

Razi Soccer 2D Simulation

Team Description Paper 2019

Morteza Noohpisheh¹ , Maryam Shekarriz¹
Afshin Bordbar¹ , Mohammadhassan Liaghat² , Alireza Salimi² ,
Daniel Borzoo² , Alireza Zarei¹

¹ Dept. of Computer Engineering, Shiraz University
morteza.noohpisheh@gmail.com , mshz@yahoo.com , zareialireza970@yahoo.com

²Tohid 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 IranOpen2017 and IranOpen2018. 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 The Gravity Strategy

We are implementing the Gravity Strategy Algorithm for Razi Team at the 2019 RoboCup.

The gravity player was created by members of Taylor University's RoboCup team, and was made as a way for a player to dynamically adjust movement and kicking destinations without having to manually account for certain teammate/opponent field position configurations.

One cause for this strategy was as a way to solve the problem of having players bunch up on the field. By taking into account the distance to teammates, players can be more inclined to spread out across the field rather than bunching up.

The original creators of the gravity strategy also saw benefits regarding kicking as well. Since the strategy is factoring in teammates distance and opponents distances from a point to kick to, gravity players can make fairly accurate passes, without many occurrences of an opponent intercepting.

For every cycle of the match, the 52.5x34 unit RoboCup field is divided into a 315x204 array, whose elements are points on the field. Depending on the action the player is going to take, either moving or kicking the ball, we then calculate a weight for each point in our array, whose value is dependant on the specific point's relation to certain factors in the equation. After all the

points have been assigned a weight, we find the maximum weighted point, and either move towards that point or kick the ball at that point.

The gravity strategy utilizes two separate, but similar, equations to calculate the weights that are assigned to each of the points. Equation 1 is specific to player movement, and Equation 2 for kicking the ball. If the player is able to kick the ball they will evaluate the kick equation, and if they are not able then they will evaluate the move equation. These two equations are evaluated independently, thus the weights for one equation do not affect the weights for another equation.

If the player is going to perform a move action, for each point in our field array we first check to see if an opponent can reach the location of the ball before we could at that point. If they can, then we set the weight of that point to negative infinity, which indicates a lowest possible weight for that cycle. If the opponent cannot, we then calculate the weight of that point based off the move equation. The equation is the following :

Equations 1:

$$Z_{\text{move}} = \min((\alpha_1 \Delta_E), (\alpha_2 \Delta_B)) - \frac{\Delta_P}{\alpha_3} - \frac{\Delta_G}{\alpha_4} + (\alpha_5 \Delta_O) + (\alpha_6 \Delta_T)$$

If the player is going to perform a kick action, for each point in our field array not on the border of the field, we check to see if the point is in a place where a teammate could not get to the ball before an opponent could. If the point meets this criteria, then we set that point's weight to the default equation, otherwise we set the weight to the main kick equation's result. The full equation is the following :

Equations 2:

$$Z_{\text{Kick}} = \begin{cases} f() & \text{Teammate can not reach the ball first} \\ g() & \text{Teammate can reach the ball first} \end{cases}$$

$$f() = -\left(\frac{\Delta_T}{\alpha_7}\right) - \left(\frac{P_y}{\alpha_8}\right) - (\alpha_9 \Delta_G) + (\alpha_{10} \Delta_E) + (\alpha_{11} \Delta_O)$$

$$g() = (-\Delta_G)$$

Equations 1 and 2 were designed by Taylor RoboCup team, and take into account factors they believe to be important in regards to the action that they pertain to. The alpha coefficient values are the focus of this experiment, and determine the importance, or weight, that a particular factor in the equation has on the overall action.

Regarding the training games performed by Razi team, with the implementation of these formulas, we improved the above formulas and replaced the two formulas below.

$$Z_{\text{move}} = \min((\alpha_1 \Delta_E), (\alpha_2 \Delta_B)) - \frac{\Delta_P}{(\alpha_3 \Delta_B)} - \frac{\Delta_G}{(\alpha_4 \Delta_B)} + (\alpha_5 \Delta_O) + (\alpha_6 \Delta_T) - (\alpha_7 \Delta_{TG})$$

$$f() = -\left(\frac{\Delta_T}{\alpha_8}\right) - \left(\frac{P_y}{\alpha_9}\right) - (\alpha_{10} \Delta_G) + (\alpha_{11} \Delta_E) + (\alpha_{12} \Delta_O) + (\alpha_{13} \Delta_{TG})$$

This change made the team more likely to move to the opponent's goal and avoid the pass to the goal and the defense line

Value	Description
Z_{move}, Z_{Kick}	The gravity weight value of a specific point.
$\alpha_8 \dots \alpha_{13}$	Coefficient constants controlled by genetic algorithm.
Δ_E	Distance to the nearest edge of the soccer field.
Δ_B	Distance to the ball.
Δ_P	Distance to the player.
Δ_G	Distance to the opponent's goal
Δ_O	Distance to closest opponent.
Δ_T	Distance to closest teammate.
Δ_{TG}	Distance to the goal
P_y	The yaxis coordinate of the player, with range [34,34].

Table 1: Variables used in the kick and move equations for the gravity player. Z values are computed at every point on the field. Distances are computed from the point.

3 Skills of each player

3.1 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.

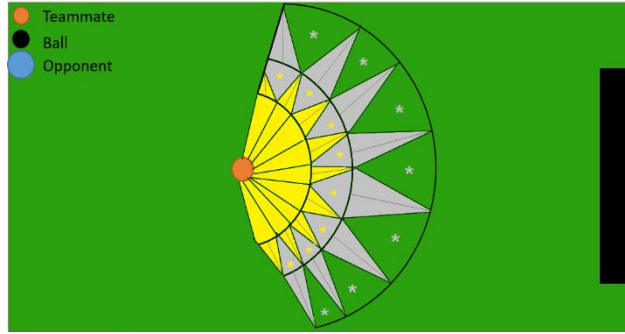


Figure 1: Find Target Point To Dribble

3.2 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.

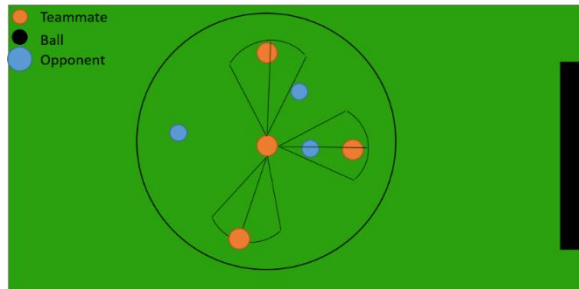


Figure 2: Find best Teammate For Pass

Parameters of the pass algorithm, including:

D = Teammates Distance From Agent

C = Opponents In Pass Cone

OD = Nearest Opponent Distance From Pass Line

X = Opponent Distance From Target Teammate

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

3.3 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.

3.4 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(we can find them from our collected data from the game) and if there is a player on this line, a negative score will apply depending on the opponent's distance to goal.

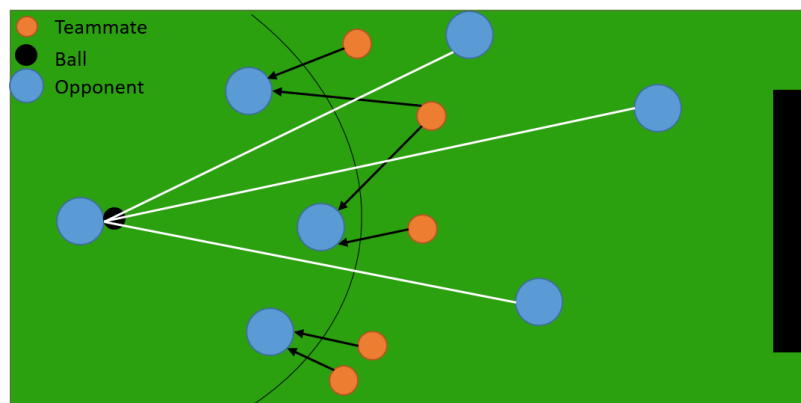


Figure 3: Marking Opponent Players

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.

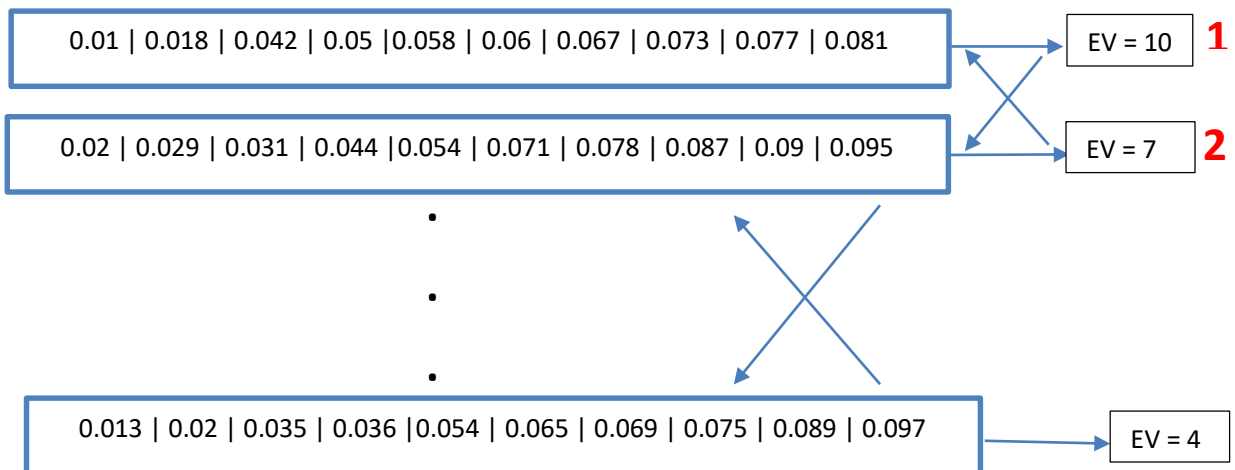


Figure 4: Example of arrays and how they will combined.

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

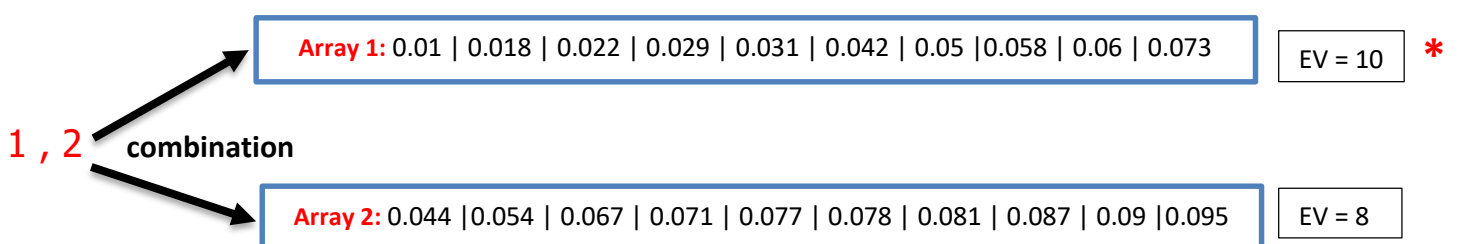


Figure 5: Array 1 and Array 2 after combination. The evaluation of array 1 is higher, so we choose array 1's populations.

We compared results of using our algorithm with default algorithm before in our 2018's Team Description.

References

1. *Frieder Stolzenburg, Oliver Obst, Jan Murry, Qualitative Velocity and ball Interception*, 2002
2. *Hausknecht, Matthew, and Peter Stone. "Deep reinforcement learning in parameterized action space."* arXiv preprint arXiv:1511.04143 (2015)
3. *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*
4. *An Introduction to Genetic Algorithms*, Jenna Carr, May 16, 2014
5. "RoboCup Soccer Simulation", <http://en.sourceforge.jp/projects/rctools>, 2017
6. *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
7. https://www.robocup2017.org/file/symposium/soccer_sim_2D/TDP_HELIOS2017.pdf
8. *Genetic Algorithm Optimized GravityBased RoboCup Soccer Team*, Tory Harter, July 28, 2015