Reward Allotment in an Event-driven Hybrid Learning Classifier System for Online Soccer Games

Yuji Sato
Hosei University
3-7-2 Kajino-cho Koganei-shi
Tokyo 184-8584 JAPAN
+81-42-387-4533
yuji@k.hosei.ac.jp

Yosuke Akatsuka
Hosei University
3-7-2 Kajino-cho Koganei-shi
Tokyo 184-8584 JAPAN
+81-42-387-4533
m02k0101@k.hosei.ac.jp

Takenori Nishizono
Hosei University
3-7-2 Kajino-cho Koganei-shi
Tokyo 184-8584 JAPAN
+81-42-387-4533
i04t0014@k.hosei.ac.jp

ABSTRACT

This paper describes our study into the concept of using rewards in a classifier system applied to the acquisition of decision-making algorithms for agents in a soccer game. Our aim is to respond to the changing environment of video gaming that has resulted from the growth of the Internet, and to provide bug-free programs in a short time. We have already proposed a bucket brigade algorithm (a reinforcement learning method for classifiers) and a procedure for choosing what to learn depending on the frequency of events with the aim of facilitating real-time learning while a game is in progress. We have also proposed a hybrid system configuration that combines existing algorithm strategies with a classifier system, and we have reported on the effectiveness of this hybrid system. In this paper, we report on the results of performing reinforcement learning with different reward values assigned to reflect differences in the roles performed by forward, midfielder and defense players, and we describe the results obtained when learning is performed with different combinations of success rewards for various type of play such as dribbling and passing. In 200 matches played against an existing soccer game incorporating an algorithm devised by humans, a better win ratio and better convergence were observed compared with the case where learning was performed with no roles assigned to all of the in-game agents.

Categories and Subject Descriptors

J.0 [**Computer Applications**]: General

General Terms: Design, Experimentation, Verification.

Keywords: Learning classifier systems, Event-driven, Real-time learning, Soccer game, Video-game.

1. 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 [20] that considers hybrid systems and events when applying a classifier system [8] to the acquisition of decision-making algorithms by soccer in-game agents. In this paper, we report on the results of performing tests in which reinforcement learning is performed using different reward values that take the different roles of forward, midfielder and defense players into consideration. Section 2 presents an overview of the conventional soccer game on which this study is based. In section 3 we present the basic concepts involved in giving rewards by considering differences between positions and by considering an event-driven hybrid learning classifier system proposed to realize real-time learning during the game. And in section 4 we present an evaluation method and the results of our tests, and finally we conclude with a summary.

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.

2. CONVENTIONAL SOCCER VIDEO GAME AND ASSOCIATED PROBLEMS

2.1 Overview of Soccer Video Game

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. Figure 2 shows an example of a rule written in IF-THEN format and the corresponding scene. The rule states "If the ball is right in front of me while I am in front of the goal and if two players of the other team are between me and the goal, then pass the ball to an unmarked player on my team." The program for determining player action consists of a detector, a decision-making section, and an effector. Based on information input from the environment, the detector determines the position and state of each player, the position of the ball, the distance between a player and the goal, etc., and passes these results to the decision-making section. This section then determines player actions according to an algorithm described in IF-THEN format as described above. Examples of player actions include kick, trap, and move in accordance with current circumstances. The effector finally executes these actions in the environment based on instructions

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

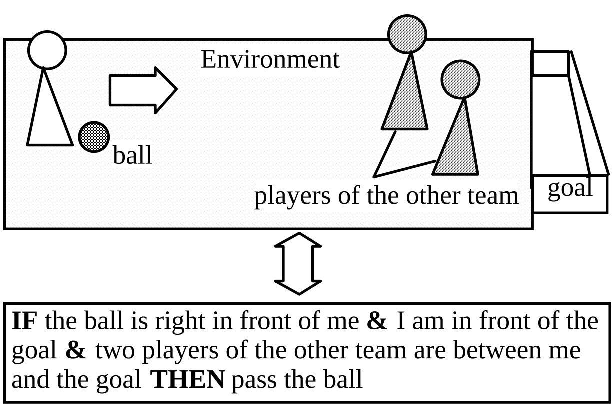


Figure 2. Example of a rule written in IF-THEN format and corresponding scene.

received from the decision-making section. Now, the operation of all or some of the 11 players making up the opposing team is performed by the user, that is, the game player. If the game player is in charge of operating only some of the players on his team, the actions of the remaining players will be controlled by the same algorithm as that of the team controlled by computer.

2.2 Problems with Conventional Technique

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 that 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.

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

3.1 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 machine learning functions such as neural networks, Q-learning [21] 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 [7, 14, 15] in the world of robot soccer games such as RoboCup [12, 19], learning classifier systems takes advantage of GAs and reinforcement learning [21] to built adaptive rule-based systems that learn gradually via online experiences [10, 11, 13], 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 [2, 6, 9, 16-18] 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 a learning section using a classifier system.

Figure 3 shows the basic idea of the hybrid decision-making system using a classifier system. This hybrid system is achieved by embedding a conventional algorithm into a classifier system as a base. The conventional algorithm is unaffected by learning and is implemented as a set of “privileged classifiers.” Specifically, the reliability (credit or strength) of a privileged classifier is set to the highest possible value and is not targeted for updating by learning. If, after analyzing a message list, there are no privileged classifiers in the classifier list that match a current condition, the strength of a classifier that does is updated. Classifiers can also be discovered here using genetic algorithms.

3.2 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.

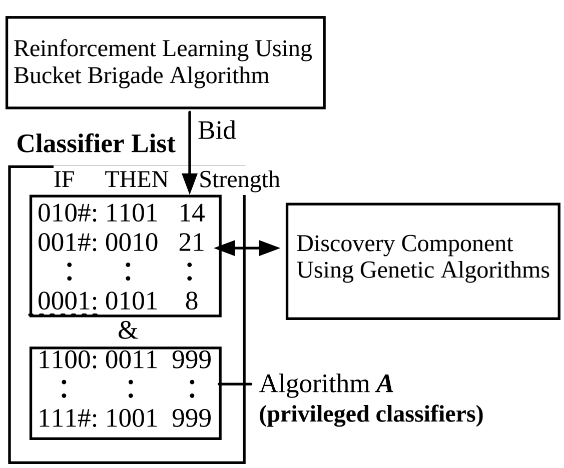


Figure 3. Basic idea of the hybrid decision-making system using a classifier system.

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 4 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 events 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.

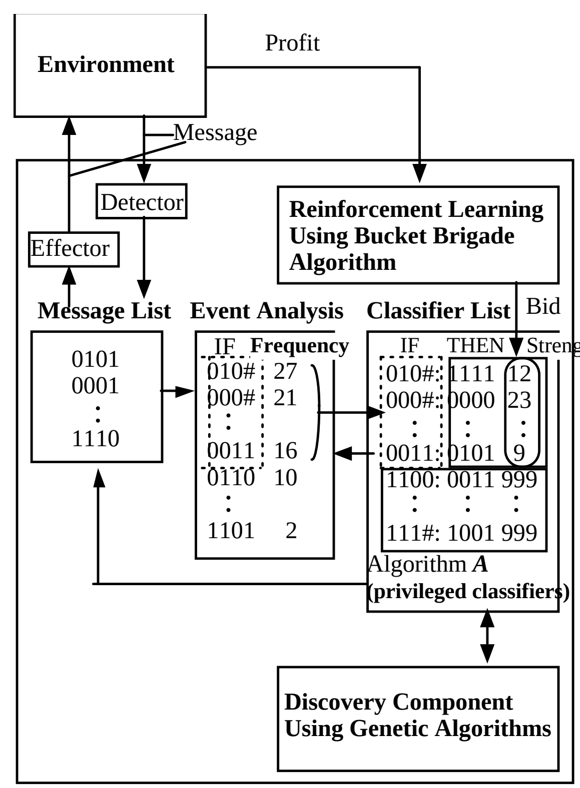


Figure 4. The configuration of the event-driven hybrid learning classifier systems.

Table 1. The success rewards for each play that were used in a team *H*1.

	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

	FW	MF	DF
GETGOAL	◎	○	△
DRIBBLE	○	◎	○
PASS	○	◎	○
GETBALL	△	○	◎
LOSTBALL	△	○	◎

Figure 5. The basic concept for determining the success rewards for each player.

3.3 Reward allotment based on the role of each position

Table 1 shows the success rewards for each play that were used in these tests. Preliminary tests were performed to make a prior survey of the number of times the players performed pass and dribble actions in a single game, on the basis of which the rewards for passing and dribbling were set so that the product of the success reward for passing and the number of passes made was more or less equal to the product of the success reward for dribbling and the number of dribble actions performed. The goal-scoring success reward was set to a high value because goal-scoring is of great importance to the outcome of a game. For all the in-game agents apart from the goalkeeper, learning was performed by applying success rewards to each play without any particular regard to differences in the role of each position. For example, when a pass was made successfully, bucket brigade learning was performed so that the same reward value (16) was obtained by each player irrespective of whether the player was assigned to a forward, midfielder or defense role. And when a player takes the ball from a member of the opposing team, bucket brigade learning is performed by obtaining the same reward value (15) regardless of the difference in roles between the players involved. On the other hand, 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.

Figure 5 shows the basic concept for determining the success rewards for each player. For example, a forward should take as many shots at goal as possible in order to gain points. Forwards are therefore given a large success reward for shots at goal, while their reward for stealing the ball from the opposing team is made relatively small. Conversely, the main duty of defense players is to prevent the opposing team from being able to take shots at goal. Defense players are thus given greater rewards for stealing the ball from the other team, and relatively small rewards for successful shots at goal. Meanwhile, the role of midfielders is to move the ball forwards to connect between the defense and forward players, and to act as surrogate defense or forward players when necessary. Accordingly, their success rewards are more evenly spread, with extra emphasis on actions such as passing and dribbling. Here, considering that all positions are of equal importance, the total of the success rewards for each type of play is set to the same value for all the forward, midfielder and defense players.

4. EVALUATION TRIALS

4.1 Evaluation method

For the evaluation data, we used three different algorithmic strategies that were employed in earlier trials [20]. One was a product prototype with a pre-prepared algorithmic strategy *A*. The other two strategies were based on algorithmic strategy *A*; one was modified to place more weight on attacking play (algorithmic strategy *B*), and the other was modified to place more weight on defensive play (algorithmic strategy *C*). Preliminary matches were played between these algorithmic strategies, and based on the results, strategies *A*, *B* and *C* were set up so that their winning rates were more or less equal.

The tests involved playing matches between two teams in a soccer environment and observing the number of games won and lost. An algorithmic decision-making system was used for the players of one team, while an event-driven classifier system was used for the players of the other team. The event-driven classifier system was evaluated by using a number of teams in which each player was set with different success reward values for plays such as passing and dribbling, according to the aims of the test. The duration of each game was set to 2 minutes, and the classifier system performed real-time learning during each game. Each match consisted of 200 successive games, and the effectiveness was evaluated from the winning rate in this match. Here, the winning R_w is defined by the following formula.

$$R_w = N_w / (N_t - N_d) \qquad (1)$$

Where N_t, N_d, and N_w are total number of matches, number of draws, and number of wins respectively.

Table 2. The success rewards for each play that were used in a team *H*2.

	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 3. The success rewards for each play that were used in a team *H*3.

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

Table 4. The success rewards for each play that were used in a team *H*4.

	FW	MF	DF
GETGOAL	60	60	60
DRIBBLE	2	2	2
PASS	8	8	8
GETBALL	10	10	10
LOSTBALL	-35	-35	-35
TOTAL	45	45	45

Table 5. The success rewards for each play that were used in a team *H*5.

	FW	MF	DF
GETGOAL	80	60	40
DRIBBLE	4	8	4
PASS	16	22	16
GETBALL	5	15	45
LOSTBALL	-60	-60	-60
TOTAL	45	45	45

In practice, we investigated whether or not changes in the game winning rate are caused by giving each player different success rewards based on the role assignments of different positions. We also investigated whether or not there were any changes in the game winning rate by changing the balance of success rewards of each type of play.

Three event-driven classifier systems incorporating algorithm *A* were prepared with differences in the rewards used for bucket-brigade learning. Specifically, these were a team *H*2 in which the success rewards of each player were set considering the role assignments shown in Table 2 in line with the basic policy mentioned above in section 3.3, a team *H*3 in which the success rewards of each player were set as shown in Table 3 based on the opposite idea to the above mentioned basic policy, and a team *H*1 which was set with the rewards used in the prior tests shown in Table 1. These three teams were each made to battle against algorithmic strategies *A*, *B* and *C*, and we comparatively evaluated them by determining the final asymptotic winning rates and the speed with which they converged on these final rates.

Next, we will describe the test method used to evaluate the relationship between the winning rate and the balance of success rewards of each type of play. The event-driven classifier system incorporating algorithm A provides a total of four teams — two different teams in which the success rewards of each player are set considering their role assignments, and two different teams in which no particular consideration is given to role assignments. Specifically, we provided two new teams — *H*4, in which the balance of success rewards for each play is modified as shown in Table 4 based on the rewards shown in Table 1, and *H*5, in which the balance of success rewards for each play is modified as shown in Table 5 based on the rewards shown in Table 2. Each of these four teams was matched against algorithmic strategies *A*, *B* and *C*, and we compared them with each other in terms of the eventual asymptotic winning rate and the speed of convergence on this rate.

We also investigated the relationships between the success rewards and the success rates of each play and between the winning rate and the success rate of each play, with the aim of using this information to analyze the strategies acquired through learning by the event-driven classifier system.

4.2 Experimental results

4.2.1 Position role assignments and winning rate

Figures 6 show the results of evaluating the relationships between the position role assignments and winning rates achieved by team *H*1, *H*2, and *H*3. In these figures, each point represents the average result obtained by playing 200 successive games 30 times. In the matches played with all three algorithms *A*, *B* and *C*, the highest winning rate was achieved with team *H*2 where the success rewards of each play were set as shown in Table 2 considering the role assignments. The lowest rising speed was achieved with team *H*3, where the success rewards of each play were set as shown in Table 3 using weightings opposite to those of the basic strategy.

4.2.2 The balance of success rewards of each play and the winning rate

Figures 6 also show the results of evaluating the relationship between the balance of success rewards of each play and the winning rate. In the matches played with all three algorithms *A*, *B* and *C*, team *H*1 ultimately converged on a higher winning rate than team *H*4, and the winning rate also rose at a faster rate. Team *H*2 ultimately converged on a higher winning rate than team *H*5, and its winning rate rose at a faster rate. Our results show that the winning rate and speed of convergence differ significantly when changes are made to the balance of success rewards for each play, regardless of whether or not the position role assignments are taken into consideration.

4.2.3 Success rate of each play

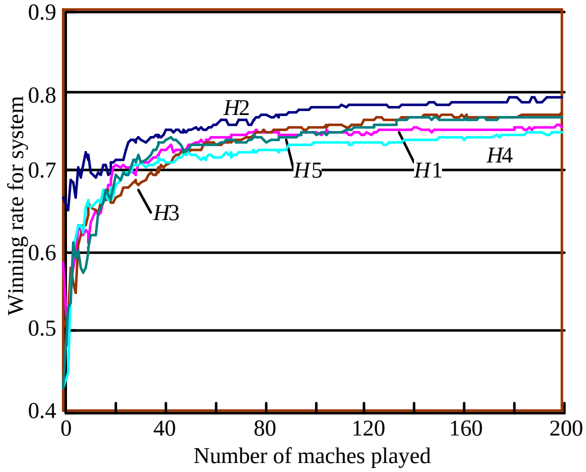
Figures 7–9 respectively show the ball possession rates, the number of pass per game, and the number of successful goals per game achieved by each team. The ball possession rates and the number of pass per game exhibit no particular correlation to the winning rate, but were highest for teams *H*1 and *H*2, while teams *H*3 produced lower values of magnitude. As for the number of successful goals per game, all the teams eventually converged on a rate of about 0.6, but our results show that this value rose much faster for team *H*2 which had a high winning rate.

5. DISCUSSION

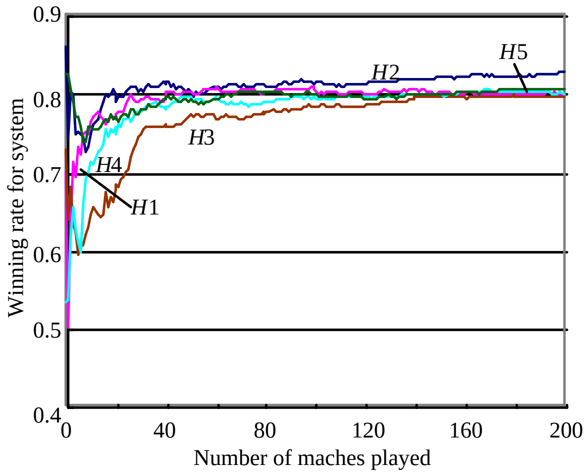
From Figures 6, the event-driven classifier systems *H*1–*H*5 incorporating algorithm *A* were able to perform learning to achieve a winning rate of more than 50% with all three of the algorithmic strategies *A*, *B* and *C*. Also, in all the matches with algorithms *A*, *B* and *C*, the winning rates were highest and converged the fastest with team *H*2, where the success rewards of each play were set considering the roll assignments. The event-driven classifier system thus seems to be able to contend with opponents having a wide variety of strategies, and it seems that conferring different success rewards to each type of play considering the role assignments of forward, midfielder and defense players results in a better winning rate and faster convergence. On the other hand, with regard to the tests for evaluating the relationship between the winning rate and the balance of success rewards for each play, Figs. 6 show that differences in the winning rate and the rate of convergence were caused by changing the balance of success rewards for each play independently of whether or not position role assignments were considered. Accordingly, by conferring different success rewards to each type of play by considering the role assignments, and by carefully setting the balance of success rewards for each type of play, it is thought that it is possible to gain further increases in the rate at which games are won and the rate of convergence. On the other hand, with regard to which specific value should be set, no explicitly determined procedure is set in particular. Although it can be determined by trial and error, it is also possible to consider determining the success reward values for each type of play by applying a procedure such as evolutionary computation. Further study will be needed relating to techniques for finding optimal values for the success rewards for each type of play.

Next, we will discuss the relationship between reward values and the success rates of each type of play. Figures 7 and 8 show that the ball possession rate and the number of pass per game had no particular bearing on the winning rate, with team *H*2 achieving higher values than team *H*1, and teams *H*3 producing low values. On the other hand, in Tables 1 through 3, the sum total of the values of the success rewards for dribbling awarded to forward, midfielder and defense players are found to be 12 for team *H*1, 8 for team *H*2, and 6 for teams *H*3, which corresponds to the order of the ball possession rates in Fig. 7. Also, with regard to the success reward values for passes, the sum total values were 48 for team *H*1, 32 for team *H*2, and 24 for teams *H*3, which corresponds to the ordering of the number of pass per game in Fig. 8. Specifically, it seems that the rate of success of individual play actions has a strong tendency to be dependent on the sum total of the success rewards for each type of play for forward, midfielder and defense players, independently of whether the game is won or lost. On the other hand, the number of successful goals per match was 180 for teams *H*1 and H2, and 200 for team *H*3, which does not correspond with the ordering in Fig. 9. In Fig. 9, the goal success rate of team *H*2, which has the highest winning rate, rises the fastest. In order to successfully score a goal, it is impossible to ignore the relationships with other forms of play, and it is thought that rather than being determined solely by the value of the success reward for an individual play, it is strongly related to whether the game is won or lost.

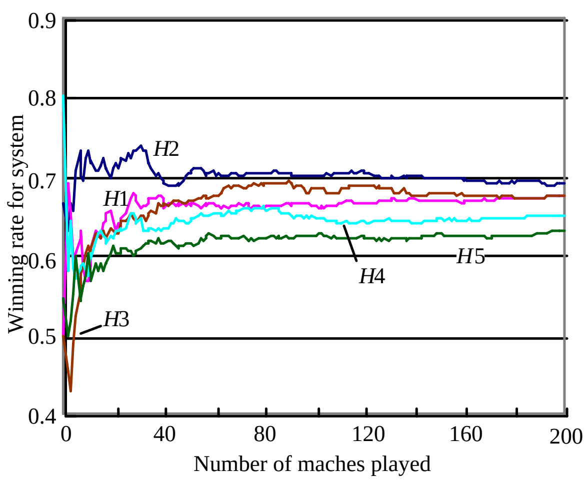
For a practical implementation, the number of learning cycles should be further reduced. Additional studies will be needed to make improvements so that learning can be achieved efficiently in a smaller number of matches. We have also discussed an extended proposal that involves applying the classical classifier system proposed by Holland to real-world problems, and we have presented the results of evaluating this proposal. In recent years, the use of eXtended Classifier System (XCS) [22] has begun to spread [1, 3-5, 11]. As with traditional learning classifier system, XCS is a problem-independent and adaptive machine learning model and gives several advantages [10, 11, 23]. In the future it will probably be necessary to conduct comparative tests with systems implemented using XCS. It will also be necessary to conduct further studies on cooperative learning between players or on the analysis of strategies acquired by learning.



(a) Algorithm *A* v.s. *H*1 - *H*5



(b) Algorithm *B* v.s. *H*1 - *H*5



(c) Algorithm *C* v.s. *H*1 - *H*5

Figure 6. The relationships between the position role assignments and winning rates.

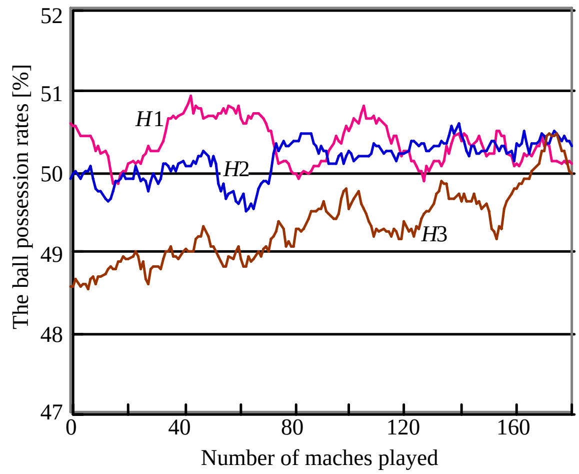


Figure 7. The ball possession rates achieved by team *H*1, *H*2, and *H*3. (v.s. Algoritm *A*)

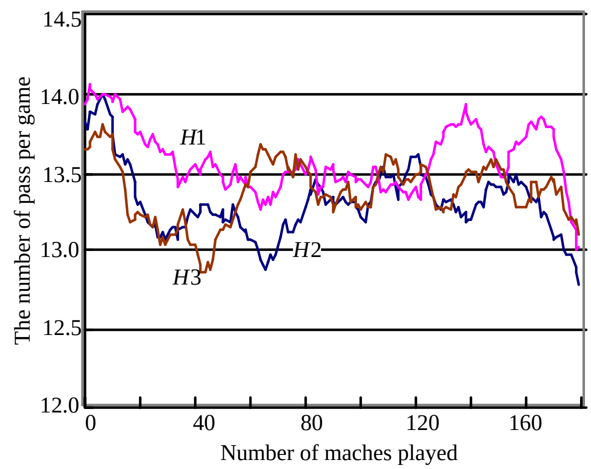


Figure 8. The number of pass per game achieved by team *H*1, *H*2, and *H*3. (v.s. Algoritm *A*)

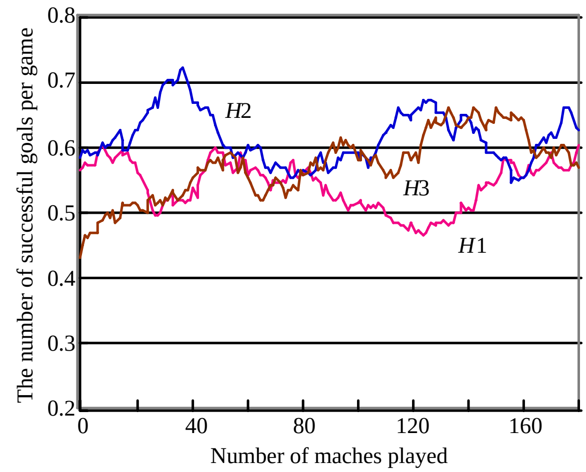


Figure 9. The number of successful goals per game achieved by team *H*1, *H*2, and *H*3. (v.s. Algoritm *A*)

6. CONCLUSIONS

In this paper we have investigated the concept of rewards in the application of a classifier system to the acquisition of decision-making algorithms for in-game agents in soccer video games. In particular, by considering differences in the role assignments of forward, midfielder and defense players, we have investigated the effects of performing learning by applying differences to the reward values conferred in reinforcement learning, and the effects of performing learning by varying the balance of success rewards between each type of play. Experiments were performed consisting of 200 matches against an existing soccer game with an algorithm devised by humans, and we evaluated the effectiveness from the eventual winning rate and the speed with which the winning rate converged on this eventual figure. As a result, by conferring different success rewards for each type of play based on role assignments, and by carefully setting the balance of success rewards between each type of play, we have been able to achieve better game winning rates and a faster convergence on these improved rates, and we have gained the prospect of achieving effective real-time learning with an event-driven hybrid classifier system.

7. REFERENCES

[1] Barry, A. Limits in Long Path Learning with XCS. In *Proceedings of the Fifth Annual Genetic and Evolutionary Computation Conference*. Vol. 2, LNCS 2724, Springer-Verlag, Berlin, Heidelberg, 2003, 1832-1843.

[2] 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, 275-281.

[3] Bull, L., Wyatt, D., and Parmee, I. Towards the Use of XCS in Interactive Evolutionary Design. In *Proceedings of the Fourth Annual Genetic and Evolutionary Computation Conference*. Morgan Kaufmann Publishers, San Francisco, CA, 2002, 951.

[4] Butz, M.V., Goldberg, D.E., and Lanzi, P.L. Gradient-Based Learning Updates Improve XCS Performance in Multistep Problems. In *Proceedings of the Sixth Annual Genetic and Evolutionary Computation Conference*. Vol. 2, LNCS 3103, Springer-Verlag, Berlin, Heidelberg, 2004, 751-762.

[5] Dawson, D. Improving Performance in Size-Constrained Extended Classifier Systems. In *Proceedings of the Fifth Annual Genetic and Evolutionary Computation Conference*. Vol. 2, LNCS 2724, Springer-Verlag, Berlin, Heidelberg, 2003, 1870-1881.

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

[7] 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, 291-301.

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

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

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

[11] 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, 774-784.

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

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

[14] 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, 204-222.

[15] 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, 1065-1072.

[16] 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, 184-195.

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

[18] 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, 256-263.

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

[20] 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, 2091-2098.

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

[22] Wilson, S.W. Classifier Fitness Based on Accuracy. *Evolutionary Computation*, Vol. 3, No. 2, 1995, 149-175.

[23] Wilson, S.W. Generalization in the XCS Classifier System. In *Proceedings of the Third Annual Genetic Programming Conference*. Morgan Kaufmann Publishers, San Francisco, CA, 1998, 665-674.