 2^M. And the total number of strategies is $2^{(2^M)}$.

The other important part of MG is its payoff function, which defines rewards for the winners. All winners will get rewards according to a predefined payoff function.

2.1.2 Related Work on MG

After the proposal of the standard MG, scientists have made some modifications and proposed several variants of MG, such as evolutionary MG [8], multiple choice MG [6], MG with hierarchically organization [7], and MG with local information [11].

Furthermore, a lot of mathematical characteristics of MG are utilized in real world applications. The standard MG can be considered as a very crude model of financial markets, because the minority mechanism is found in markets. Quite a lot of papers motivate their studies of MG by that of markets. But, almost all of them are focusing on the emergent collective behaviors of agents, not for achieving a common task that is pre-assigned to them. We plan to use MG in a different field, SRS, to test whether MG can help agents in the same team be better coordinated in a real competition against another team.

2.2 Simulation Robot Soccer

In the SRS league, a soccer server provides a virtual pitch, and simulates all movements of a ball and players. Players (agents) are connected to the server as clients over a local area network, and all communications must be carried out via the server. In this kind of SRS competition, no real robots are involved, and we can focus on the coordination and competition strategies among robots.

SRS is a totally distributed, multi-agent system. The architecture of a SRS client can be separated into three layers: basic behavior layer, advanced behavior layer and high level strategy layer. The strategy level requires an effective design of coordination and co-operations among agents with very limited local information. Figure 1 shows a snapshot of a real SRS competition.

3 The MG Formulation

We can notice that players in MG are learning to compete for limited resources. In Robot Soccer, the situations are quite similar. For example, each team includes only 11 agents, which means the resource is limited. These agents need to act in different roles according to different situations, sometimes as attackers, sometimes as defenders. All the agents are initially homogeneous with similar attributes. In most of SRS teams, Neural Networks are used to train agents to select proper roles during the competition. Due to the features that agents in MG can dynamically make their decisions according to their previous experiences, i.e., histories, we formulate MG strategies to SRS as follows.

3.1 Formulation of MG

Let us define the payoff function for MG first.

We can introduce a variable $a = \{-1, 1\}$ to denote the outcome of a player in the game. a is related to the provided choices of MG. After each cycle, we can know the outcome at time *t* by summarizing outcomes of all players in the game:

$$C(t) = \sum_i a_i(s_i(t), h(t)) \tag{1}$$

where $h(t)$ is the history, and $s_i(t)$ is the strategy used by player *i* in this cycle.

From Eq. 1, we get the difference between the numbers of winners and losers in the last cycle. Suppose that 5 players face with choices {A, B}, if a player chooses A, the outcome is 1; otherwise, the outcome is -1. Assume the result after a cycle is {1, -1, -1, 1, -1}. Therefore $C(t)$ is -1, representing that players who chose B lose the game, and the difference between the numbers of winners and losers is $|C(t)|$, whose value is 1.

The payoff $r_i(t)$ to player *i* should be proportional to $C(t)$:

$$r_i(t) = \begin{cases} -k_1 \cdot a_i(s_i(t), h(t)) \cdot C(t), & \text{if } B1 < |C(t)| < B2 \\ -k_2 \cdot a_i(s_i(t), h(t)) \cdot C(t), & \text{otherwise} \end{cases} \tag{2}$$

where $B1$ and $B2$ are predefined constants, k_1 and k_2 are adjustable coefficients. We use k_1 and k_2 to control the payoff according to $|C(t)|$. If we set $k_1 > k_2$, it means we hope to encourage less difference between the numbers of losers and winners. On the contrary, if $k_1 < k_2$, we hope the difference fall out of the range {B1, B2}, that is, the more loser the better. The total payoff of MG is:

$$R(t) = \sum_{i=1}^{m} r_i(t) \quad (3)$$

where m is the number of winners.

Therefore, after each cycle of MG, the virtual value of a strategy is updated by the following equation:

$$V_{is}(t+1) = V_{is}(t) + R(t) \quad (4)$$

where V_{is} is the virtual value of a strategy s for player i. In the description of MG, players always select the strategy with the highest virtual value to make their next decision.

3.2 SRS Scenarios using MG

As we known from the above, MG usually provides two choices, and each player needs to choose one of the choices. We need to assign meaningful roles to these choices according to different scenarios in SRS, such as:

1. To act as defender or attacker;
2. To coordinate with others or just select the best situation for itself;
3. To act as a commander to others or to obey commands from others

3.3 The Evaluation Function of SRS

How can we evaluate whether MG strategies is effective in SRS? We need to construct a function to evaluate the performance of a team. During a soccer match, we can examine the following statistics:

$m_f(t)$: the times of failure ball-controlling;

$m_s(t)$: the numbers of successful shots on goal;

$m_i(t)$: the numbers of successful interception;

$t_c(t)$: the total time when the ball is under control;

$t_h(t)$: the total time when the ball is in our half yard;

$s(t) = \sum s_i(t)$: the summation of agents' stamina in a team.

There are some facts that can be derived from real-world soccer games, such as, the longer the time to control a ball, the more effective the shots, the more successful the interceptions, and the better the performance of a team. We are not simply to make our team stronger than others and to be a winner in a competition. We aim to explore whether MG can really influence the coordinations among players in SRS.

Therefore we evaluate one aspect of SRS, that is, the ball-controlling time of a team. The ball-controlling time involves several related actions, such as, the role assignment among players, the dynamic formations, the interception ability of an agent, the personal ball-control skill, the pass accuracy, etc. One may wonder that why we choose this aspect of SRS, but not the final score as an evaluation indicator. The final score is a result of several skills, not only the coordinations, but also the personal dribbling and shooting skills that can be trained using Reinforcement Learning. The ball-controlling time can be a more reasonable reflection of coordination effectiveness, i.e., the role assignment effectiveness among agents, but not some personal skills.

We can define the function of ball-controlling time $BC(t)$ as follows:

$$BC(t) = \sum_{\tau=1}^{t} (\max_{j \in T1}(IT_j(\tau)) > \max_{j \in T2}(IT_j(\tau))) \quad (5)$$

where t is the current time step in SRS, $\max()$ means the maximal value of a collection, $IT_j(\tau)$ is the interception possibility for a player j at time step τ, $T1$ is the team for which we calculate the $BC(T)$, $T2$ is the opponent team. We can easily recognize from above equations that, if a player has the maximal interception possibility of the ball, the teams that this player belongs to is in control of the ball.

3.4 The MG Formulation in SRS

From the above subsections, we know how to formulate MG, and how to evaluate the current status of a SRS team. In this subsection, we will propose a formulation of MG strategies to SRS.

According to the introduction of MG in Section 2.1, each strategy has an intrinsic value, called virtual value, which is defined as the total number of times the strategy has predicted the right selection. At the beginning of a game, each player is equipped with a limited set of S strategies; then it learns to use the best of them, that is, it uses the strategy with the highest virtual value.

In Section 3.3, we defined an evaluation function of the ball-controlling time in SRS: $BC(t)$.

Let us derive $\triangle BC(t)$ from $BC(t)$:

$$\triangle BC(t) = BC(t) - BC(t-1) \quad (6)$$

If $\triangle BC(t) = -1$, it means our team loses the ball; If $\triangle BC(t) = 1$, it means our team successfully intercepts the ball; while $\triangle BC(t) = 0$, it means the same team is in control of the ball.

So we can modify our rule for updating a strategy in MG, which has been defined in Eq. 4, as follows:

$$V_{is}(t+1) = \begin{cases} V_{is}(t) - \omega_1 \cdot R(t), & \text{if } \triangle BC(t) = -1 \\ V_{is}(t) + \omega_2 \cdot R(t), & \text{if } \triangle BC(t) = 1 \\ V_{is}(t) + \omega_3 \cdot R(t), & \text{if } \triangle BC(t) = 0 \end{cases} \quad (7)$$

where $\omega_1, \omega_2, \omega_3$ are weights of rewards.

From Eq. 7 we note that the virtual value of a strategy of $agent_i$ is not only dependent on the result of MG, but also dependent on the evaluation value of the current status in SRS.

4 Experimental Design

In a SRS team, if all the agents are doing the same thing at the same time, it is a large waste of resources. In most of SRS teams, agents are separated into two types, attackers and defenders. In a specific situation, they should try to hold different positions and to execute different actions based on their pre-assigned roles and the real-time evaluation of the situation. In the TsinghuAeolus team, they are using some statistics from experiences and from the learning through the neural network.

In our research, we use MG to dynamically assign *attacker* or *defender* roles to agents, that is, let agents decide to take *defensive* activities or take *attacking* activities in the course of the game. The relationship between MG and the actions of an agent is:

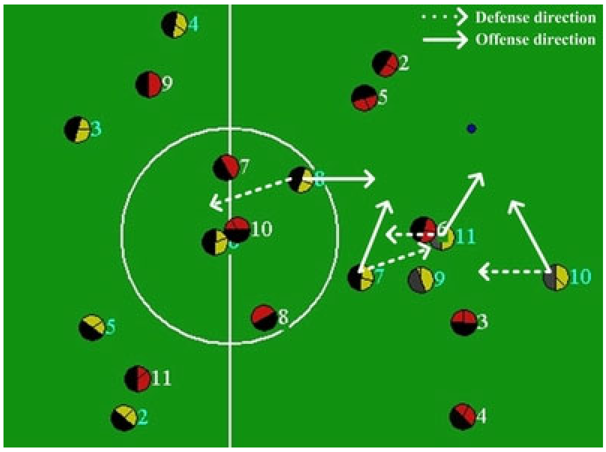
$$s_i(t) = \begin{cases} \text{offense} & \text{if } a_i(t) = -1 \\ \text{defense} & \text{if } a_i(t) = 1 \end{cases} \quad (8)$$

where $s_i(t)$ is the selected strategy for agent i in time step t, and $a_i(t)$ is the result of player i in cycle t in MG.

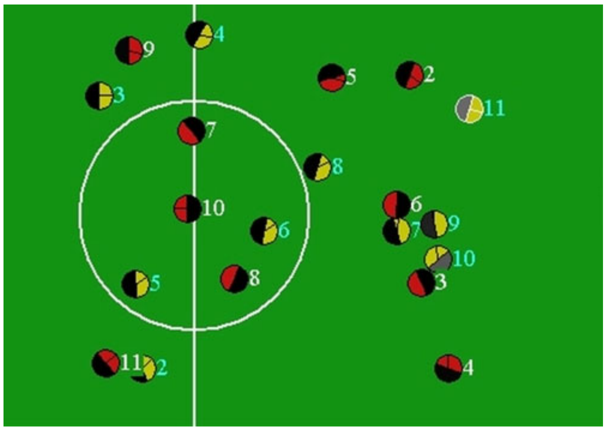
Figure 2 shows an illustration of how to use the strategy to guide agents' actions.

Each agent is equipped with a set of strategies randomly chosen at the beginning, as we have described in the introduction of MG in Section 2.1. During the process of a competition, using the evaluation value or the current status, the agent has a set of ranked strategies. The rank order is changing dynamically after each cycle. The action of an agent is guided by the strategy with the highest value, and the value of a strategy is updated every cycle based on the results of MG and the evaluation value of SRS. Therefore we can observe how MG dynamically assists the decision-making in SRS.

We also need to examine whether the important factor M (the memory size) in MG will influence the result, and how much it will influence.



(a) Agents' positions at time step $t = 71$



(b) Agents' positions at time step $t = 75$

Figure 2. (a) The positions of agents in a competition between TsinghuAeolus and BUTT, at time step $t = 71$. The decisions of $agent_1$ to $agent_{11}$ in the current cycle of MG were 0,0,1,1,0,1,1,0,1,1,0, respectively. According to Eq. 8, $agents\ 1, 2, 5, 8, 11$ selected to take offensive actions, while $agents\ 3, 4, 6, 7, 9, 10$ decided to take defensive actions. We use solid-line arrows and dash-line arrows for showing the directions of offensive actions and defensive actions, respectively. To simplify the illustration we took $agents\ 7, 8, 10, 11$ as examples. (b) The positions of agents at the next cycle $t = 75$. We can note that $agents\ 8, 11$ went in offensive directions, and $agent\ 7, 10$ went in defensive directions.

Our program BUTT, which is equipped with MG strategies in the role assignment, is based on the published source code of TsinghuAeolus. Using TsinghuAeolus as a benchmark, we can compare the performance of these two teams. In our work, we use the ball-controlling time to determine the rewards of players in MG, so we will compare the two teams on this special aspect.

5 Experimental Results

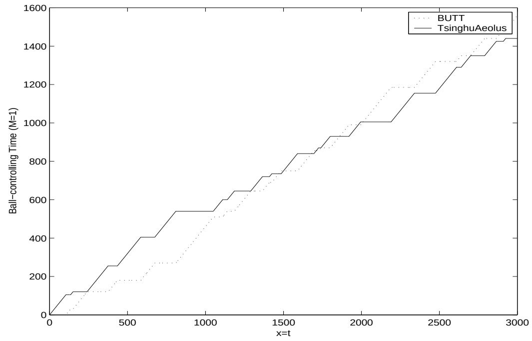
1. Figures 3(a), (b) and (c) show that by increasing the memory size M, the ball-controlling time is increasing, which shows the improvement of the performance of a SRS team. We first let the players using memory size $M = 1$, which means the random position selection during the match. Then we increase the memory size to $M = 2, 3, 5$. We can see a larger brain size makes the agents smarter than a smaller brain size $M = 1$ which can do better in controlling the ball. When $M = 3, 5$, the improvement on the performance becomes obvious.

2. In Figure 3(d), we present the curves of the ball-controlling time during the competition between TsinghuAeolus and my program BUTT. In this figure, the situation is changing as the competition goes on. At the beginning, BUTT does not perform well, and the ball is mostly controlled by TsinghuAeolus. As time goes by, BUTT performs much better than before and begins to control the ball with a longer time than TsinghuAeolus.

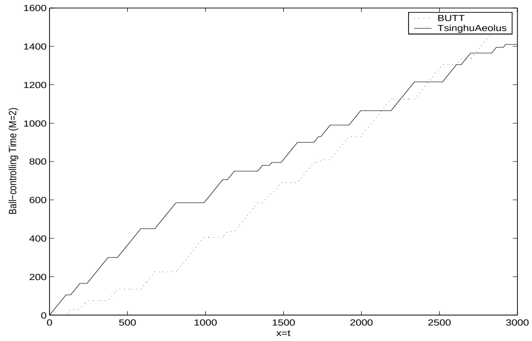
3. In order to observe a long term performance of agents who play SRS with MG, we modified the **half-time** parameter of the SRS server, in this way agents can play a soccer game in 6000 time steps. The improvement of our ball-controlling time have been drawn in Figure 4.

   From Figure 4 the improvement of BUTT becomes more and more obvious in a nonlinear fashion. As time goes by, each player of BUTT adjusts its virtual value of its strategies according to the evaluation of the current situation. Thus the movement of players are more and more reasonable and the role assignments are more and more effective. From the above results, we can demonstrate that, by adopting MG in SRS, the role assignments among agents can be dynamically improved by optimizing their strategy selections.

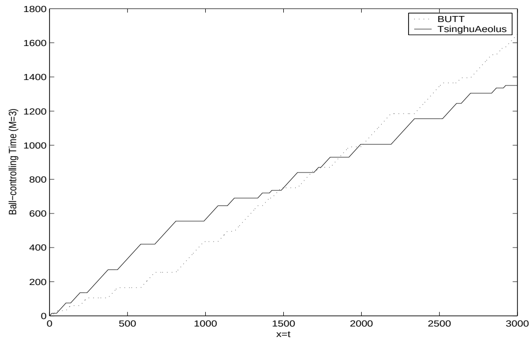
4. From the above experiments, it is clearly that the memory size of an agent in MG is related to its performance in SRS. Does the performance always enhance as the M increasing?



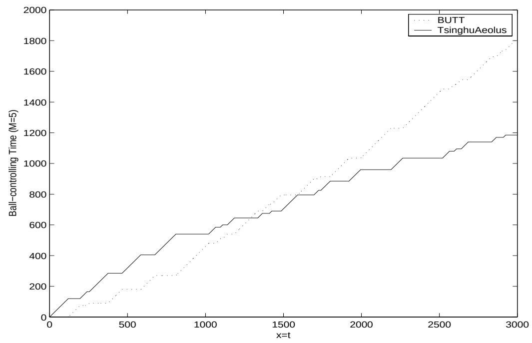
(a) $M = 1$



(b) $M = 2$



(c) $M = 3$



(d) $M = 5$

Figure 3. Ball-controlling time in SRS competition, where $x - axis$ denotes the time step in one game. (a)$M = 1$; (b)$M = 2$;(c)$M = 3$;(d)$M = 5$;

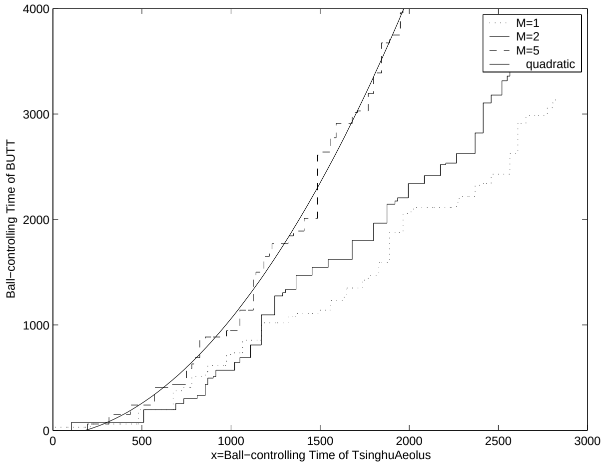


Figure 4. The ball-controlling time of BUTT vs. TsinghuAeolus in the case of $M = 1, 2, 5$.

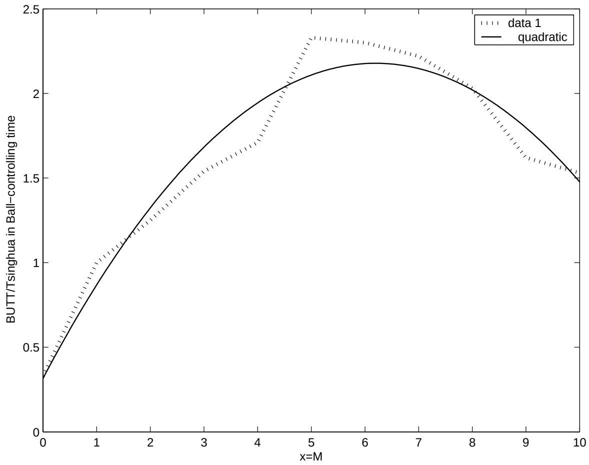


Figure 5. The ratio of ball-controlling time by BUTT vs TsinghuAeolus using different memory sizes.

In Figure 5, we plot the ratio R of BUTT versus TsinghuAeolus using different memory size M.

$$R = \frac{BCT_{BUTT}}{BCT_{TsinghuAeolus}}$$

where BCT is the ball-controlling time of a team in a competition. All the points are the average values of five experiments. We can see that, when $M = 5, 6, 7$, the performance of our team reaches a peak value, and the further increasing of the memory size cause a deduction of the performance of our team. That's to say, the memory size (M) of an agent is not the larger the better. Too large M will deduce the deterministic trend of the game.

Why the performance of a team will not always increase as the memory size increases? We can try to analyze this phenomenon using some analytical results by Challet and Zhang. They have explained the phase transition phenomena in MG (as shown in Figure 7) in paper [5]. Here,

$$\rho = 2^M/N \tag{9}$$

and

$$\sigma = \frac{1}{T}\sum t = 0^T (N_0(t) - \frac{N}{2})^2 \tag{10}$$

where $N_0(t)$ is the number of players selecting side A as time step t, and N is the total number of players in MG.

As they described, the strategy space of MG can be divided into three phases. One is $\rho << \rho_c$ region, which means a crowded phase (Here ρ_c is the X-coordinate where σ^2 is minimal.) In this region, $\sigma^2/N^2 \propto 1/2^M$. If N is constant, the system will differ by different M. The second is $\rho \sim \rho_c$, where the fluctuations are minimal. The third one is $\rho >> 1$ region, in which $\frac{\sigma^2}{N} \cong \frac{1}{4} - C(< d > -\frac{1}{2})$, where $< d >$ is the a average actual distance of strategies between the players.

It is obvious that in Challet and Zhang's model, the σ^2 is the less the better in terms of Eq. 10. From this equation, the less the fluctuation around $\frac{2}{N}$, the smaller the σ value, showing that the better performance of a MG system. However, in our model, we try to maximize the effectiveness of the coordination among agents, i.e., the ball-controlling time. From Figure 6, when using the strategy set $S = 5$, while our team get the maximal ball-controlling time, the corresponding σ is about 2.014. From Figure 7, we can find that, when $\sigma \approx 2.014$, $M \approx 5$, which fits our experimental result in Figure 5.

From the above figures and analysis, we can conclude that, MG can have agents in SRS dynamically select their roles so as to be adaptive to the changing environment. Thus MG can improve the coordination effectiveness among agents in a competitive situation. Furthermore, some intrinsic characteristics of MG are also found in the performance transition of agents in SRS, such as the phase transition phenomenon.

6 Conclusion

In this paper, we described a challenging domain in multi-agent research, the role assignment problem and a recently emerged game theory, the Minority Game (MG). Due to the similar characteristics between these two domains, we

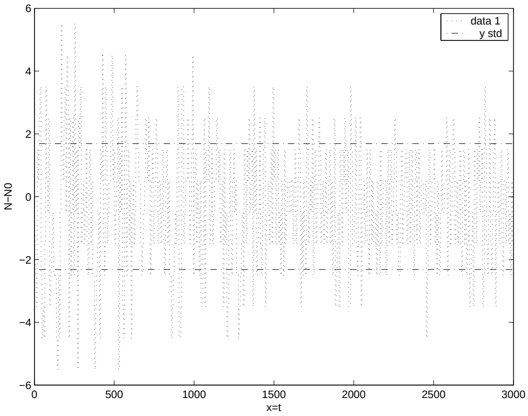


Figure 6. The fluctuations of the number of winners in MG while the players are playing a SRS competition. N is the number of winners in MG, and N_0 is the half of total players.

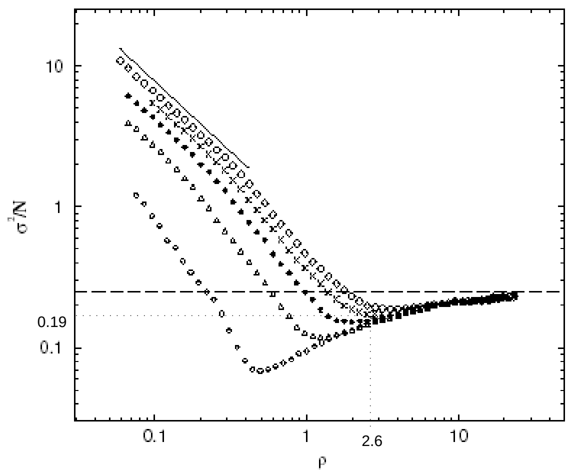


Figure 7. Dependence of σ^2/N on ρ for $S = 2, 3, 4, 5$ and 6 (circles, triangles, stars,x and diamonds) [5]. The dashed one represents the random case's performance. The dotted line indicates the correspondence between the M and σ^2 in our model.

formulated MG strategies to the role assignment problem in a particular multi-agent system: Simulation Robot Soccer (SRS). We then presented the experimental results, comparing our team (with MG strategies) to TsinghuAeolus (without MG strategies) with their coordination effectiveness.

From the obtained results, we note that MG strategies really help agents in SRS make their role assignments adaptive to their changing environment, making the best use of limited resources. Also we observed that there are some important factors affect the performance of MG strategies, such as the memory size (M). The intrinsic phase transition phenomenon related with M is also be observed while agents coordinate in a competitive environment.

As we know, in the previous work, players in MG are selfish. Each player wants to win the game without coordinating with the others to achieve a common goal. Although after several cycles, there appear some collective behaviors in the system, these behaviors are not goal-directed.

In our work, agents with MG strategies can coordinate to achieve a common goal, i.e., to control the ball as long as possible, by adjusting the points of their strategies proportional to the real-time situation evaluation value in SRS. This gives us a new way to study MG in real competitive games, and also provides the research on multi-agent systems with a new method to dynamically improve the decision-making process.

References

[1] What is robocup. *http://www.robocup.org/overview/21.html*.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, 2002.

[3] D. Challet, M. Marsili, and Y.-C. Zhang. Modeling market mechanism with minority game. *Physica A*, 276:284–315, 2000.

[4] D. Challet and Y. C. Zhang. Emergence of cooperation and organization in an evolutionary game. *Physica A*, 246(3-4):407–418, 1997.

[5] D. Challet and Y.-C. Zhang. On the minority game: Analytical and numerical studies. *Physica A*, 256:514–532, 1998.

[6] L. Ein-Dor, R. Metzler, I. Kanter, and W. Kinzel. Multi-choices minority game. *Physical Review E*, 63(066103), 2001.

[7] C. Foldy, Z. Somogyvari, and P. Erdi. Hierarchically organized minority games. *Physica A*, 323:735–742, 2003.

[8] S. Hod and E. Nakar. Self-segregation vs. clustering in the evolutionary minority game. *Physical Review Letters*, 88, 238702, May 2002.

[9] B. Huberman and R. Lukose. Social dilemmas and internet congestion. *Science*, 277(5325):535–537, July 1997.

[10] H. Kitano. Robocup: The robot world cup initiative. *Proceedings of the First International Conference on Autonomous Agent (Agents-97), Marina del Ray, The ACM Press*, 1997.

[11] S. Moelbert and P. D. L. Rios. The local minority game. *Physica A*, 303:217–225, 2002.

[12] S. Wolfram. *Theory and Applications of Cellular Automata*. World Scientific, 1986.