Razi Soccer 2D Simulation Team Description Paper 2018

Morteza Noohpisheh[1] , Maryam Shekarriz[2]
Sepehr Barzegar[3], Danial Borzoo[4] , Ali kariminia[5]

[1] Dept. of Computer Engineering, Shiraz University
Persia.saftware@gmail.com,
mshz@yahoo.com

[3]Tohid High-school

[4]Ehsan High-school

[5]Tafazzoli2 High-school

Abstract. This paper includes some explanations about algorithms implemented by Razi team members. We will introduce algorithms that are used for pass, shoot, dribbling and marking, and in particular with the ranking algorithm for the states of chain action. The base code used by Razi is agent-2d 3.1.1.

1 Introduction

The goal is to implement intelligent agents, that have the ability to decide on using soccer individual skills as well as the ability to work with other agents in the soccer field. In addition, programming has been done by agents to improve individual skills (shoot, pass, dribble, positions of players on the field).

Razi team is formed from students, has started its activities from November 2012 and has made plan in order to achieve its goal, participating in World Cup competitions. This team has participated in prestigious competitions and has won the third place in Iran Open 2017. This team has begun to use artificial intelligence in its decision making. It continues its work with the aim of using artificial intelligence algorithms and online analysis.

2 Skills of each player

2.1 Dribble

The Ball owner player forms 3 circle with specific radius to choose the best path to move. It forms 10-cones in the first front circle then forms other cones along those 10 cones axis, with the vertices on the second circle. Every both successive cones form a rhomboid (Yellow rhomboids). This action will be also done on the third circle to form gray rhomboids. The number of opponent's players will be earned from the outer circle, they will influence on their adjacent triangles depending on their scores. Each triangle also influences on its adjacent cones. This method continues until the last cone and the high score cone will be selected to dribble toward the middle of it.

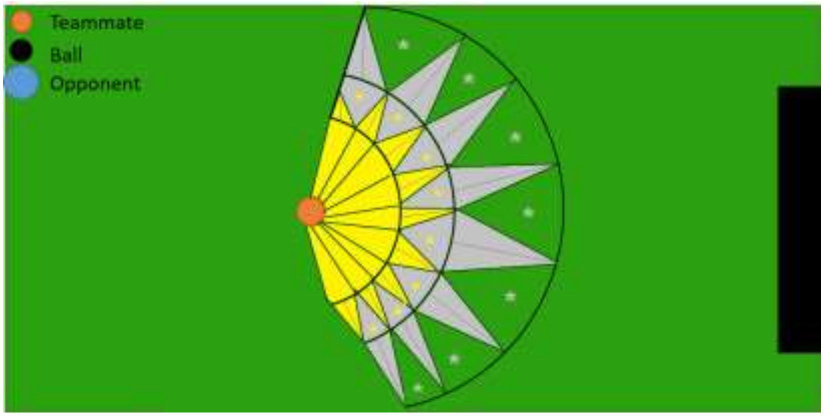


Figure 1: Find Target Point To Dribble

2.2 Pass

First the ball owner player draws a hypothetical circle with specific radius around itself then distinguishes the teammates in the circle. As shown in Figure, it draws a specific cone for each one.

It checks the following parameters for each of the players in the circle and gives each player a single point. It chooses the player who takes the highest score and passes it.

Parameters of the pass algorithm, including:

D = Teammates Distance From Agent

X = Opponent Distance From Target Teammate

OD = Nearest Opponent Distance From Pass Line

C = Opponents In Pass Cone

Teammates Pass Score = (20-d*2.5)+od*7+x*5.625+(3-c)*3.25

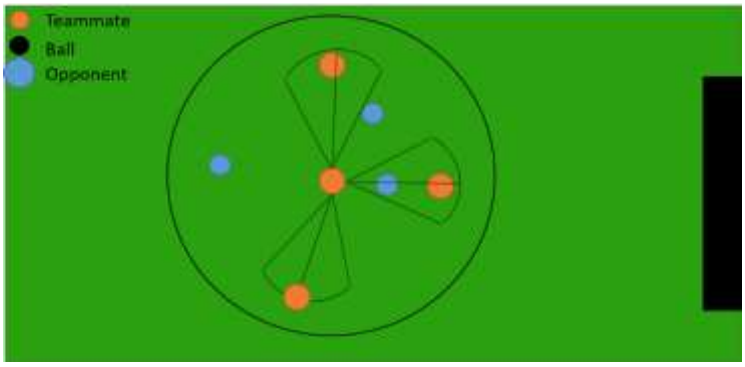


Figure 2: Find best Teammate For Pass

2.3 Shoot

Divide and conquer method is used for shooting to the goal. In this method, at first, three points will be considered on the opponent's goal and the points between these three points will be chosen for the shoot. The scores of each shooting points will be calculated and depending on the scores, a region will be omitted and the first, center and last points will change. These steps will be repeated five time. In this way, finding the best point to shoot will be faster and also the player will have 32 points instead of 25-points to choose.

In the following figure, you can see how the algorithm is implemented.

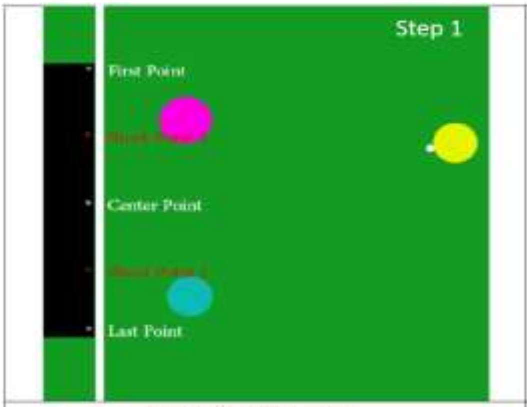


Figure 3: Select shoot point

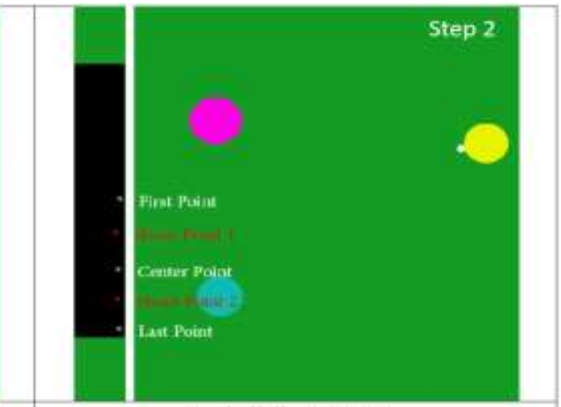


Figure 4: Select shoot point

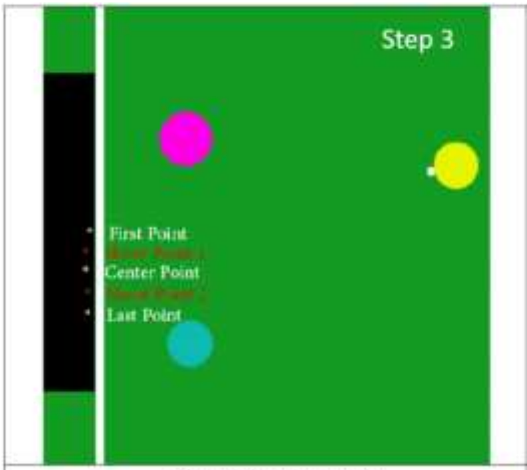


Figure 5: Select shoot point

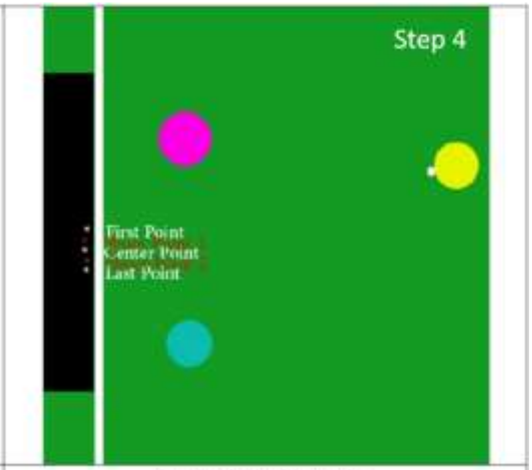


Figure 6: Select shoot point

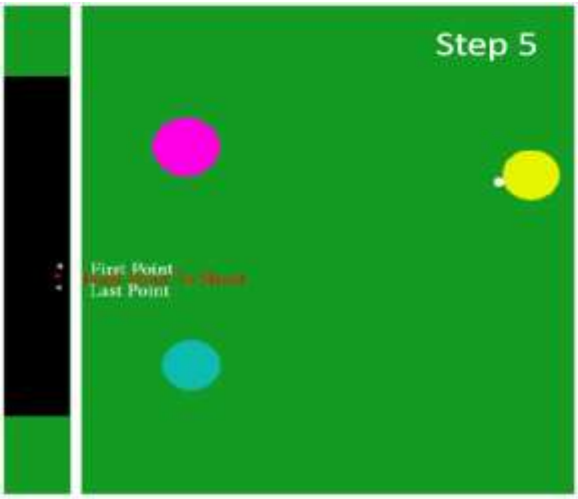


Figure 7: Choosing Final Point

2.4 Mark

To mark the opponent's players, Initially, a hypothetical circle will be drawn with the center of the ball and specific radius and all the opponents in the front semicircle will be considered and the closest player to each opponent will go to Mark it.

If one of the following three applies to your own player, it will not do the mark:

1- The player don't be a central defender.
2- Stamina do not be less than an amount.
3- Do not be in the kick circle.

For marking at the free kick time, 2 of the closest players to the opponent's players in the circle will be chosen then they will get some points, depending on the rating, it should mark.

For example, we will specify two of the following parameters:

D = Distance to Mark Point (Opponent)

GD = Distance to goal (negative score)

We draw a hypothetical line from kick point to players outside the circle which can be effective in the attacks and if there is a player on this line, a negative score will apply depending on the opponent's distance to goal.

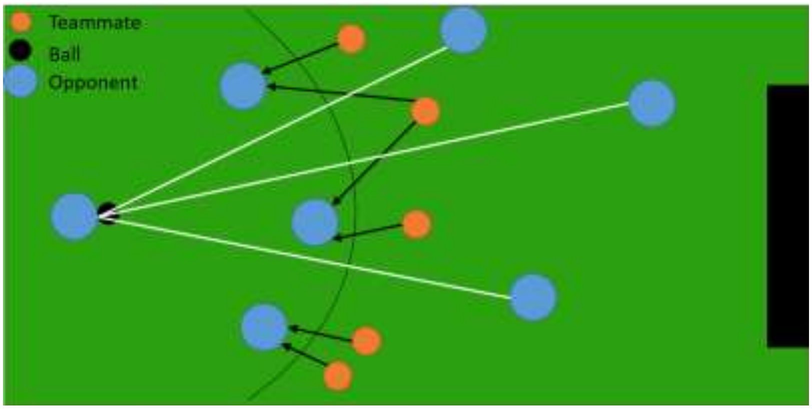


Figure 8: Marking Opponent Players

3 Attack strategy

3.1 How to rate states in chain action

Giving points will be done in every state in decision tree which involve some parameters. One of the parameters is the distance from the goal, the amount of the effectiveness of this distance will also be different due to the position of the ball. For this purpose, we consider a 10-array, which include the effective coefficients. We compute the following index in order to determine which coefficient must be given.

$$\text{Index} = \lfloor \text{BallPosition} / 10 \rfloor$$

This means that as far as the ball is farther from the goal, our distance from the goal has less effective coefficient.

To find the best coefficients, we use the genetic algorithm. In this way, we consider 100 arrays that contain random and ordered coefficients and 10 games have been played with each of the populations. We considered the number of goals scored in each game after each match, as an evaluation of that game.

we combined the previous populations and formed new populations of 100 new ones then we repeated 10-games. We did this several times, until we finally got to a population that got the high and acceptable number of goals scored and its coefficients were chosen as the main coefficients.

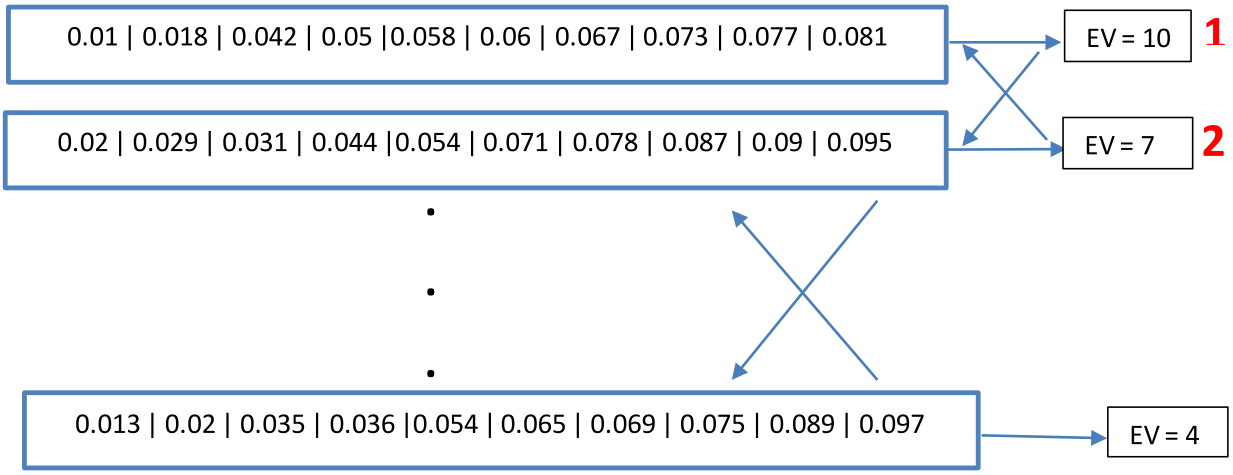
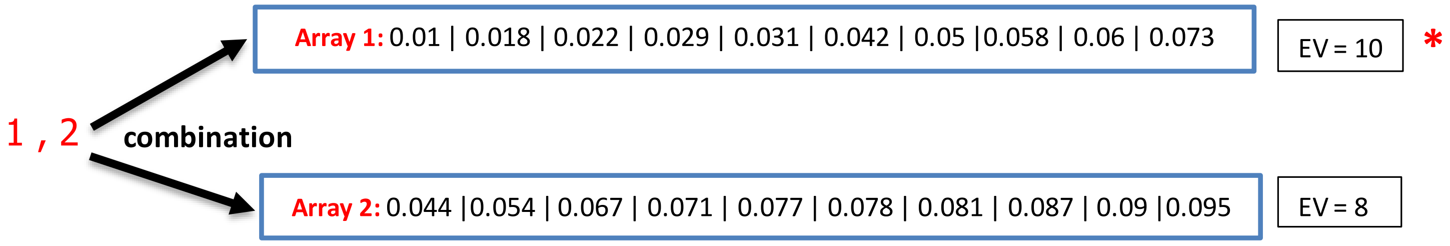


Figure 9: Example of arrays and how they will combined.

For example when we combine 1 and 2, we get two arrays that are shown below :



Array 1: 0.01 | 0.018 | 0.022 | 0.029 | 0.031 | 0.042 | 0.05 |0.058 | 0.06 | 0.073
EV = 10
*
1 , 2
combination
Array 2: 0.044 |0.054 | 0.067 | 0.071 | 0.077 | 0.078 | 0.081 | 0.087 | 0.09 |0.095
EV = 8

Figure 10: Array 1 and Array 2 after combination. The evaluation of array 1 is higher, so we choose array 1's populations.

In the following tables we are comparing results of using our algorithm with default algorithm.

Match number	Cyrus2017 vs. Razi2017	Cyrus2017 vs. Razi2018
1	3-3	1-0
2	3-2	1-1
3	1-2	2-1

Table 1. Matching Results between Cyrus2017 and Razi

Match number	Helios2017 vs. Razi2017	Helios2017 vs. Razi2018
1	6-0	3-0
2	5-1	3-1
3	4-0	2-1

Table 2. Matching Results between Helios2017 and Razi

References

1. Frieder Stolzenburg, OliverObst, Jan Murry, *Qualitative Velocity and ball Interception*, 2002
2. Gabel, T.,Riedmiller, M.Trost, F.:A Case Study on Improving Defense Behavior in Soccer Simulation 2D:The NeuroHassle Approach. In:Springer-Verlag Berlin Heidelberg (2009) 61-72
3. An Introduction to Genetic Algorithms , Jenna Carr , May 16, 2014
4. " RoboCup Soccer Simulation", http://en.sourceforge.jp/projects/rctools , 2017
5. Nader Zare, Ashkan Keshavarzi, Seyed Ehsan Beheshtian, Hadi Mowla, Aryan Akbarpour, Hossein Jafari, Keyvan Arab Baraghi, Mohammad Amin Zarifi, Hossein Jafari, “CYRUS 2D Simulation Team Description Paper 2016”, http://www.robocup2016.org/media/symposium/Team-Description-Papers/Simulation2D/RoboCup_Sim2D_TDP_CYRUS.pdf
6. https://www.robocup2017.org/file/symposium/soccer_sim_2D/TDP_CYRUS.pdf
7. https://www.robocup2017.org/file/symposium/soccer_sim_2D/TDP_HELIOS2017.pdf