Proceedings of the 2007 IEEE Symposium on
Computational Intelligence and Games (CIG 2007)

Reward Allotment Considered Roles for Learning Classifier System For Soccer Video Games

Yosuke Akatsuka
Hosei University
3-7-2 Kajino-cho Koganei-shi Tokyo
184-8584 JAPAN
E-mail: yousuke.akatsuka.dn@gs-cis.hosei.ac.jp

Yuji Sato
Hosei University
3-7-2 Kajino-cho Koganei-shi Tokyo
184-8584 JAPAN
E-mail: yuji@k.hosei.ac.jp

***Abstract*- In recent years, the video-game environment has begun to change due to the explosive growth of the Internet. As a result, it makes the time for maintenance longer and the development cost increased. In addition, the life cycle of the game program shortens. To solve the above-mentioned problem, we have already proposed the event-driven hybrid learning classifier system and showed that the system is effective to improving the game winning rate and making the learning time shorten. This paper describes the investigation result of the effect in case we apply the reward allotment considered each role for classifier learning system. Concretely, we investigate the influence to each player's actions by changing the algorithm of the opponent and to team strategy by changing reward setting, and analyze them. As a result, we show that the influence of learning effects to each player's actions does not depend on the algorithm of opponent. And we also show that the reward allotment considered each role has possible to evolve the game strategy to improving the game winning rate.**

Keywords: Soccer game, Video game, Learning classifier system, Event-driven, Reward allotment considered roles

I. Introduction

It is common in the production of video games for human designers to explicitly specify the decision-making algorithms to be used by game agents. It is also common to use IF-THEN type of production rules as a format for describing these algorithms. This is because production rules of this type make it relatively easy to describe algorithms at design time and to understand them during maintenance. Game programs developed by this production technique have achieved positive results based on a fixed usage environment.

In recent years, however, the video-game environment has begun to change due to the explosive growth of the Internet. Because of the Internet, it is becoming increasingly easier for anyone to use video games, and the number of game users is increasing dramatically as a result. User knowledge is also diversifying ranging from children to adults playing levels. These developments have two main consequences. First, a single algorithm cannot possibly satisfy all users, and as the number of users increase, differences in strategies that users prefer and excel in can no longer be ignored. The need is therefore felt for simultaneous support of multiple strategy algorithms. Second, the appearance of users with advanced techniques has generated a need for decision-making algorithms under even more complicated environments. And finally, as the Internet makes it easy for new users to appear one after another, it must be possible to provide and maintain bug-free programs that support such complex decision-making algorithms in a time frame much shorter than that in the past.

As a means for addressing the above problem, taking a soccer game as an example of a video game, we have already proposed a learning scheme [17] that considers hybrid systems and events when applying a classifier system [4] to the acquisition of decision-making algorithms by soccer in-game agents. Moreover, it was shown that there was a possibility that the improving of game winning rate and settling of learning by the allocating rewards considered each role of the forward(FW), midfielder(MF), and defense(DF) [16]. This paper describes the investigation result of the effect in case we apply the reward allotment considered each role for classifier learning system.

Fig 1. Example of a typical game scene targeting the area around the current position of the ball.

II. CONVENTIONAL SOCCER VIDEO GAME AND ASSOCIATED PROBLEMS

The type of soccer game that we will deal with here is a software-driven video game with soccer as its theme in which two teams battle for the most points. Figure 1 shows a typical game scene targeting the area around the current position of the ball. The screen also includes a diagram showing a total view of the game in the lower right hand corner. Each team has 11 players, and the movements of the 11 players of one team are controlled by computer. The algorithm to control player action is thought up beforehand by a game designer and programmed as a set of control rules in IF-THEN (condition-action) format.

As described above, the conventional approach to producing a soccer video game is to have a game designer devise the algorithm for controlling player action and to then describe and program that algorithm as a set of rules in IF-THEN format. For a fixed usage environment, this approach has produced positive results. This is because a new algorithm could be devised before users lost interest in the current algorithm set up beforehand on the game-maker side, and because a program written in IF-THEN format could be easily understood and maintained.

Recently, however, the Internet is making it easier for anyone to participate in video games and the number of game users is increasing as a result. This development is generating a whole new set of problems. First, the increasing number of users means that the differences in strategies that users prefer and excel in can no longer be ignored and that multiple strategy algorithms must be simultaneously supported. Second, the appearance of users with advanced techniques has generated a need for decision-making algorithms under even more complicated environments. And finally, as the Internet makes it easy for new users to appear one after another, it must be possible to provide and maintain bug-free programs hat support such complex decision-making algorithms in a time frame much shorter than that in the past. In other words, the human- and time-related resources required by development and maintenance work are increasing dramatically while the life cycle of each game is shortening. The conventional technique is hard pressed to deal with this situation.

III. AN EVENT-DRIVEN BUCKET BRIGADE LEARNING METHOD AND THE ALLOTMENT OF REWARDS

A. Hybrid Decision-making System

We have studied the equipping of game programs with machine learning functions as an approach to solving the above problems. This is because incorporating machine learning functions in an appropriate way will enable the system to learn the game player's strategy and to automatically evolve a strong strategy of its own. It will also eliminate worries over program bugs and significantly reduce the resources required for development and maintenance. A number of techniques can be considered for implementing

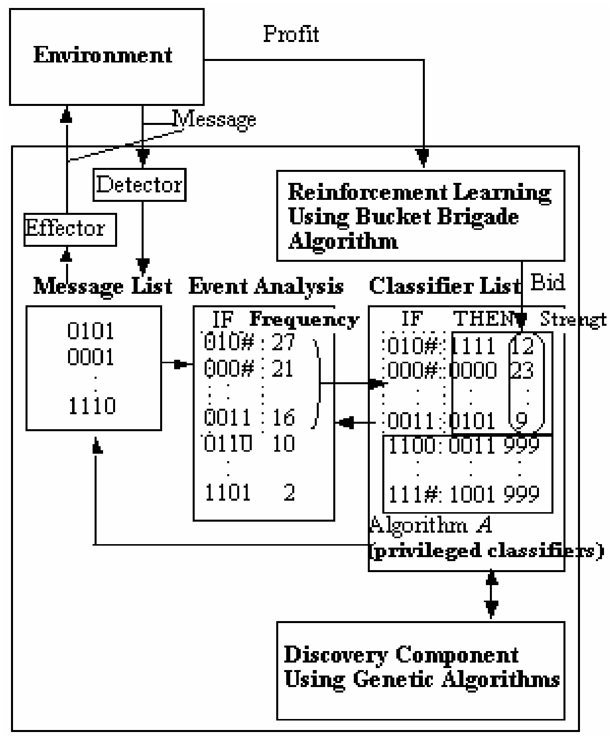

Fig 2. The configuration of the event-driven hybrid learning classifier systems.

machine learning functions such as neural networks, Q-learning [16] and genetic algorithms (GAs), and we have decided, in particular, on incorporating functions for acquiring rules based on classifier systems. We came to this decision considering the many examples of applying evolutionary computation to the acquisition of robot decision-making algorithms [10, 11, 18] in the world of robot soccer games such as RoboCup [8, 15], learning classifier systems takes advantage of GAs and reinforcement learning [16] to built adaptive rule-based systems that learn gradually via online experiences [6, 7, 9], and considering the compatibility between the IF-THEN production-rule description format and classifier systems and the resulting ease of program migration.

At the same time, the bucket brigade algorithm [1, 2, 5, 12-14] used as a reinforcement learning scheme for classifier systems needs time to obtain an effective chain between classifiers. As a result of this shortcoming, the bucket brigade algorithm is not suitable for learning all strategies from scratch during a game. A conventional algorithm, on the other hand, provides solid strategies beforehand assuming fixed environmental conditions, but also includes a rule that states that a player encountering undefined environmental conditions must continue with its present course of action. In light of the above, we decided to apply classifier-based learning to only conditions/actions not described by an explicit algorithm. In short, we adopted a hybrid configuration combining a conventional algorithm and learning section using a classifier system.

B. Event-driven Learning Classifier System

The preliminary experiments revealed that a hybrid-type system has the potential of exceeding a human-designed algorithm provided that search space can be contracted by limiting the target of learning to actions. On the other hand, having humans select conditions beforehand does nothing to eliminate the problems associated with the conventional way of generating conditions.

To solve this dilemma, we decided to switch the rules to be learned for each game player (user) that the computer opposes. This is because the total possible search space in theory need not be the target of learning if only the strategy of the game player in the current match can somehow be dealt with. Furthermore, it was decided that all of the current player's strategies would not be targeted for learning but rather that the number of events targeted for learning would be limited to that that could be completed in real time. Figure 2 shows the configuration of the proposed event-driven classifier system. This system differs from standard classifier systems in three main ways. First, the proposed system adds an event analysis section and creates a table that records event frequency for each game player. Second, the classifier discovery section using genetic algorithms targets only actions while conditions are generated by adding new classifiers in accordance with the frequency of actual events. Third, the system updates the strength of classifiers by the bucket brigade algorithm starting with high-frequency event and continuing until learning can no longer be completed in real time. The proposed system also adopts a hybrid configuration combining a conventional algorithm and classifier system as before. Finally, the system provides for two types of rewards that can be obtained from the environment: a large reward obtained from winning or losing a game and a small reward obtained from succeeding or failing in a single play such as passing or dribbling the ball. In short, the above system focuses only on strategy that actually occurs with high frequency during a game and limits learning space to the range that learning can be completed in real time.

IV. Influence of Reward Allotment to Any Actions

A. Reward allotment based on the role of each position

In real soccer games, the forward, midfielder and defense players are assigned different roles and emphasize different aspects of their play depending on these assigned roles. Accordingly, it is thought that giving different success rewards to each player considering the role assignments of forward midfielder and defense players might lead to a better game winning rate. These role assignments into consideration might lead result in cooperative learning that contributes to a better winning rate. In preliminary experiments, we revealed that giving different success rewards to each player considering each player's role has a possible of leading to a better game winning rate.

In here, we investigate the influence of the reward allotment considered each role more in detail. Concretely, we investigate 1) whether or not there are influences to each player's action by changing the algorithm of opponent, 2) what effect change of the reward values gives to the team strategy (it appears as a combination of each player's actions).

B. Relation between each player's action and the algorithm of opponent

First, we investigate the influence that difference of opponent's algorithm gives to each player's action. There are three algorithms to be used as opponents as follows.

Algorithm A: Algorithm that has well-balanced offence and defense. This is the basic strategy.

Algorithm B: Algorithm that has offence mainly. Strategy that gather the ball in players located on side of field, and takes the offensive positively.

Algorithm C: Algorithm that has defense mainly. It is prioritizing the stop of the other party's attack, and so as not to be off its guard.

The reward setting considered roles at each position is assumed to be Role Considered Reward Setting (RCRS). Table 1 shows the reward setting of RCRS concretely. The future of RCRS is, 1) the player that is position of FW gives priority to making the shot succeed, the player that is position of MF gives priority to the action that carries the ball (dribble and pass). The player that is position of DF gives priority to steal the ball from the opponent. Imitating the soccer of the reality, and the reward values thought to be efficient to learn.

Table 1.
Reward values of RCRS

	FW	MF	DF
GETGOAL	80	60	40
DRIBBLE	2	4	2
PASS	8	16	8
GETBALL	5	15	45
LOSTBALL	-50	-50	-50
TOTAL	45	45	45

Table 2.
Reward values of RNCRS

	FW	MF	DF
GETGOAL	60	60	60
DRIBBLE	4	4	4
PASS	16	16	16
GETBALL	15	15	15
LOSTBALL	-50	-50	-50
TOTAL	45	45	45

C. Influence of changing the reward values to team strategies

Second, we investigate the influence to the team strategy by changing the reward values for verifying the factor of the improvement of the game winning rate. Examining what decision-making appears is concretely from success rate, success frequency, and trial frequency of each player's actions.

Experiment is evaluated by comparing the experiment result in case applying the reward setting of Role Not Considered Reward Setting (RNCRS) which newly prepared is different from the above-mentioned reward values RNRS (defined above). Table 2 shows the reward values of RNCRS concretely.

V. EVALUATION TRIALS

A. Experimental environments

22 soccer players are connected to the experimental environment, and the hybrid type decision making system or the algorithm type decision making system is applied to each player as a decision-making system. Using this experimental environment, the team that adopts the hybrid type decision making system has a game with the team that adopted the algorithm type decision making system. The game repeats 200 games 30 times. The action rules of each player that is applied the hybrid type decision making system is updated once per 20 games by applying the operation of genetic algorithm, such as crossover, mutation.

B. Experimental results

1) Relation between each player's action and the algorithm of opponent

Figure 3 and Figure 4 show the shot success frequency of the player in each position in case algorithm A and algorithm C are the opponents. In these figure, using algorithm C as the opponent, compared with algorithm A, we can find that the shot success frequency of the player of FW reduce by half. In addition, some decreases are similarly seen for the player of MF and DF. In case using algorithm B as the opponent, compared with algorithm A, the shot success frequency of the player of FW reduce a little, but the frequency of player of MF and DF are almost same.

Figure 5 and figure 6 show the pass trial frequency of the player in each position in case algorithm A and algorithm C are the opponents. In these figure, using algorithm C as the opponent, the pass trial frequency of the player of FW and MF increase, on the other hand, the value of the player of DF is higher when compared with algorithm A of the player of DF. Especially, it is remarkable that the increasing of the pass trial frequency of the player of the FW in case algorithm C is the opponent. In case using algorithm B, the experimental result was almost same as the result of using algorithm A as the opponent. Moreover, using any algorithm, it is found the tendency that the FW's pass trial frequency is the most, the next is MF's, and the last is DF's.

Figure 7 and 8 show the dribble trial frequency of the player in each position in case algorithm A and algorithm C are the opponents. In these figure, using algorithm C as the opponent, it is found that the dribble trial frequency of the FW and MF are greatly decreased. And it is also found that the DF's is a little decreased similarly. In case using algorithm B, the FW's is a little higher when compared with algorithm A, on the other hand, there is little difference to MF's and DF's. About the dribble frequency, using any algorithm, it is found the tendency that the FW's is the most, the next is MF's and the last is DF's.

Figure 9 and 10 show the ball steal frequency of the player in each position in case algorithm B and algorithm C are the opponents. In these figures, using algorithm C, compared with algorithm B, it is found that the MF's ball steal frequency increases. About the FW's, it is higher value in case using algorithm B than using algorithm C. In case using the algorithm A as an opponent, compared with algorithm B, the FW's ball steal frequency is less, on the other hand, the MF's is greater. Using any algorithm, it is found the tendency that the MF's ball steal frequency is the most, and the next is DF's, and the last is FW's.

2) Influence of changing the reward values to team strategies

The experimental result applying the RCRS and RNRCS is shown below. In these experiment, the algorithm A is used as the opponent.

Figure 11 and 12 show the shot success frequency of the player in each position in case applying the both reward settings. In these figures, in case using RNRS, compared with RNCRS, we can find that the FW's shot success frequency is higher. There is little difference to the MF's and DF's.

Figure 13 and 14 show the pass trial frequency of the player in each position in case applying the both reward settings. In these figures, in case applying RNCRS compared with RCRS, it is found that FW's pass trial frequency is higher. There is little difference to the MF's and DF's similarly the experimental result of shot success frequency. On the other hand, there is little difference when which reward setting was applied for the pass success rate.

Figure 15and 16 show the dribble success rate of the player in each position in case applying the both reward settings. In these figures, in case applying RCRS compared with RNCRS, it is found that FW's dribble success rate is higher. And MF's is also a little higher in case applying RCRS.

Figure 17 and 18 show the ball steal frequency of the player in each position in case applying the both reward settings. In these figures, we can find that MF's and DF's ball steal frequency is a little higher in case applying RCRS than the case applying RNCRS. On the other hand, FW's ball steal frequency is lower in case applying RCRS than the case applying RNCRS.

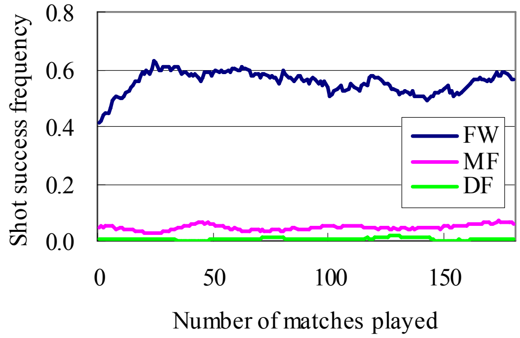

Fig 3. Shot success frequency of each player at each position (Against algorithm A).

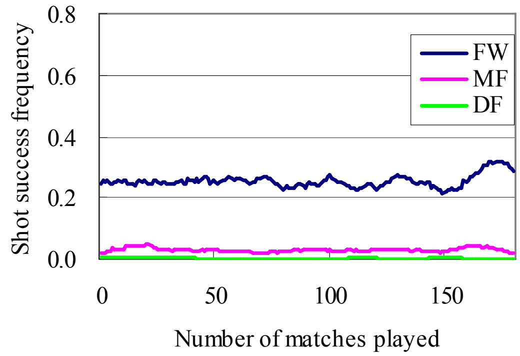

Fig 4. Shot success frequency of each player at each position (Against algorithm C).

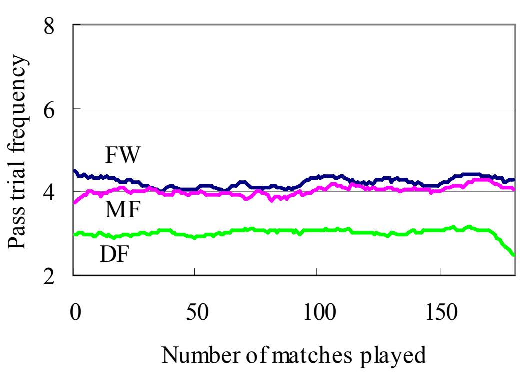

Fig 5. Pass trial frequency of each player at each position (Against algorithm A).

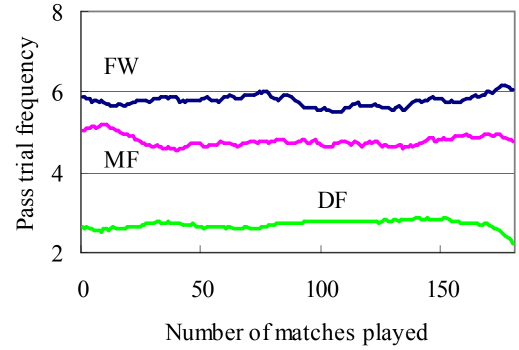

Fig 6. Pass trial frequency of each player at each position (Against algorithm C).

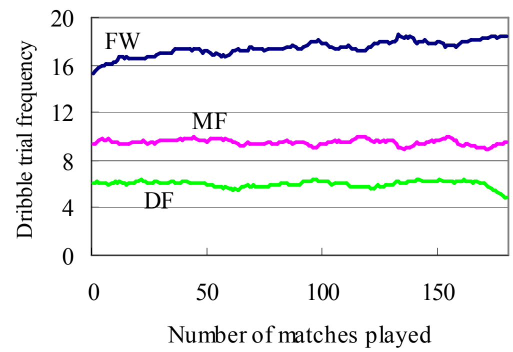

Fig 7. Dribble trial frequency of each player at each position (Against algorithm A).

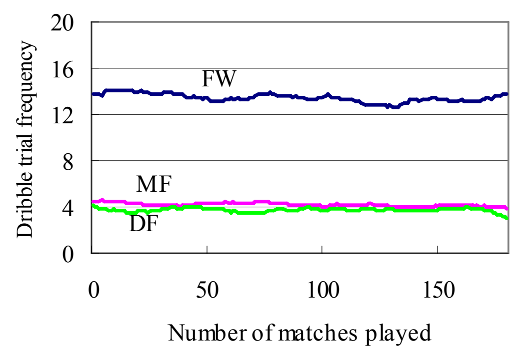

Fig 8. Dribble trial frequency of each player at each position (Against algorithm C).

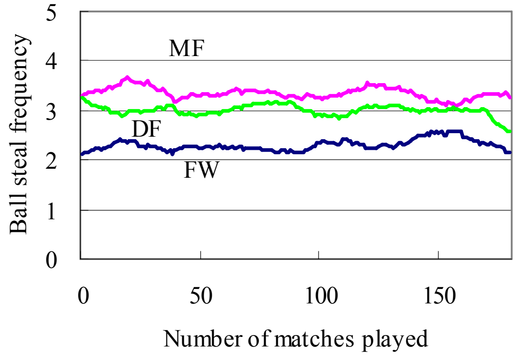

Fig 9. Ball steal frequency of each player at each position (Against Algorithm B).

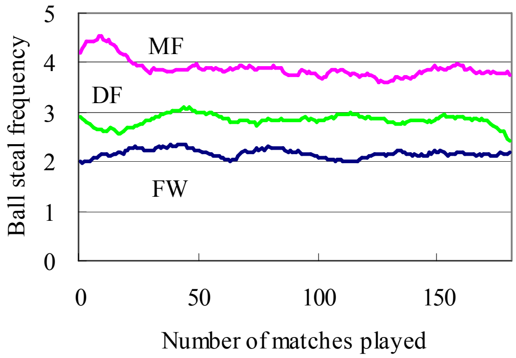

Fig 10. Ball steal frequency of each player at each position (Against Algorithm C).

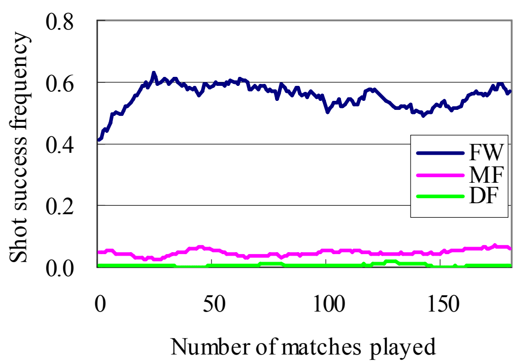

Fig 11. Shot success frequency when adopting RCRS.

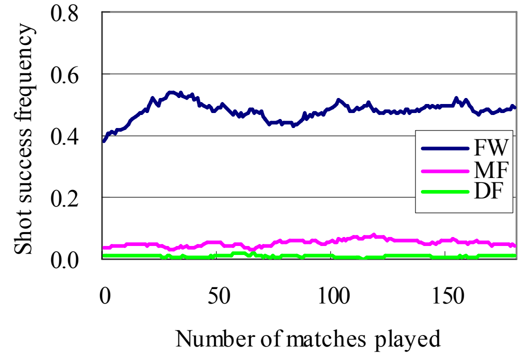

Fig 12. Shot success frequency when adopting RNCRS.

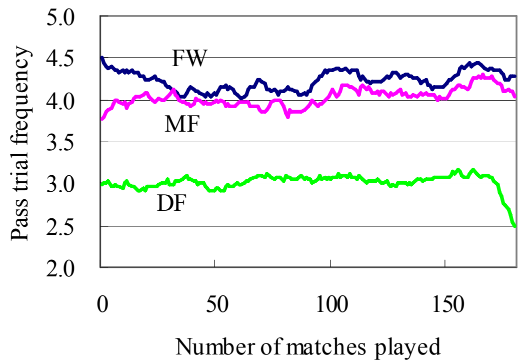

Fig 13. Pass trial frequency when adopting RCRS.

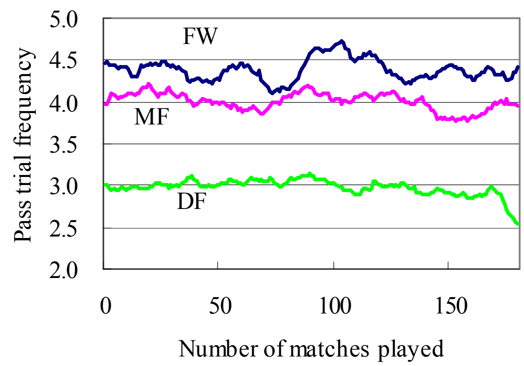

Fig 14. Pass trial frequency when adopting RNCRS.

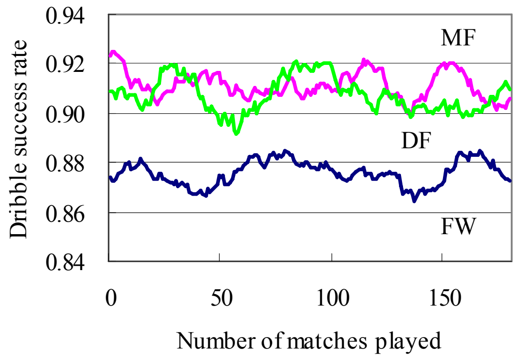

Fig 15. Dribble success rate when adopting RCRS.

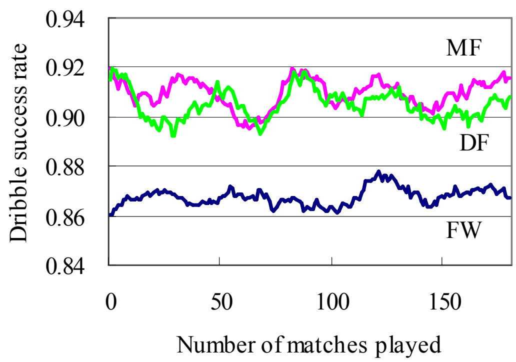

Fig 16. Dribble success rate when adopting RNCRS.

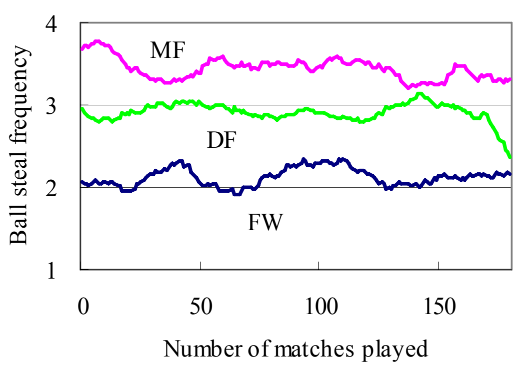

Fig 17. Ball steal frequency when adopting RCRS.

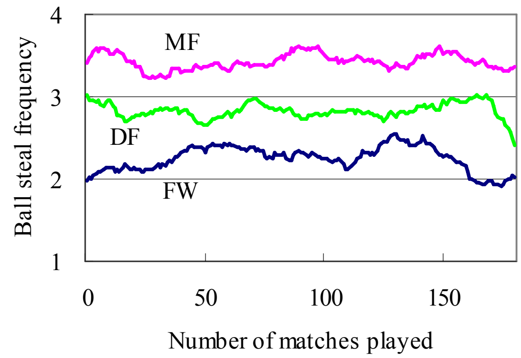

Fig 18. Ball steal frequency when adopting RNCRS.

VI. Discussion

A. Relation between each player's action and the algorithm of opponent

In case using the algorithm A and algorithm B as the opponent, we can found the tendency that the experimental result, especially about the absolute value of each action such as shot, pass, dribble and ball steal, depends on the algorithm of the opponent, on the other hand, the order of the value of the experimental result does not depend on the algorithm of the opponent. From this factor, it is said that the actions of each player does not depend on the algorithm of the opponent. On the other hand, in case using the algorithm C as the opponent, the tendency is different from the algorithm A and algorithm B appears. Therefore, in case using the algorithm C as the opponent, it is thought that the different actions of each player which is different from the case using algorithm A and algorithm B as opponents appears.

The tendency in case using the algorithm C as the opponent is the dribble trial frequency decreasing and the pass trial frequency increasing. This is thought that this decision making is generated because in case against the defensive algorithm (future of algorithm C), it has higher risk to attack by dribbling and is not efficient. The tendency that the game winning rate rises by the dribble success rate high is appeared. From this, it is thought that the algorithm each player dribbles frequently appeared for improving the game winning rate. On the other hand, in case using the algorithm C as the opponent, the tendency that the game winning rate rises by the dribble trial frequency decreasing appears. Therefore, it is thought that the difference of each player's action depends on the algorithm of the opponent appears.

In case using the algorithm C as the opponent, MF's ball steal frequency is increased. From this, it is thought that the difference not only offensive actions but also defensive actions are depended on the algorithm of the opponent.

B. Influence of changing the reward values to team strategies

In case applying the RCRS, FW's dribble failure is decreased. It is said that the rate of FW losing its ball is decreased. It is also observed that the shot success rate is increased. These factors say that the FW player has evolved the action algorithm as use the dribble efficiently, and make the chance of shooting much more. Moreover in case applying RCRS, the tendency that the FW player decreases the pass trial frequency is appeared. From this, it is thought that dribbling or shooting is chosen instead of the action of passing. Moreover, the FW players do not join the defensive actions positively because of the experiment result of the ball steal frequency. In short, it is thought that the strategy that the player of MF and DF contracts the defensive action, on the other hand, the player of FW waits for the ball passing from MF or DF and locates the front line to get goal, is appeared. This strategy is typically used in real soccer. From this, it is said that applying RCRS evolves the team strategy to win the game effectively. Because of this strategy, the FW player can receive the ball at a position near the goal easily, and the shot chance can be obtained in a little dribble frequency.

VII. Conclusion

This paper describes the investigation result of the effect in case we apply the reward allotment considered roles for classifier learning system for soccer video games. Concretely, we investigated the influence to each player's actions by changing the algorithm of the opponent and to team strategy by changing reward setting. As a result, by applying reward allotment that considered roles, it is shown that the cooperating action in team is appeared which depend on the algorithm of opponent and it contribute to increase efficiency to learn. Therefore, there is similar tendency that the influence of reward allotment which considered roles to each player's actions does not depend on opponent algorithm.

References

[1] Belew. R.K and Gherrity. M, "Back Propagation for the Classifier System" In *Proceedings of the Third International Conference on Genetic Algorithms.* Morgan Kaufmann Publishers, CA, 1989, pp. 275-281.

[2] Goldberg. D.E, "Genetic Algorithms in Search, Optimization and Machine Learning" Addison-Wesley, Reading, MA, 1989.

[3] Gustafson. S.M and Hsu. W.H "Layered Learning in genetic Programming for a Cooperative Robot Soccer Problem" In *Proceedings of the Fourth European Conference on Genetic Programming,* 2001, pp. 291-301

[4] Holland. J.H, "Adaptation in Natural and Artificial System". The MIT Press, Cambridge, MA, 1992. The University of Michigan Press, Ann Arbor, 1975.

[5] Holland. J.H, "Escaping brittleness: The possibilities of general-purpose learning algorithms applied to parallel rule-based systems" In Michalski, R.S. et al. (cds.): *Machine Learning II,* Morgan Kaufmann Publishers, CA, 1986, pp. 593-623.

[6] Holmes. J.H, Lanzi. P.L, Stolzmann. W, Wilson. S. W, "Learning Classifier Systems: New Models, Successful Applications" *Information Processing Letters,* Vol. 82, 2002, pp. 23-30.

[7] Huang. C-H and Sun. C-T, "Parameter Adaptation within Co-adaptive Learning Classifier Systems" In *Proceedings of the Sixth Annual Genetic and Evolutionary Computation Conference.* Vol. 2, LNCS 3103, Springer-Verlag, Berlin, Heidelberg, 2004, pp. 774-784.

[8] Kitano. H, Asada. M, Kuniyoshi, Y. Noda, I. Osawa, E, and Matsubara. H, "RoboCup: A challenge problem for AI" *AI Magazine,* Vol. 18, 1997, pp. 73-85.

[9] Kovacs. T, "What Should a Classifier System Learn and How Should We Measure It?" *Journal of Soft Computing,* Vol. 6, No. 3-4, 2002, pp. 171-182.

[10] Luke. S, "Genetic Programming Produced Competitive Soccer Softbot Teams for RoboCup 97" In *Proceedings of the Third Annual Genetic Programming Conference.* Morgan Kaufmann Publishers, San Francisco, CA, 1998, pp. 204-222.

[11] Pietro. A.D, While. L, and Barone. L, "Learning in RoboCup Keepaway using Evolutionary Algorithms" In *Proceedings of the Fourth Annual Genetic and Evolutionary Computation Conference.* Morgan Kaufmann Publishers, San Francisco, CA, 2002, pp. 1065-1072.

[12] Riolo. R.L, "Bucket brigade performance: I. Long sequences of classifiers, genetic algorithms and their application" In *Proceedings of the Second International Conference on Genetic Algorithms.* Lawrence Erlbaum Associates, Publishers, 1987, pp. 184-195.

[13] Riolo. R.L, "Bucket brigade performance: II. Default hierarchies" In *Proceedings of the Second International Conference on Genetic Algorithms.* Lawrence Erlbaum Associates, Publishers, 1987, pp. 196-201.

[14] Riolo. R.L, "The emergence of coupled sequences of classifiers" In *Proceedings of the Third International Conference on Genetic Algorithms.* Morgan Kaufmann Publishers, CA, 1989, pp. 256-263.

[15] RoboCup web page. http://www.robocup.org/

[16] Sutton. R.S, Barto. A.G, "Reinforcement Learning: An Introduction" The MIT Press, Cambridge, MA, 1998.

[17] Sato. y, and Kanno. R, "Event-driven Hybrid Learning Classifier Systems for Online Soccer Games" In *Proceedings of the 2005 IEEE Congress on Evolutionary Computation.* IEEE Press, Edinburgh, 2005, pp. 2091-2098

[18] Sato. y, Akatsuka. Y and Nishizono. T, "Reward Allotment in an Event-driven Hybrid Learning Classifier System for Online Soccer Games" In *Proceedings of the 2006 Genetic and Evolutionary Computation Conference*. ACM Press (2006).