To appear in *Proceedings of the RoboCup International Symposium 2016* (RoboCup 2016), Leipzig, Germany, July 2016.

Prioritized Role Assignment for Marking

Patrick MacAlpine and Peter Stone

Department of Computer Science, The University of Texas at Austin, USA
{patmac,pstone}@cs.utexas.edu

Abstract. This paper presents a system for marking or covering players on an opposing soccer team so as to best prevent them from scoring. A basis for the marking system is the introduction of prioritized role assignment, an extension to SCRAM dynamic role assignment used by the UT Austin Villa RoboCup 3D simulation team for formational positioning. The marking system is designed to allow for decentralized coordination among physically realistic simulated humanoid soccer playing robots in the partially observable, non-deterministic, noisy, dynamic, and limited communication setting of the RoboCup 3D simulation league simulator. Although it is discussed in the context of the RoboCup 3D simulation environment, the marking system is not domain specific and can readily be employed in other RoboCup leagues as prioritized role assignment generalizes well to many realistic and real-world multiagent systems.

1 Introduction

Coordinated movement among autonomous mobile robots is an important research area with many applications such as search and rescue and warehouse operations. The RoboCup 3D simulation soccer competition provides an excellent testbed for this line of research as it requires coordination among autonomous agents in a physically realistic environment that is partially observable, non-deterministic, noisy, and dynamic. The presence of adversarial opponent agents gives rise to the challenging coordination problem of how to cover or mark opponents so as to best prevent the opposing team from scoring. Agents must work together as a team to counteract their opponents' actions while maximizing their own game performance.

A common paradigm for specifying where players on a soccer field should move is the use of formations. Positioning players in a formation requires agents to coordinate with each other and determine where each agent should position itself on the field. The work in this paper focuses on role assignment—specifically tackling the problem of assigning interchangeable mobile robots to move to a set of target positions in a formation such that robots are present at their assigned positions in as little time as possible. Previous work on role assignment, and the basis for this work, is that of Scaleable Collision-avoiding Role Assignment with Minimal-makespan (SCRAM) role assignment functions [12]. SCRAM role assignment functions assign robots to target positions such that the makespan (time for all robots to reach their target positions) is minimized while avoiding collisions between robots. SCRAM only minimizes the completion time of the entire formation however, and does not take into consideration the possibility of some target positions being more important—and thus needing a robot to arrive at them sooner—than other potentially less important positions. Within the context of soccer we find it is often a high priority for a player to arrive as quickly as possible at a position for marking an opponent in a dangerous offensive location. Marking engenders the need for prioritized role assignment in which higher priority positions in a formation (those for marking opponents) are assigned to be reached by robots before lower priority positions.

Primary contributions of this paper are twofold. First, we introduce a new extension of SCRAM role assignment that allows for prioritization of roles. Second, we provide a detailed description and analysis of a marking system, incorporating prioritized role assignment, that we have implemented for use in the RoboCup 3D simulation domain.

The remainder of the paper is organized as follows. In Section 2 a description of the 3D simulation domain is given. Section 3 provides background information on SCRAM role assignment, while Section 4 introduces and motivates prioritized SCRAM role assignment. Section 5 details a marking system using prioritized role assignment, and analysis of the performance of the marking system is given in Section 6. Related work is discussed in Section 7 and Section 8 concludes.

2 Domain Description

The RoboCup 3D simulation environment is based on SimSpark [1], a generic physical multiagent system simulator. SimSpark uses the Open Dynamics Engine[1] (ODE) library for its realistic simulation of rigid body dynamics with collision detection and friction. ODE also provides support for the modeling of advanced motorized hinge joints used in the humanoid agents.

Games consist of 11 versus 11 agents playing on a 30 meters in length by 20 meters in width field. The robot agents in the simulation are modeled after the Aldebaran Nao robot,[2] which has a height of about 57 cm, and a mass of 4.5 kg. Each robot has 22 degrees of freedom: six in each leg, four in each arm, and two in the neck. In order to monitor and control its hinge joints, an agent is equipped with joint perceptors and effectors. Joint perceptors provide the agent with noise-free angular measurements every simulation cycle (20 ms), while joint effectors allow the agent to specify the speed and direction in which to move a joint.

Visual information about the environment is given to an agent every third simulation cycle (60 ms) through noisy measurements of the distance and angle to objects within a restricted vision cone (120°). Agents are also outfitted with noisy accelerometer and gyroscope perceptors, as well as force resistance perceptors on the sole of each foot. Additionally, agents can communicate with each other every other simulation cycle (40 ms) by sending 20 byte messages.

[1] `http://www.ode.org/`

[2] `http://www.aldebaran-robotics.com/eng/`

3 SCRAM Role Assignment

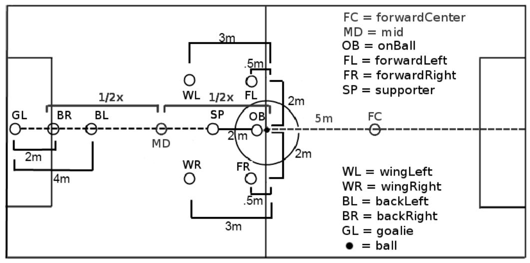

Fig. 1: Example formation where role positions are computed as offset positions from the ball.

Given a desired team formation, such as the formation used by UT Austin Villa in Figure 1, we need to map players to roles (target positions on the field). A naïve mapping having each player permanently mapped to one of the roles performs poorly due to the dynamic nature of the game. With such static roles an agent assigned to a defensive role may end up out of position and, without being able to switch roles with a teammate in a better position to defend, allow for the opponent to have a clear path to the goal.

A role assignment algorithm can be thought of as implementing a role assignment *function*, which takes as input the state of the world, and outputs a one-to-one mapping of players to roles. SCRAM role assignment functions [12] compute in polynomial time mappings from players to roles that both minimize the longest distance any agent has to travel (thereby minimizing the makespan or formation completion time) and avoid collisions between agents.

The SCRAM Minimum Maximal Distance Recursive (MMDR) function, which is a basis for the work in this paper, finds a mapping of agents to target positions which recursively minimizes the maximum distance that any agent travels. Let $\mathbb{M}$ be the set of all one-to-one mappings between agents and roles. If there are n agents and n target role positions, then there are $n!$ possible mappings $M \in \mathbb{M}$. Let the *cost* of a mapping M be the n-tuple of distances from each agent to its target, sorted in decreasing order. We can then sort all the $n!$ possible mappings based on their costs, where comparing two costs is done lexicographically. The lowest cost mapping is the one returned by MMDR. Sorted costs of mappings for a small example are shown in Figure 2.

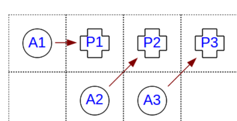

Fig. 2: Lowest lexicographical cost (shown with arrows) to highest cost ordering of mappings from agents (A1,A2,A3) to role positions (P1,P2,P3). Each row represents the cost of a single mapping.

1: $\sqrt{2}$ (A2→P2), $\sqrt{2}$ (A3→P3), 1 (A1→P1)
2: 2 (A1→P2), $\sqrt{2}$ (A3→P3), 1 (A2→P1)
3: $\sqrt{5}$ (A2→P3), 1 (A1→P1), 1 (A3→P2)
4: $\sqrt{5}$ (A2→P3), 2 (A1→P2), $\sqrt{2}$ (A3→P1)
5: 3 (A1→P3), 1 (A2→P1), 1 (A3→P2)
6: 3 (A1→P3), $\sqrt{2}$ (A2→P2), $\sqrt{2}$ (A3→P1)

Proof that the MMDR role assignment function both minimizes the makespan and avoids collisions among agents, as well as $O(n^5)$ and $O(n^4)$ algorithms for computing MMDR, can be found in [12].

4 Prioritized SCRAM Role Assignment

It is not always the case that minimizing the makespan (completing a formation as fast as possible) is what is best for a team of robots. There are cases where it is preferable to have a subset of high priority role positions in a formation be reached by agents as soon as possible. One example of this is soccer where it is often desirable for players to arrive as fast as possible at positions for marking opponents in dangerous offensive locations. This section introduces a new extension to SCRAM role assignment allowing for subsets of role positions to be given different priorities.

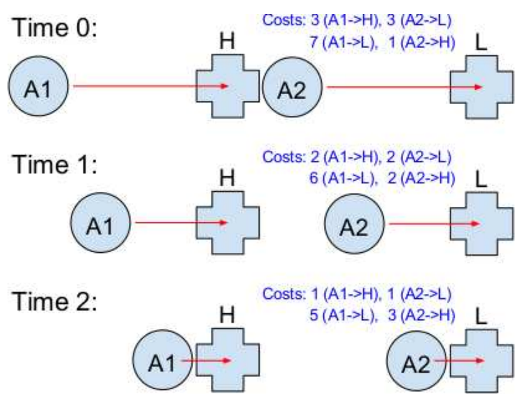

Fig. 3: Agents A1 and A2 being assigned and moving to the high priority (H) and low priority (L) target positions using the MMDR SCRAM role assignment algorithm.

Figure 3 shows an example of two agents being assigned to both high priority (H) and low priority (L) target positions using the MMDR SCRAM role assignment algorithm. As MMDR does not take into account priorities of different positions, the high priority position H will not be reached by an agent until time = 3 despite agent A2 starting only a distance of 1 from H.

To bias MMDR into producing an assignment that has agents reach all high priority positions as fast as possible we can add a large priority value P to costs for reaching all high priority positions. As long as P is greater than all possible distances to lower priority positions, MMDR will assign the closest agents to high priority positions before considering the assignment of agents to lower priority positions. This bias of MMDR assigning closer agents to higher priority positions is due to all costs to higher priority positions being greater and thus needing to be minimized before that of costs to lower priority positions.

Figure 4 shows an example of two agents being assigned to both high priority (H) and low priority (L) target positions using the MMDR SCRAM role assignment algorithm, but with a large priority value P added to the costs of reaching

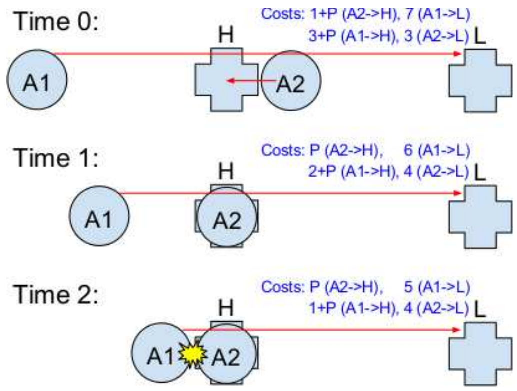

Fig. 4: Agents A1 and A2 being assigned and moving to the high priority (H) and low priority (L) target positions using the MMDR SCRAM role assignment algorithm, but with a large priority value P added to the costs of reaching H. At time = 2 agents A1 and A2 collide with each other.

H. This results in H being reached at time = 1 by agent A2, but unfortunately later agent A1, on its way to its assigned position L, collides with A2. Assigning the closest agents to high priority target positions, and thereby no longer necessarily recursively minimizing the maximum distance that any agent must travel to reach its assigned target, breaks the collision avoidance property of MMDR.

To preserve collision avoidance, but still prioritize a subset of targets being reached as fast as possible, we can define a priority distance D around high priority targets for which agents within D distance of a target will not have the priority value P added to the cost of that target.

$$\texttt{cost}(agent,target) = \begin{cases} |\overline{agent,target}| + \mathrm{P} \text{ if } |\overline{agent,target}| > \mathrm{D} \\ |\overline{agent,target}| \qquad\quad \text{otherwise} \end{cases}$$

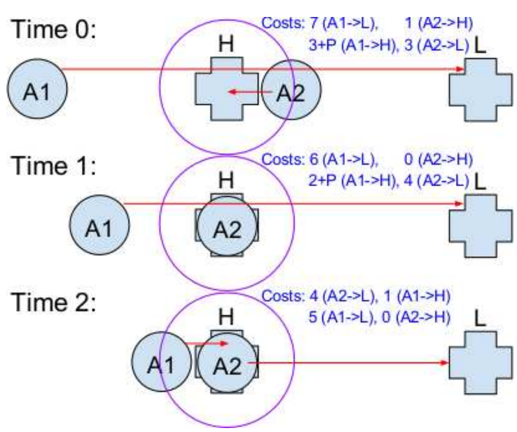

Fig. 5: Agents A1 and A2 being assigned and moving to the high priority (H) and low priority (L) target positions using the MMDR SCRAM role assignment algorithm, but with a large priority value P added to the costs of reaching H for any agents outside the priority distance of H (the purple circle). At time = 2 agents A1 and A2 switch targets due to agent A1 being within the priority distance of H.

Figure 5 shows an example of two agents being assigned to both high priority (H) and low priority (L) target positions using the MMDR SCRAM role assignment algorithm, but with a large priority value P added to the costs of reaching H when agents are outside a priority distance D of H. This results in H being reached at time = 1 by agent A2, and then later when agent A1 gets within D of H agents A1 and A2 switch targets and avoid colliding.

Defining a priority distance D causes agents to arrive within a distance D of all high priority targets as fast as possible. Although agents might not arrive exactly at the high priority targets in as little time as possible, this is often fine for many applications including marking in soccer. When marking a player does not need to be right next to an opponent, but just within a close enough distance to the opponent to be able to react quickly and prevent the opponent from receiving the ball. Assuming D is not too large, should a player come within D distance of an opponent who is already being marked by a teammate, it should then be safe for the players to switch who is marking the opponent.

Augmenting MMDR with a large P priority value and D priority distance for high priority positions extends SCRAM role assignment to allow for prioritization of targets. The collision avoidance property of SCRAM, based on the triangle inequality and fully explained in [12], is still preserved with prioritization as any agents within D distance of a high priority target will switch targets before they collide.

It is possible to have multiple subsets of targets with different priorities, or a hierarchy of prioritization, by assigning different P values to different subsets of targets. An example of this is given later in Section 5.3 for which a highest priority target is given a priority value of P_s, and other high priority targets are given a priority value of P_m, where $P_s >> P_m$.

5 Marking System

The marking system implemented by the UT Austin Villa team is a sequential process encompassing the following steps:

1. Decide which players to mark
2. Select which roles to use for marking purposes
3. Use prioritized role assignment to assign players to positions

Each of these steps (shown in Figure 6), as well as additional details of the marking system, are described in the following subsections.

5.1 Deciding Who to Mark

The first step in the marking system is to decide which if any opponents should be marked (those opponents considered to be in dangerous offensive positions). The decision on whether or not to mark an opponent is heuristic based and uses the following rules:

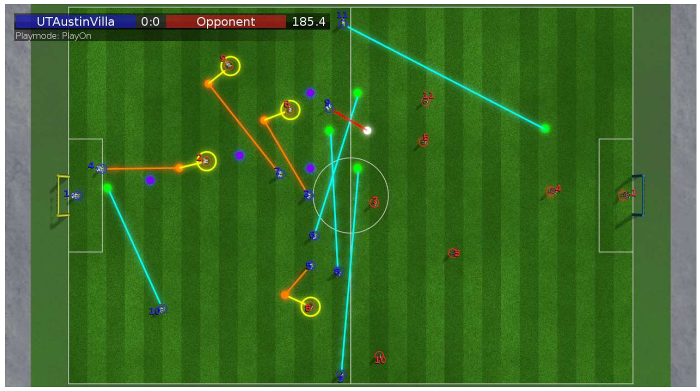

Fig. 6: Steps of the marking system. The white dot is the ball.
Step 1 (Deciding Who to Mark): Opponent agents selected to be marked are circled in yellow.
Step 2 (Selecting Roles for Marking): Green dots represent target formation positions with purple dots representing target formation positions that have been selected to be replaced by the orange dot marking positions.
Step 3 (Assigning Roles): Orange lines represent agents assigned to marking positions, light blue lines represent agents assigned to target formation positions, and the red line shows the agent assigned to go to the ball.

1. Opponent is close enough to take a shot on goal
2. Opponent is not the closest opponent to the ball
3. Opponent is not too close to the ball
4. Opponent is not too far behind the ball

The first rule suggests that an opponent is in a dangerous scoring position. As we always send one player to the ball (the *onBall* role in Figure 1), the second and third rules prevent marking of opponents when we should already have a player moving toward their positions. The fourth rule is due to very few teams passing the ball backwards. Figure 6 shows opponent agents selected to be marked circled in yellow.

5.2 Selecting Roles for Marking

The next step is to select which formation role positions should be given up in favor of having agents who would otherwise be assigned to those roles instead move to marking role positions. Marking role positions are calculated as the position 1.5 meters from a marked opponent along the line from that opponent to the center of our goal (shown in orange in Figure 6). The selection of formation positions to replace with marking positions is determined by using the Hungarian algorithm [4] to compute the minimum sum of distances matching between all formation and marking positions in a bipartite graph. This matching results in the closest formation positions to marking positions being replaced by the marking positions that they are nearest. If there are ever more marking positions than available formation positions then some marking positions will be matched to dummy nodes in the bipartite graph and not be assigned to an agent.

Figure 6 shows the result after selecting formation positions to be used as marking role positions with the formation positions selected drawn in purple, and those not selected and still being used drawn in green.

5.3 Assigning Roles

Agents are assigned to marking positions and formation positions (Figure 6 shows these assignments in orange and light blue respectively) using prioritized SCRAM role assignment discussed in Section 4. Marking positions are considered higher priority than formation positions, and use a priority value $P_m = 100$ and a priority distance $D_m = 3$. Additionally, when a teammate is kicking the ball, a couple of players are assigned to high priority kick anticipation position roles near the location where the ball is being kicked to [10]. Kick anticipation roles are given the same priority value and distance as marking roles.

There are several roles in Figure 1 that are never reassigned to be marking roles. The *goalie* role is always assigned to a single agent designated as the goalie who is allowed to dive and block a ball when an opponent takes a shot on goal. The *onBall* role is always assigned to the agent closest to the ball as it is that agent's job to gain possession of the ball. The *supporter* role is also very important as the *supporter* is in a critical position right behind the ball should the *onBall* role agent lose possession of the ball. The *supporter* is considered a higher priority role than marking roles, and thus uses a priority value $P_s = 10000$ along with a priority distance $D_s = 1.5$. All P and D values were chosen experimentally.

5.4 Coordination

In order for agents on a team to assume correct positions on the field they all must coordinate and agree on which mapping of agents to roles to use. If every agent had perfect information of the locations of the ball and its teammates this would not be a problem as each could independently calculate the optimal mapping to use. Agents do not have perfect information, however, and are limited to noisy measurements of the distance and angle to objects within a restricted vision cone (120°). To synchronize their assigned roles agents using a voting coordination system as described in [9].

Coordination can become more difficult if an opponent is standing in a position right on the borderline of whether or not the opponent should be marked. To prevent thrashing between different role assignments in such a situation, opponents who are currently being marked must move at least .25 meters outside a mark-able position on the field before they will stop being marked. Also, to prevent thrashing between different selections of formation positions to use for marking, a selection is never changed (assuming the cardinality of matchings are the same) unless the new selection's matching's sum of distances is at least one meter less than the previous selection's matching's sum of distances.

6 Results and Analysis

After the 2015 RoboCup competition was over we played 1000 games of our team's released binary against all teams' released binaries (this includes playing against ourselves) and found that only the UTAustinVilla and FCPortugal teams' binaries were able to score over 100 goals against our released binary [11]. Both the UTAustinVilla and FCPortugal teams created set plays allowing them to score quickly off kickoffs which empirically we found to be the source of the majority of the goals that they scored against our released binary (74.5% of goals for UTAustinVilla and 78.2% of goals for FCPortugal)

To test the effectiveness of our marking system using prioritized role assignment we played 1000 games against both the UTAustinVilla and FCPortugal teams' released binaries using the marking system (`Prioritized Marking`). We also played 1000 games against both teams without using marking (`No Marking`) as well as with marking but using normal non-prioritized SCRAM role assignment (`Marking No Prioritization`). Results of the number of goals against scored by opponents can be seen in Table 1, and an analysis of the scoring percentage of opponents' set plays is shown in Table 2.

Table 1: Number of goals against when playing 1000 games against the released binaries of UTAustinVilla and FCPortugal from RoboCup 2015.

Opponent	No Marking	Marking No Prioritization	Prioritized Marking
UTAustinVilla	1525	336	319
FCPortugal	230	40	37

Table 2: Scoring percentage of opponents' set plays when playing 1000 games against the released binaries of UTAustinVilla and FCPortugal from RoboCup 2015.

Set Play	No Marking	Marking No Prioritization	Prioritized Marking
UTAustinVilla Kickoff	48.31	0.16	0.16
FCPortugal Kickoff	6.22	0.06	0.06
UTAustinVilla Corner Kick	15.97	12.31	7.59

Table 1 shows a dramatic drop in the number of goals scored by opponents when using marking. There is also a small decrease in the number of goals against when using prioritized SCRAM instead of non-prioritized SCRAM for marking. Table 2 reveals the source of the reduction in goals against as using marking almost completely eliminates the opponents' abilities to score on kickoff set plays. FCPortugal's kickoff (shown in Figures 7, 8, and 9) consists of a player first passing the ball backwards on the kickoff to a waiting player who then passes the ball forward to a teammate running forward to a dangerous offensive position on the side of the field. UTAustinVilla's kickoff and corner kick set plays are described in [11]. Although the numbers in Table 2 do not show an advantage in using prioritized SCRAM over non-prioritized SCRAM against kickoffs, we have seen some instances such as in Figure 8 where not using prioritization is harmful. Prioritized SCRAM does however show an advantage against UTAustinVilla's corner kick set plays. Videos of set plays and marking are available online.[3]

Overall using marking, and to a greater extent using marking with prioritized SCRAM role assignment, provides a considerable defensive advantage when

[3] `http://www.cs.utexas.edu/~AustinVilla/sim/3dsimulation/AustinVilla3DSimulationFiles/2016/html/marking.html`

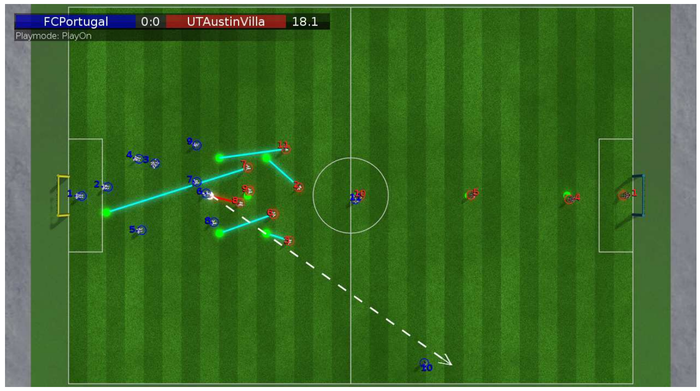

Fig. 7: Not marking against FCPortugal kickoff. Dashed white line shows trajectory of ball during pass. Not marking allows for an opponent to run forward and receive a pass in an open position to score a goal (blue 10 is not marked).

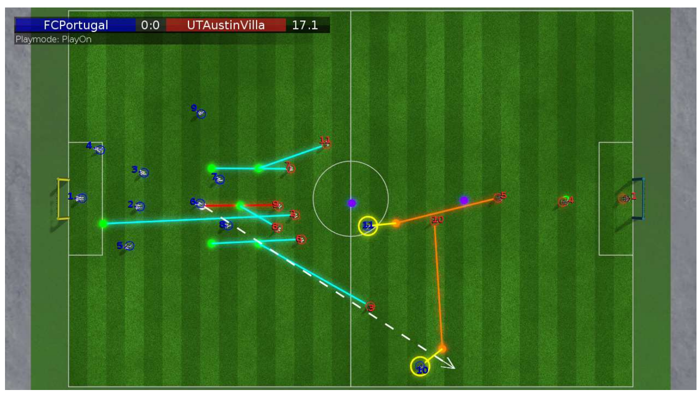

Fig. 8: Marking, but not prioritized, against FCPortugal kickoff. Dashed white line shows trajectory of ball during pass. Not using prioritization with marking results in a player assigned to mark an opponent being too far away from that assigned opponent to prevent the opponent from scoring a goal (red 10 instead of red 3 assigned to mark blue 10).

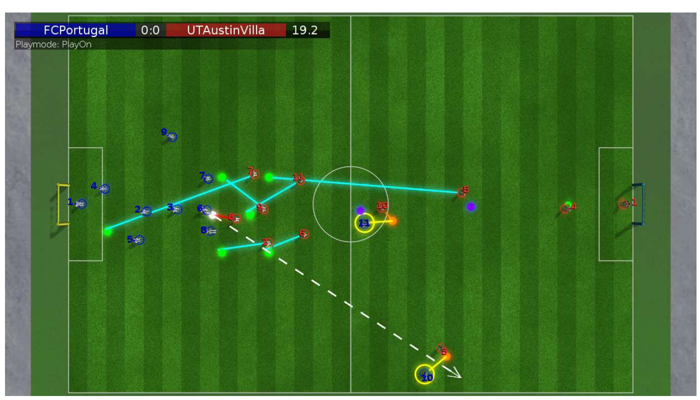

Fig. 9: Prioritized marking against FCPortugal kickoff. Dashed white line shows trajectory of ball during pass. Prioritized marking prevents opponents from receiving a pass in an open position to score a goal (red 3 marking blue 10).

playing soccer against opponents who use set plays. The average goal difference across 1000 games when playing against UTAustinVilla with prioritized marking improved to 0.657 (+/-0.028) from $\sim$0 without marking, and this same number against FCPortugal improved to 2.530 (+/-0.040) with prioritized marking from 2.476 (+/-0.043) without marking. We also played 1000 games against the other ten teams' released binaries from RoboCup 2015, none of which we knew to use passing for set plays, and found no measurable difference in goals against or game performance when using marking versus those teams.

7 Related Work

While there has been considerable research focused on role assignment in the 2D soccer simulation domain (for example by Stone et al. [15] and Reis et al. [14]), relatively little outside of [2] has been published on this topic in the more physically realistic 3D soccer simulation environment. [2], as well as related work in the RoboCup Middle Size League (MSL) [6], rank positions on the field in order of importance and then iteratively assign the closest available agent to the most important currently unassigned position until every agent is mapped to a target location. Previous approaches to marking in 2D simulated soccer have included using a fuzzy logic inference system [8], a pareto-optimal approach that attempts to maximize the total prevented threat of opponents while minimizing the total time needed to move to positions [5], and opponent modeling [3]. In the RoboCup Small Size League (SSL) teams have computed threat levels of opponents when deciding who to mark [7, 13]. The work presented in this paper differs from the mentioned previous work as it takes into account real-world concerns and movement dynamics, such as the need for avoiding collisions of robots, when assigning players to mark opponents.

8 Conclusion

We have introduced a new extension of SCRAM role assignment that allows for prioritization of roles. This prioritized role extension to SCRAM has been incorporated into a marking system that is very effective in defending against set plays used by teams within the RoboCup 3D simulation domain. Ongoing work includes improving the heuristics in our marking system for identifying who to mark, as well as possibly adding different priorities for different marking positions. A promising direction for future work is to apply SCRAM prioritized role assignment to applications outside of marking and RoboCup such as general patrol and coverage tasks.

Acknowledgments

This work has taken place in the Learning Agents Research Group (LARG) at UT Austin. LARG research is supported in part by NSF (CNS-1330072, CNS-1305287), ONR (21C184-01), and AFOSR (FA9550-14-1-0087). Peter Stone serves on the Board of Directors of, Cogitai, Inc. The terms of this arrangement have been reviewed and approved by UT Austin in accordance with its policy on objectivity in research.

References

1. Boedecker, J., Asada, M.: Simspark–concepts and application in the robocup 3d soccer simulation league. Autonomous Robots pp. 174–181 (2008)
2. Chen, W., Chen, T.: Multi-robot dynamic role assignment based on path cost. In: 2011 Chinese Control and Decision Conference (CCDC). pp. 3721 –3724 (May 2011)
3. Habibi, J., Younesy, H., Heydarnoori, A.: Using the opponent pass modeling method to improve defending ability of a (robo) soccer simulation team. In: RoboCup 2003: Robot Soccer World Cup VII, pp. 543–550. Springer (2003)
4. Kuhn, H.W.: The hungarian method for the assignment problem. Naval Research Logistics Quarterly 2(1-2), 83–97 (1955)
5. Kyrylov, V., Hou, E.: Pareto-optimal collaborative defensive player positioning in simulated soccer. In: RoboCup 2009: Robot Soccer World Cup XIII, pp. 179–191. Springer (2009)
6. Lau, N., Lopes, L., Corrente, G., Filipe, N.: Multi-robot team coordination through roles, positionings and coordinated procedures. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2009). pp. 5841 –5848 (oct 2009)
7. Li, C., Xiong, R., Ren, Z., Tang, W., Zhao, Y.: Zjunlict: Robocup 2014 small size league champion. In: RoboCup 2014: Robot Soccer World Cup XVIII, pp. 47–59. Lecture Notes in Artificial Intelligence, Springer (2015)
8. Li, X., Chen, X.: Fuzzy inference based forecasting in soccer simulation 2d, the robocup 2015 soccer simulation 2d league champion team. In: RoboCup 2015: Robot Soccer World Cup XIX, pp. 144–152. Lecture Notes in Artificial Intelligence, Springer (2015)
9. MacAlpine, P., Barrera, F., Stone, P.: Positioning to win: A dynamic role assignment and formation positioning system. In: RoboCup 2012: Robot Soccer World Cup XVI. Lecture Notes in Artificial Intelligence, Springer Verlag, Berlin (2013)
10. MacAlpine, P., Depinet, M., Liang, J., Stone, P.: UT Austin Villa: RoboCup 2014 3D simulation league competition and technical challenge champions. In: RoboCup 2014: Robot Soccer World Cup XVIII. Lecture Notes in Artificial Intelligence, Springer Verlag, Berlin (2015)
11. MacAlpine, P., Hanna, J., Liang, J., Stone, P.: UT Austin Villa: RoboCup 2015 3D simulation league competition and technical challenges champions. In: RoboCup 2015: Robot Soccer World Cup XIX. Lecture Notes in Artificial Intelligence, Springer (2015)
12. MacAlpine, P., Price, E., Stone, P.: SCRAM: Scalable collision-avoiding role assignment with minimal-makespan for formational positioning. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15) (January 2015)
13. Mendoza, J.P., Biswas, J., Zhu, D., Wang, R., Cooksey, P., Klee, S., Veloso, M.: CMDragons 2015: Coordinated Offense and Defense of the SSL Champions. In: RoboCup 2015: Robot Soccer World Cup XIX. Lecture Notes in Artificial Intelligence, Springer (2015)
14. Reis, L., Lau, N., Oliveira, E.: Situation based strategic positioning for coordinating a team of homogeneous agents. In: Balancing Reactivity and Social Deliberation in Multi-Agent Systems, Lecture Notes in Computer Science, vol. 2103, pp. 175–197. Springer Berlin / Heidelberg (2001)
15. Stone, P., Veloso, M.: Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork. Artificial Intelligence 110(2), 241–273 (June 1999)