AUT MasterMinds Team Description Paper 2013

Mohammadhossein Malmir [1], Shahin Boluki [1], Mohammad Simchi

[1] AmirKabir University of Technology (Tehran Polytechnique), Hafez Ave., Tehran, Iran

{ mhmalmir, sh.boluki, m.simchimail }@Gmail.com

Abstract. AUT MasterMinds team is started as a 2D soccer simulation team with the aim of preparing a platform for applying machine learning techniques and advanced artificial intelligence tools. In this paper we introduce some of our efforts and researches and some of the algorithms that are newly implemented in our team. We introduce our pressing behavior and show the application of Fuzzy controller [6,7] in approaching the opponent's ball owner and application of "Maximum Weighted Bipartite Matching" in its decision making part [1-5]. We introduce our offensive without ball positioning and application of "Voronoi Diagram" and its changes since last year. We also show the application of Reinforcement Learning (RL) and Markov Decision Process (MDP) [8-13] in improving dribble skill.

Keywords. 2D soccer simulation, artificial intelligence, fuzzy controller, reinforcement learning

1 Introduction

AUT MasterMinds team is started as a 2D soccer simulation team with the aim of preparing a platform for applying machine learning techniques and advanced artificial intelligence tools. The AUT team was established in JUNE 2011. Our goal is to use 2D soccer simulation as a multi-agent environment to develop our team in such a way that advanced Artificial Intelligence (AI) tools and machine learning techniques have the main role in developing our team performance. Our team is based on Agent2D (release 3.1). We have participated in RoboCup2012 in Mexico City under the name of AUT_2D and we have achieved the 6th place.

In this paper we introduce some of our team's algorithms. In section 2 we describe our pressing skill and its different parts and application of fuzzy controller [6,7] for approaching the opponent's ball owner in order to steal the ball. We also describe its decision making part in which one agent makes the decision and transmits it to others.

In section 3 we introduce our new offensive positioning and application of "Voronoi Diagram" vertices in it.

We introduce the application of Reinforcement Learning (RL) and Markov Decision Process (MDP) [8-13] in dribble skill.

Section 4 describes the changes of our task evaluator and section 5 gives a summary of our discussion and future work.

2 Pressing:

We define pressing as a behavior of eliminating all the pass chances for opponent's ball owner and preventing opponent's team from creating spaces for planning a scoring chance and trying to retake the ball possession as soon as possible to form our scoring opportunities. With this definition the importance and necessity of pressing is obvious. As it seems pressing is a comprehensive defensive behavior. We firstly divide this behavior into two parts: decision making and action execution. Then the different roles in action execution part are introduced. At last we will discuss the application of Fuzzy-based controller in one of the roles for approaching the ball owner.

2.1 Decision making

In the decision making part, in order to reduce the probability of conflicts in our player's decisions, one player of our team which has the best view of the field and players in defensive situations and situations where the ball is in midfield makes the decision for all and transmits the decision to our players via say ability. The decision maker in defensive situations near our goal is our central offender and in farther from our goal situations is our goalkeeper. In fact the decision maker doesn't participate in action execution part. The players are informed of their role and in case of marker role the player which should be marked. (The roles that are named below are introduced in section 2.2)

In the decision making process, at first stage our blocker and ball stealer is chosen based on a scoring method. The decision maker scores all of our players respecting some features like distance from ball owner, cycles it takes to reach the ball, area scores regarding the ball position and our player strategy position and the like. The feature scores are multiplied by their relative coefficient and summed together for each player of us. The players which take the highest score for blocker role and stealer role are chosen and eliminated from our defenders for next stage. Note that the features for choosing the blocker are slightly different from the features for ball stealer.

In the second stage our markers are chosen. The algorithm used to do the matching of the opponent's attackers to our players for man to man marking is "Maximum Weighted Bipartite Matching" [1-5] with features like the marking decision previously implemented in our team [4].

In the third stage if we still have players with no roles assigned, our decision maker informs them to cover the dangerous spaces or to cover the through pass routes through our defensive line.

2.2 Action execution and roles

After being informed of their roles, our players execute their roles. The short definitions of roles are as below:

-*Blocker:* the task of the blocker is to stop the advancement of ball by the opponent's ball owner. The second priority of the blocker is to regain the possession of the ball [4].

-*Marker:* the duty of the marker is to prevent the assigned opponent's player from receiving pass, whether it's a leading or direct pass [4].

-*Ball Stealer:* the first priority of ball stealer is to retake the ball from opponent team. The second priority of this role is to approach the ball owner in a way to cover the most angle of ball owner.

-*Coverer:* the task of a coverer as it sounds is to cover the dangerous spaces or to cover the through pass routes in our defense line. The decision maker just assigns the role to the coverer and it is its own decision to cover which one. The routes and the dangerous areas are also recognized by coverer its own.

2.3 Application of Fuzzy controller in ball stealer role and approaching the ball owner

As we mentioned before the main task of ball stealer is to approach the ball owner in an optimized way to regain the ball possession as soon as possible. Movement of an agent consists of a direction and a velocity. We want to prevent our agent from just moving along a straight line toward the ball owner's current position. In fact the smoothness of changing direction of our agent's movement due to the change in ball owner's direction, position and velocity is important.

Our agent determines its velocity in each cycle by a function of features like our player's type and stamina, opponent ball owner's position, velocity, type and the distance between our agent and opponent's ball owner and the area of ball owner. In fact we divide the field into several areas and one of the features that the function considers is the area containing the ball owner position.

For determining the direction of its movement, we applied a Fuzzy-based controller [6,7]. The inputs that we consider are:

-Area position of ball owner: We divide the field into several areas and consider the area that contains the position of ball owner in terms of levels of being dangerous.

-Distance between our agent and ball owner.

-Velocity of ball owner.

-The difference between our agent and ball owner on the X axis.

-The condition of the area between the opponent ball owner and our goal in terms of the difference of the number of our teammates and opponents.

-The relative body angle difference between our agent and ball owner

For maintaining the desired velocity and direction together (as an agent cannot turn and dash in one cycle simultaneously) we define two thresholds. If the change of determined direction in comparison to last cycle's direction is less than a threshold our agent does not change its direction. We have another threshold for the maximum time of not changing the direction. The threshold value is different due to field's condition.

3 Offensive Positioning

Due to great improvements in teams' defensive skills like marking skill, offensive positioning is an essential task for creating scoring chances. We have two kinds of positioning, one for receiving through pass (that we call escaping) and one for receiving direct pass [4].

Escaping is executed for agents expecting to receive a through pass. Based on field conditions the agents that have the chance of receiving a through pass execute this action. The algorithm consists of diagnosing through pass routes in opponent's defender line; generating points on them, scoring them respecting some features like distance to opponent's goal, opponent's defenders mean distance and the like, determining the best position and going for it.

The other type of positioning is executed for escaping opponent's marking and forming clear routes for receiving direct passes. For this in we create the "Voronoi Diagram" of opponent players. The vertices of the diagram are safe and good positions for our attackers. In [4] we just calculated these vertices' positions as only options and matched them to our agents using "MWBM". But now we determine the best vertices for our agents with priority. In fact we match the most current strategic agent at first, then omit the vertex it was matched to and complete the same procedure for other agents. The suitable vertex for each agent is decided by scoring method. We do not

constrain our decision to Voronoi vertices in order to make our decision more reliable. Now we generate positions in a complete circle around our player, with fifteen degrees and one meter step. The upper bound of the radius of generated positions is dynamic due to our player role and field conditions. For reducing conflicts of approaching two very near positions or two very far positions that are not compatible with each other by two agents, one agent matches our players to suitable positions and transmits the decision to our players (currently our central midfielder). Our agent scores the positions respecting some features listed in Table 1 for each player and matches it to the position getting the most score with priority described earlier. The scoring is a linear summation of value of the features multiplied by their coefficients. As it is shown the effect of the Voronoi vertex that was determined for our player is as a feature of scoring the final position.

Table 1. Important factors of our positioning decision

	Features	Description	Value
F1	Distance from nearest opponent	The distance between the position and closest opponent to that position	The longer the distance is, the higher value is allocated to F1.
F2	Home position distance	The distance between the position and the formation position of our player	The less the distance is, the higher value is allocated to F2.
F3	Distance to goal	The distance between the position and opponent's goal	The less the amount is, the higher value is allocated to F3.
F4	Distance to ball	The distance between the position and ball position	In a specified range the longer the distance is, the higher value is allocated to F4; and in other cases the shorter the distance is, the higher value is allocated.
F5	Opponent-distance-mean	The opponents' agents mean distance to the position	The longer the distance is, The higher value is allocated to F5.
F6	Distance to teammate	The position's distance from our nearest teammate to that position	In a specified range the longer the distance is, the higher value is allocated to F6; and in other cases the shorter the distance is, the higher value is allocated.
F7	Distance to Voronoi diagram vertex	The position's distance to the matched Voronoi vertex for that player	The shorter the distance is, The higher value is allocated to F7.

4 Application of RL and MDP in Dribble Skill

Our assumed problem is that we give the target position to our agent to dribble to and our agent should find the best path to it without losing the ball. For this we employ RL and MDP process [8-13]. The total environment of our problem is constrained by boundaries that are determined by our current and target position and field condition and its size is dynamic. We define the states as small rectangles on field with their centers as their indicators. We assume opponents players as obstacles. The state that contains our target position is given the highest reward. We also define our actions as moving right, left, up and down (respecting X &Y axes). In this case it seems that we didn't consider diagonal direction. But the fact is that due to small surfaces of states (respecting our dribble speed and the distance ball travels between two consecutive kicks of dribbling) the algorithm is executed in time durations less than the time duration between two consecutive kicks of our dribble, in fact our agent chooses a sequence of states in the time duration between two consecutive kicks and not just moving from one state to the other. Then our agent will dribble on the direction of the resultant vector summation direction of vectors connecting its current position to the centers of chosen states.

In reality our problem is not stationary and we should employ Finite Horizon MDP for example but the amount of calculations is a lot and can be a problem itself. For this we solve our problem with quasi-static assumption. In fact we execute the algorithm in time durations that our environment remains static. For this our defined states are not of same size always and we change the sizes considering the field condition and the level of desired speed and accuracy. It is obvious that speed and accuracy are in contrast in dribbling and we need a trade off. For confirming our assumption in determining rewards of our states, we don't give minus rewards only to the states that contain opponents and we give minus rewards (with lower absolute values) to some neighboring states considering opponents' players velocity (direction and absolute size).

For example consider the first situation as we need an accurate dribble because there are opponents players between our agent and the target position and we don't need a fast dribble. In this situation we define states with very small surfaces, such that moving to several states takes a cycle of game. The States' sizes due to dribble accuracy and speed trade-off are shown in Fig1.

In other case for example the field condition is in a way that we want to dribble very fast and accuracy is not very important. In this case the powers of our agent's kicks are much greater than the previous case and the distance that ball travels between two consecutive kicks is not short. So we define states with bigger surfaces but note that the small ratio of state sizes in comparison to the distance that ball travels between two consecutive cycles is still preserved for verifying our assumption.

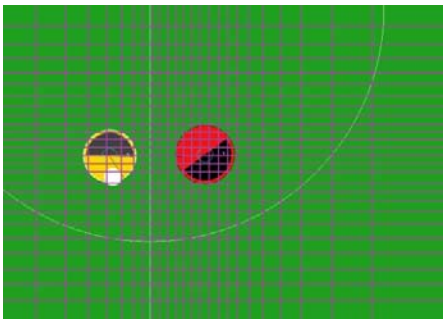

Fig 1. States' sizes due to dribble accuracy and speed trade-off

5 Task Evaluating

When our agent owns the ball has to decide whether to shoot, pass the ball, dribble with the ball or simply hold the ball [4]. In our earlier versions we firstly checked the shooting to opponent's goal situation. If we didn't have the goal scoring chance with a kick, then we virtually executed our pass and dribble and got the best output of each of them and decided between them. But there are situations that shooting with lower probability of scoring is better because you don't have any teammate or dribble chance or holding the ball and not shooting can create a passing situation for scoring. So now we divide our shooting situation into two types; hundred percent of scoring chance shoots and other shoots. When our agent has the first shooting situation type shoots but when we don't have hundred percent chance of scoring we calculate the chance of scoring in our shoot considering different factors. Then we make an array of the outputs of our pass and dribble that was virtually executed and then we evaluate and score them by our Task Evaluator class considering the role and position of the ball owner player, the condition of the field and whether the situation is now defensive or offensive. Now our agent selects between pass or dribble the one that got the higher score. It now compares the best dribble or best pass that was selected with our shooting chance considering its effectiveness due to our position, our teammates' positions, field condition and the like, and decides what action to do. If we didn't have any safe output of our pass and dribble and our shooting chance was lower than a threshold making decision is limited to clearing the ball and holding the ball. At this situation our player chooses one of the mentioned actions.

6 Summary

In this paper we introduced some of our novel algorithms. We introduced our pressing behavior (section 2) and its decision making part (section 2.1) and described the application of Fuzzy controller in approaching the opponent's ball owner to steal the ball (section 2.2). Our new offensive positioning and the application of Voronoi Diagram was shown in (section 3). As we mentioned before in this paper we are currently working on improving our dribble skill with Reinforcement Learning (RL) and Markov Decision Process(MDP) (section 4) and our Task Evaluator(section 5). And also we are still developing our Pressing skill, because it is a comprehensive skill.

References:

1. M. Paul. Algorithmen für das Maximum Weight Matching Problem in bipartiten Graphen. Master's thesis, Fachbereich Informatik, Universität des Saarlandes, Saarbrücken, 1989.

2. Erwin Kreyszig , *Advanced Engineering Mathematics*(10th ed.) , Wiley , ISBN 0470458364

3. West, Douglas Brent (1999), *Introduction to Graph Theory* (2nd ed.), Prentice Hall, ISBN 0-13-014400-2

4. M.Malmir, S.Boluki, M.Simchi, H.Hediehloo, AUT_2D Team Description Paper 2012,RoboCup 2012.

5. M.Norouzitallab, A.Javari, A.Noroozi, S.M.A. Salehizadeh, K.Meshgi, Nemesis Team Description Paper 2010,RoboCup 2010.

6. Lin-Xin Wang, *A Course in Fuzzy Systems and Control* (1st ed.), Prentice-Hall, ISBN: 9780135408827

7. Timothy J.Ross, *Fuzzy Logic with Engineering Applicationsl* (2nd ed.), Wiley, ISBN 0-470-86075-8

8. S.J.Russell, P.Norvig, *Artificial Intelligence: A Modern Approach* (3rd ed.), Prentice-Hall, ISBN 0-13-604259-7

9. Q. Zhang, M. Li, X. Wang, Y. Zhang, "Reinforcement Learning in Robot Path Optimization". Journal of Software, 2012, 03.

10. R.S.Sutton, A.G.Barto, *Reinforcement Learning: An Introduction.* Cambridge, MA: MIT Press, 1998.

11. L.Busoniu, R.Babuska, B.De Schutter, D.Ernst, *Reinforcement Learning and Dynamic Programming Using Function Approximators.* FL:CRC Press, 2009.

12. C.Watkins, "Learning from delayed rewards". Ph.D. dissertation, Cambridge University, Cambridge, U.K., 1989.

13. C.J.C.H.Watkins, P.Dayan, "Q-learning". *Mach.Learn.,* vol.8, no.3-4, pp.279-292, 1992.