AUT Team Description Paper 2012

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

Shahin Boluki [1] , Hessamoddin Hediehloo [1]

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

{ mhmalmir, m.simchi, sh.boluki , hessam.hedieh}@aut.ac.ir

Abstract. AUT 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 implemented in our team. We show that applying “Maximum Weighted Bipartite Matching” (MWBM) [1-3] is useful for reducing defending conflicts [4] if executed by one player and the decision is transmitted to other players by say ability. We introduce two kinds of our offensive without ball positioning. We also show the application of “MWBM” and “Voronoi Diagram” for improving our offensive positioning.

1 Introduction

AUT team is a newly started 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 have the main role in developing our team performance. Our members had already worked on separate teams based on UVA-Trilean base in 2010 but started as a team based on agent 2D base (release 3.1) in June 2011.And since then we haven’t participated in any major competition but we have just qualified for Iran Open 2012 competition.

In this paper we introduce some of our team’s algorithms. We divide them mainly in two parts, first without ball actions and then with ball actions are described. In without ball actions we have the division of defensive skills and offensive skills. And at the end we describe our decision making algorithm.

In section 2 we show that by applying “MWBM” [1-3] and executing it by one player and transmitting the decision to other players by say ability we can remove the conflicts of the decision of which defender of us should now mark which offender of the opponent.

The decision making of who should now block the opponent ball owner is described in section 3.

In section 4 we show the application of “MWBM” and “Voronoi Diagram” for our offensive positioning.

We introduce our dribble algorithm in section 5 and our pass skill and its safety checking in section 6. And we end our paper with our decision making algorithm in section 7 and a short summary in section 8.

Without ball actions:

Defensive skills:

2 Marking

Marking opponent's attackers is one of the most important defending skills and plays an important role in preventing the attackers from scoring goals.

The most challenging part in this skill is determining which defender of us should mark which attacker in order to get the best result without any conflict and have a one by one marking.

For finding the solution for assignment of opponent's attackers to our defenders, we were inspired by *Nemesis 2010* team. We found "Maximum Weighted Bipartite Matching" helpful for our assignment problem and we employed it with Hungarian algorithm for solving it in our team.

We derived the basic idea of the way of applying the "MWBM" in our defense from *Nemesis* but we reached to this conclusion that if each player of us employs the algorithm by itself we still have conflicts in our decision, but by executing the algorithm by one player and informing the others of the decision we can reduce the problems to nearly zero (as will be mentioned in section 2.2).

2.1 Maximum Weighted Bipartite Matching (MWBM) [1-3]

In a maximum bipartite matching [2,3] if we allocate a weight to each edge we have a maximum weighted bipartite matching.

If we call the sum of the weights of our edges in a matching (M_i), the weight of our match $W_{(Mi)}$=$\sum_{es \in M} w(es)$,a maximum weighted matching M is a match in such a way that every other matching has lower weight than the weight of M.

2.2 Applying "MWBM" in our marking problem [4]

At first we make our graph consisting of our defenders at one side and the attackers on the other side as the nodes. Be mentioned that we omit some of opponent's attackers by some initial checking. We don't use our blocker in our nodes either (our block skill is described in the next part).

Then we give a weight $W_{(i,j)}$ to each edge respecting some features. The weight in fact indicates the importance of marking the attacker j by our defender i.

In fact if F_k is the value of feature k and R_k is the related coefficient and N is the number of the features the weight of each edge is the sum of the features value multiplied by their related coefficients as shown in formula (1).

$$W_{(i,j)} = \sum_{k=1}^{N} R_k \times F_k(i,j) \qquad (1)$$

Some of the important elements in our calculation are shown in the table 1.

Table 1. Important factors of our marking decision

	Features	Description	Value
F1	Distance difference	The distance between defender and the attacker	The less the distance is, the higher value is allocated to F1.
F2	Home position distance	The distance between the formation position of the defender and the attacker position	The less the distance is, the higher value is allocated to F2.
F3	Formation changing	The amount of the mess of our formation with that marking	The less the amount is, the higher value is allocated to F3.
F4	Scoring chance	The chance of scoring goal of the attacker	The higher the chance is, the higher value is allocated to F4.
F5	Area score	The area which the attacker position is in, gets a score regarding to the area of the defender	The higher the score is, The higher value is allocated to F5.

The coefficients are now set by hand, but we look forward to applying training procedures and high level algorithms for optimizing them.

In the first versions of our code each defender of us applied this algorithm by itself and reached to a conclusion to mark which attacker, but we saw some conflicts in this way, i.e., two or more defenders decided to mark one attacker and the other attacker(s) remained free.

So we concluded that one player of us should do the matching and inform the others of that. Now our goalie plays this role because it has a good view of the field in our defending conditions.

We use the say ability for transmitting the matching result to our defenders, so they can mark the proper attacker.

3 Blocking:

Blocking is a skill for stopping the opponent's attacker dribble in order to cut their attack and obtain the possession of the ball. It is divided in two parts:

1) The decision of who should now block the dribbler.

2) The way of implementing the action.

The decision of who should now block the advancing of the ball by the ball owner attacker is done by the goalie. The goalie scores each of our defenders for being the blocker respecting some features like the distance of the defender to the ball owner, being the fastest teammate to reach the ball, area scoring regarding to the area position of the ball and our strategy position and the like .Each feature has a respective coefficient that the feature's score is multiplied by it. The final score of a defender is the sum of all of its features scores multiplied by their respective coefficient.

The player who gets the highest score is the best blocker. The goalie transmits the decision by say message to our defenders, like our marking decision.

For the way of implementing the block action we are working on an algorithm for predicting the opponent's attacker behavior in order to stop him in the best way.

Offensive skills:

4 Offensive Positioning

Due to increasing development of marking abilities and defense power of 2D teams, scoring goal is a hard and challenging task. Attackers should escape the marking of the opponent's defenders in order to create space for themselves to have the chance to score. Our attackers have mainly two kinds of positioning.

4.1 *Escaping from opponent's defenders when we have cross situation in order to receive pass and score.* For this we apply the "Voronoi Diagram" idea and "Maximum Weighted Bipartite Matching". First we create the Voronoi diagram of opponent's defenders positions and we find the vertices of the diagram. The vertices of the Voronoi diagram are good positions for our attackers because they have safe distance from opponent's defenders. For assigning our attackers to vertices positions without any conflict we apply "MWBM" to our problem employing Hungarian algorithm for solving it (like our mark skill section 2). We assume our attackers on one side and the vertices positions on the other side as the graph's nodes. One of our players who have the best view in attacking situations executes the algorithm and informs our attackers by using say ability. We use the same equation (1) in section 2.2 with different features and coefficients. Some of the features are distance from goal, distance from current position of our attacker, distance from formation position of our attacker and distance from ball position. This part is currently under development and is not fully implemented in our team.

4.2 *Creating space and escaping from defender's line to receive through pass when we have the situation.* For this our attackers considering the defensive line of opponent's team choose some positions between opponent's defenders and score them considering parameters like distance from ball, distance from goal, distance from formation position and opponent's defenders mean distance. At last each attacker chooses the position which he has concluded has the biggest score and goes for it. For not having conflict between our attackers we use some partitioning of field based on our formation in choosing the initial positions to be scored for each attacker.

With ball actions:

5 Dribble

Our dribble algorithm chooses the position which the dribbler should go and the speed of the dribble and uses the "Body_Dribble" function in rcsc library with an extra argument for determining the dribble speed to survey to that position (Table 2).

Table2. Different kinds of dribble

Dribble	Body_Dribble(pos , pos_count , dash_count , dash_power , dodge_mode , speedy)
Fast Dribble	Body_Dribble(pos , pos_count , dash_count , dash_power **,** **false** **, true**)
With ball Dribble	Body_Dribble(pos , pos_count , dash_count , dash_power **,** **true** **, true**)
Slow Dribble	Body_Dribble(pos , pos_count , dash_count , dash_power , **true** **, false**)

We choose the target of the dribble by first generating points around our dribbler and scoring them by the equation (3). The target with the biggest score is chosen. The description of the features is shown in Table 3.

Score = cycle_diff * cycle_diff_weight +
dir_diff_to_near_opp * dir_diff_to_near_opp_weight +
dist_diff_to_near_opp * dist_diff_to_near_opp_weight. (2)

Table3. Features of scoring of dribble

Feature Name	Description
cycle_diff	Difference of reaching cycles of our dribbler and the opponent's minimum
dir_diff_to_near_opp	Angle with the vertex of our player consisting of the lines from our player to opponent and from our player to dribble target.
dist_diff_to_near_opp	Distant of nearest opponent to the dribble target position

6 Pass:

The passing skill which can be mentioned as the main act of play making is divided in two kinds, direct pass, and through pass. The former is being utilized for moving the ball all around the field either for starting an attack or play making; especially in the opponents danger area to have a better chance for scoring. The latter one is good to disorganize the opponent's defenders and to make a good attacking opportunity. This pass is only executed for some agents.

We initially generate possible targets for our pass (section 6.1). After points are generated, each point is given a score by factors like the distance of the point toward the opponent's goal, the safety of pass (section 6.2), the mean distance of the opponents to the position, distance of the nearest opponent, the difference between the passer and the target position co-ordinates, the ability of the receiver to pass. The features of scoring direct and through pass are the same but their coefficients differ. After the points are scored the point with maximum score is the best pass.

6.1 Point generation: The main algorithm that is utilized for point generation in pass is based on the fact that processes for safety of pass can be so much, causing the system to break down if the points are more than needed. A commonly used algorithm for generating points is to draw a circle around each agent with desired diameter, however, based on the fact that all players do not need all the points but other players may need, (for example, in this method some points that are generated for agents in two sides of the field, will lay outside the field and are useless.) we decided to generate the points with cone like shapes for some players, yet using the old method for others. This change in generation method gave us the power to increase safety to nearly desired level without losing the desired points. In the new method we have divided the field into three zones based on the zone that the pass receiver positions fit in we choose the method. In the defense and offense zone the first method and in the midfield zone the new method is used. The zones are shown in fig.1.

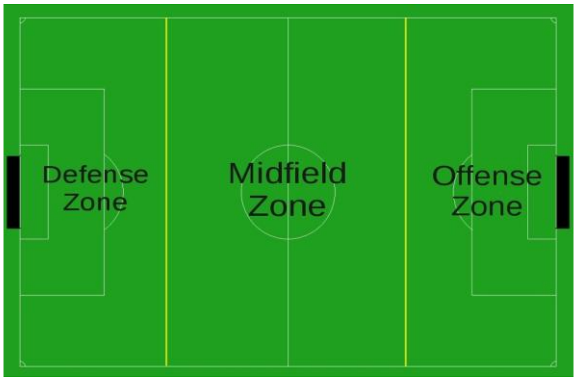

Fig.1 Different zones of the field

6.2 Safety: The safety has two parts, a general part which is taken into account for both kinds of pass, and the one that is specialized for direct pass. The general part of safety is the one that predicts the cycles that receiver needs to reach the ball, and also checks that whether any of opponent players (and surprisingly, teammate players!) can reach the ball faster than the desired receiver or not. This allows us to check whether the pass is safe or not, in addition to calculate the cycles it takes to finish the task. It also checks the time that the spatial information (including position, velocity, body angle and neck angle) of objects were updated to assure about the accuracy of the calculation, which is of critical importance, especially in leading passes that the ball needs to pass opponents by a narrow margin. The specialized part of safety for direct passes determines the safety considering factors like distance between the target and the passer, distance of the opponents and the target point, as well as other factors.

Decision Making:

7 Task Evaluating:

When our player owns the ball has to decide whether to shoot, pass the ball, dribble with the ball or simply hold the ball. For this we firstly check if we have the shooting to opponent's goal situation. If we haven't the goal scoring chance with a kick, then we virtually execute our pass and dribble and get the best output of each of them. Then we make an array of the outputs of our pass and dribble 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. If we didn't have any safe output of our pass and dribble making decision is limited to clearing the ball and holding the ball. At this situation our player chooses one of the mentioned actions.

We are currently working on evaluating each task with a kind of prediction for other player's next actions and decisions to have a good evaluation with future trend.

8 Summary:

In this paper we introduced some of our novel algorithms. We described the application of "MWBM" in our defense (section 2) and in our offensive positioning with the help of "Voronoi Diagram" (section 4). We also introduced our pass and its safety checking (section 6).As we mentioned before in this paper we are currently working on our Block Skill (section 3) to have a reliable prediction of the opponent's attacker dribble. And also we are developing our Task Evaluator (section 7) and Offensive Positioning (section 4.1) as mentioned in their sections. The other task that we plan to work on is optimizing the coefficients that we have used for scoring in different parts like our Mark Decision (section 2.2) and Dribble (section 5) using advanced AI methods (e.g. Particle Swarm Optimization [6]).

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

[5] S.Saharkhiz, P.Kaviani, M.Bakhtiari, M.R.Montazeri, Eskilas Team Description Paper 2011, RoboCup 2011.

[6] R.C. Eberhart, J. Kennedy, "*A new optimizer using particle swarm theory,*" Proceedings of the Sixth International Symposium on Micro Machine and Human Science (MHS'95), Nagoya, Japan, 1995, pp. 39–43.