Capture The Flag

Report

Mafalda Ferreira, 92513
Department DEI
Instituto Superior Técnico
Lisbon, Lisbon, Portugal
mafaldasofiacf@tecnico.ulisboa.pt

Rita Oliveira, 92546
Department DEI
Instituto Superior Técnico
Lisbon, Lisbon, Portugal
ritatmcoliveira@tecnico.ulisboa.pt

Francisco Cabrinha, 102141
Department DEI
Instituto Superior Técnico
Lisbon, Lisbon, Portugal
francisco.cabrinha@tecnico.ulisboa.pt

ABSTRACT

In this project, the game Capture The Flag will be developed. This game is a multiagent system where the environment is dynamic and unpredictable. For that reason, the agents need to have the capability to learn and cooperate with others. This is what we call multi-agent learning: many individual agents must act independently, yet learn to interact and cooperate with other agents. With this game, agents can learn teamwork, an ability needed to solve various real-word problems.

CCS CONCEPTS

• Artificial Intelligence • Statistics

KEYWORDS

Multi-Agent System, Deliberative Agents, Coordination Game, Capture The Flag, Communication, Cooperation, Social Ability.

1 Introduction

1.1 Motivation

It is important that an intelligent system learns social ability to solve real-world problems. Online games and traditional sports are great systems for this purpose. For that reason, the game Capture the Flag will be the system of this learning due to the complex cooperation that it contains.

1.2 Related Work

Multi-agent systems are difficult to coordinate due to the complexity of the systems. As opposed to purely reactive agents, logic-based reasoning may lead to an optimal performance [1].

1.3 Problem Definition and Relevance

Capture the Flag is a team-based multiplayer game that owns multiple variations. In our approach, the game considers players distributed among two teams of different colors, red and blue. There are multiple red and blue flags as well as colored defensive and 2 delivery areas. The flags are initially spawned in the defensive areas with the respective color, as well as the agents. Each team needs to capture as many flags as possible of the opposite color and deliver them. Let R and B be red or blue opposite colors. The game complies to the following rules:

- A R agent can steal a B flag from a B defensive area.
- A R agent can steal an B flag from a B agent that is not defending an area
- A R agent can recover (by using the action steal) a R flag stolen from B agent and return it to the defensive area R
- A R agent can freeze a B agent (for 5 steps) inside a R defensive area.
- A R agent delivers a B flag in a R delivery area to gain 1 point. One team wins if they reach 10 points.
- After delivering a B flag, it is spawned again in a B defensive area.
- Agents have to decide between stealing or freezing the enemy
- Three agents of each team, three defensive areas and two delivery areas

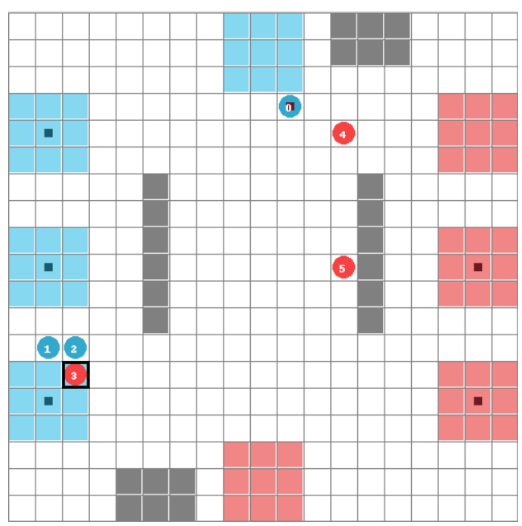


Figure 1: **Capture the Flag board**

The study of the game will allow us to assess which are the best models and techniques for the implementation of cooperation and coordination of autonomous agents.

1.4 **Objectives**

The project focuses on creating and evaluating on how agents cooperate together as a team to achieve the same objective.

Ultimately, we try to design an optimal system with intelligent agents that make the best decisions in order to increase the team effiencity/wins, and reduce the time that a team takes to win a game.

2 **Approach**

2.1 **Environment**

2.2 **Multi-Agent System**

The Capture the Flag game is a multi-agent system in which all agents are **deliberative agents**. **Social Convention** and **Roles** were used to solve coordination problems. The three agents in each team,will need to coordinate to determine who will steal and who will freeze

2.2.1 Deliberative Agent

Deliberative Agents with mental attitudes of beliefs, desires and intentions will build a plan to achieve the goals. These agents will **replan** when the plan goes wrong, will stop to determine whether intentions have **succeeded** or are **impossible** and will **reconsider** intentions after every action. Desires are divided into seven parts:

1. Deliver the flag to team delivery area
2. Deliver the flag to team defensive area
3. Steal the flag from enemy defensive area
4. Steal the flag from the enemy in movement
5. Freeze the enemy in the team defensive area
6. Go to enemy defensive area
7. Go to team defensive area

The selection of the best intention depends on **beliefs** and **on the first desire produced**. Intentions will be the coordinates where the agent needs to go. In this way, the intentions of each desire are:

1. Coordinates of team delivery area
2. Coordinates of team defensive area to deliver the flag
3. Coordinates of enemy defensive area to steal the flag
4. Coordinates of enemy in movement to steal the flag
5. Coordinates of enemy to freeze
6. Coordinates of enemy defensive area
7. Coordinates of team defensive area

The agent replans the plan when it **verifies that the intention changed,** meaning that the plan goes wrong. The agent determines whether intentions have succeeded if the agent **arrives at the destination or if it is close to the enemy in the case the agent desires to freeze or steal the enemy**. The agent determines whether the intention is impossible if the **agent is frozen** or **if the agent intended to freeze the enemy but none of the enemy is close.** The agent reconsiders intentions if it intended **to deliver the flag but it does not have the flag (because it was sloten)** or if intended to freeze but the enemy is not in the location that the agent believed it was.

2.2.2 Social Convention

The three agents of each team will need to coordinate the two actions: steal and freeze. In order to win the game, agents **need to steal the enemy's flag and deliver it to the delivery area's team**. Agents can recover the stolen team's flag (by stealing from the opponent that has the flag) and can freeze in order to defend the defensive area's team. **The agent that steals the flag also delivers it.** If agents of the team **only concentrate on freezing** then this team **will never win the game** because none of them stole flags and delivered them. That is why all agents will have the payoff zero. If agents of the team **only concentrate on stealing** then the team **has the possibility to win**. However, the **opponents can easily steal the team's flag** because none of the agents froze the enemy. The only way that the team can defend is recovering the flags. For that reason, all agents will have the payoff one. If **two agents concentrate on freezing and one agent concentrates on stealing** then the team has a **greater possibility to win than the previous case**. So, all agents will have the payoff two. If **two agents concentrate on stealing and one agent concentrates on freezing** then the team has a **greater possibility to win than the previous case** because more agents are stealing the flags.

Joint actions(a)	u1(a)	u2(a)	u3(a)
(Steal,Steal,Steal)	1	1	1
*(Steal,Steal,Freeze)	3	3	3
*(Steal,Freeze,Steal)	3	3	3
*(Freeze,Steal,Steal)	3	3	3
(Steal,Freeze,Freeze)	2	2	2
(Freeze,Freeze,Steal)	2	2	2
(Freeze,Steal,Freeze)	2	2	2
(Freeze, Freeze,Freeze)	0	0	0

* Nash equilibrium

Table 1: **Payoff of each agent depending on joint actions. u1, u2 and u3 correspond to the payoff of agent 1, agent 2 and agent 3, respectively.**

In this table, it can be verified that agents are faced with the problem of how to choose their actions to select the same Nash equilibrium.To solve this problem, **Social Convention** was used, which uses a particular ordering scheme of joint actions. Agents select the first equilibrium according to this ordering scheme, which is constructed by:

- Order of agents: Agent 1 > Agent 2 > Agent 3
- Order of actions: Steal > Freeze

That means agent 1 has 'priority' over agent 2, agent 2 has 'priority' over agent 3 and the action steal has 'priority over the action freeze.

According to the ordering scheme, it is concluded that the first equilibrium is: **(Steal,Steal,Freeze)**.

In this way, in the CTF game agents 1 and 2 will be focused on stealing and delivering the flags and agent 3 will be focused on defending and freezing.

2.2.3 Roles

An alternative is to assign roles to each agent.

A role restricts the number of actions that an agent can select, i.e, if an agent is given a role at the first step, some of its actions will be unavailable throughout the entire step/episode, according to the chosen architecture (e.g. an attacker agent cannot freeze another agent).

For our game, we defined the following roles:

- Roles of Agents: R = {Attacker, Defender}
- Order of Roles: Attacker > Defender
- Attacker: Agents can only steal an enemy flag and deliver it.
- Defender: Agents can only steal a flag of their own color, from an enemy (recover), reposition that flag in the defensive area, and freeze enemies inside their defensive area.

We also defined 2 potential functions:

1. Potential Function Attacker: Consists of the negative Manhattan distance between the agent position and the target position, which can either be:
   a. Enemy that has the flag;
   b. Enemy's defensive area.
2. Potential Function Defender: Consists of the negative Manhattan distance between the agent position and the target position, which is:
   a. Own defensive area.

For each agent, in each team, both potential functions are calculated. Then, a comparison is made between all agents within a team. Since the role Attacker has a higher priority than the role Defender, the Agent with the highest potential_function_attacker will be assigned with the Role Attacker. This will be done recursively until all Attacker Roles are assigned. Then, the same will be done to the Defender Role, but since there are only 2 Roles, and one of them is fully assigned, all other agents will be Defenders.

Note that every agent knows the potential function algorithm and is able to assign a role for every agent. Initially, every agent-role will compute its own potential functions and broadcast their result to their teammates. When all potential functions have arrived, each agent will perform the role assignment in parallel.

2.2.3.1 Fixed Roles Team

In Teams with Fixed Roles, each agent will be assigned to a Role without any calculations. They will simply follow the order of the Roles passed, i.e, [Agent0, Agent1, Agent2], [Defender, Attacker, Defender] will correspond to Agent 0 and Agent 2 being Defenders, and Agent 1 being an Attacker.

2.2.3.2 Roles Team

In Teams with Roles, each agent calculates both potential functions and makes comparisons with its teammates (as explained previously), within the first step, according to the spawn position. The Role assigned to that Agent no longer changes throughout the game.

2.2.3.3 Dynamic Roles with Social Convention

On the other hand, in Teams with Dynamic Roles, each agent calculates both potential functions at every step, meaning that an agent can have multiple roles throughout the entire game.

Sometimes, dilemmas occur between teammates, and they keep changing roles and just moving side to side, without any progress, since the potential function takes into account their position which is constantly changing. To help solve this Social Conventions in the Dynamic Roles Teams were implemented.

In Dynamic Roles with Social Conventions, there are two mode types, Role Mode and Social Conventions Mode. Initially, the agents are assigned different roles, similar to Dynamic Roles, having their mode set to Role Mode. However, an agent encounters a dilemma whenever it does at least one role change in 10 successive steps. To solve this, the mode is set to Social Convention Mode and the agent decides its actions according to it. After reaching 30 steps using Social Conventions, the agent returns to Role Mode.

2.3 Agent and Environment Properties

The environment contains randomly spawned boosts and obstacles such as walls and units of area that may randomly collapse. The environment is characterized as:

- **Fully accessible**: agents obtain accurate information;
- **Deterministic**: every action has a guaranteed effect;
- **Dynamic**: world may change due to a stolen/delivered flag or new obstacles when an agent is deliberating;
- **Discrete**: there is a finite number of agent actions;
- **Non-episodic**: each match is independent.

Multi-Agent system respect the following properties:

- **Proactiveness**: agents take initiative and decide;
- **Social ability**: agents interact with others;
- **Cooperation**: agents work with their team to achieve the same objective - win the game;
- **Coordination**: initially, agents distribute objectives;
- **Communication**: agents communicate within the board, regardless of the range they are a part of;
- **Adaptatively**: agents learn from previous actions;
- **Rationality**: agents make a decision;
- **Mobility**: agents move through the environment.

2.4 System Architecture

The environment has the following states: agents and their team color, flags and their color, defensive/delivery areas and walls positions. Additionally, in order to support the actions, the environment keeps track of the agents carrying a flag and frozen agents and their timestep remaining to become free.

2.4.1 Sensors

Agents have a variety of sensors to detect and interact with other agents:

- Verify if the agent/flag/defensive area is red or blue;
- Verify if the agent is frozen;
- Verify if there is a wall at the front;
- Inspect and verify if the unit of area is empty or not.

2.4.2 Actuators and Actions

Agents have different actuators to act using their sensors:

- Steal red/blue flag from opposite team agent;
- Defend own team's defensive area;
- Freeze the opposite team agent;
- Move to a given unit of area;

2.4.3 Observation space

Agents have a well defined observation space. They have an unlimited **observation (A)** range, within the area of the map, that reveals positions of all existing defensive, delivery areas. They also have limited **observation (B)** defined as 8 by 8 regarding the opposite team, team members and obstacles.

2.4.4 System requirements

Capture The Flag is a dynamic fast paced game. Thus, we need to take into account the following requirements:

- A system that operates effectively in time with low computational complexity and fast decision time;
- Intelligent behavior combining coordination within a team and interaction with the environment;
- A system where agents can learn;
- An unpredictable environment with new obstacles to test the reactivity of the agents

Agents are able to communicate with their teammates in order to acknowledge what are the different objectives of each agent.

2.5 Design choices

The deliberative agent is a very effective option for a complex system that has multiple goals that depend on beliefs and observation. Social Conventions are important when agents are faced with the problem of how to choose their actions to select the same Nash equilibrium. Roles allow reducing agents' sets of actions, which drive computational advantages in terms of speed.

3 Empirical Evaluation

3.1 Metrics

Setup for the analysis:

- 30 games between Teams;
- Teams of 3 Agents;
- Each Team has 3 flags of their color, one in each defensive area;
- Each Team has 3 defensive areas and 1 delivery area of their color;
- Each Role Team has 2 Defender and 1 Attacker;
- Social Convention Team has Steal, Steal and Freeze Actions;
- Freeze Time is 5 steps;
- Each Delivered Flag, of the opposite color, gets the Team 1 point;
- Game(Episode) ends when one of the teams reach 10 point or 1000 steps (timeout);

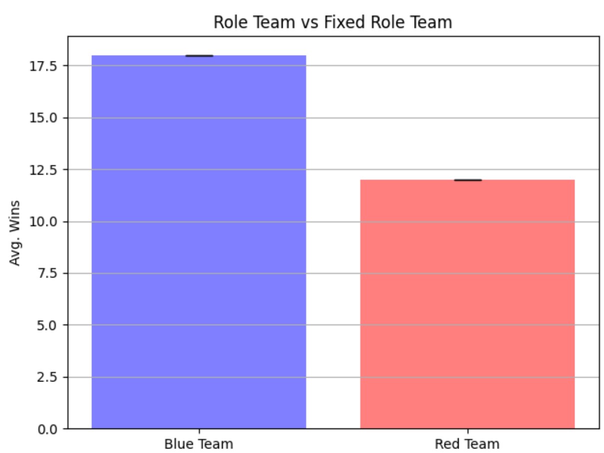
Average and Standard Deviation of the following metrics:

- Wins, per Team;
- Stolen Flags, per Team;
- Recovered Flags, per Team;
- Delivered Flags, per Team;
- Frozen Agents, per Team;
- Steps per episode for each game (each game is formed by 2 teams).

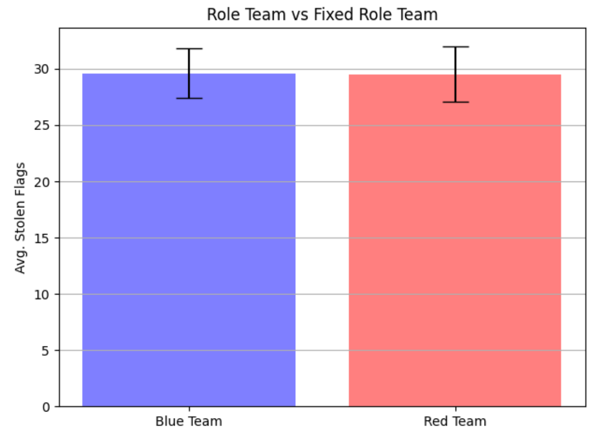
3.2 **Comparative analysis**

3.2.1 Experimental analyses

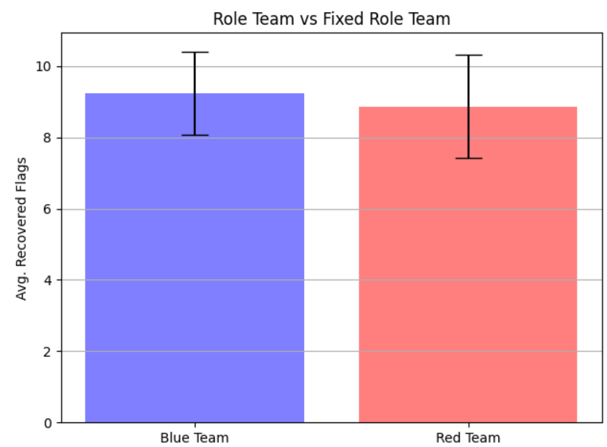
Role Team vs Fixed Role Team



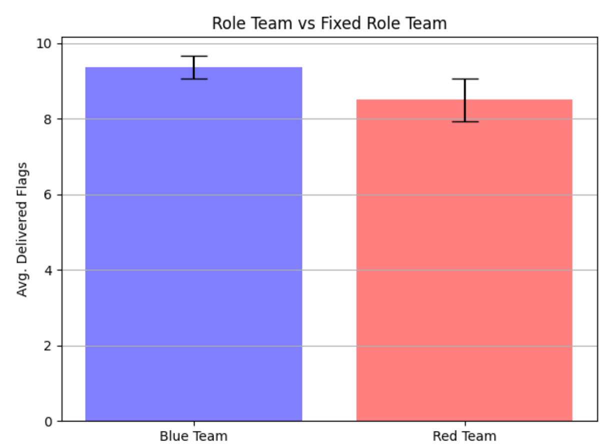
Graph 1. Role Team vs Fixed Role Team Average Wins



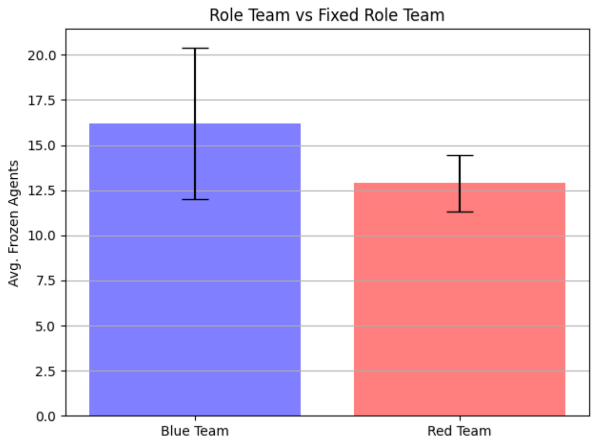
Graph 2. Role Team vs Fixed Role Team Average Stolen Flags



Graph 3. Role Team vs Fixed Role Team Average Recovered Flags

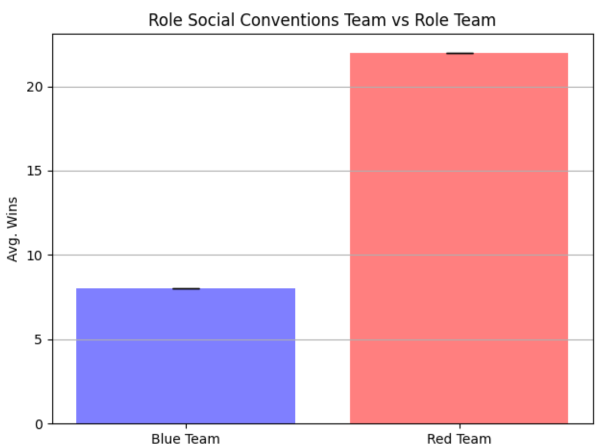


Graph 4. Role Team vs Fixed Role Team
Average Delivered Flags

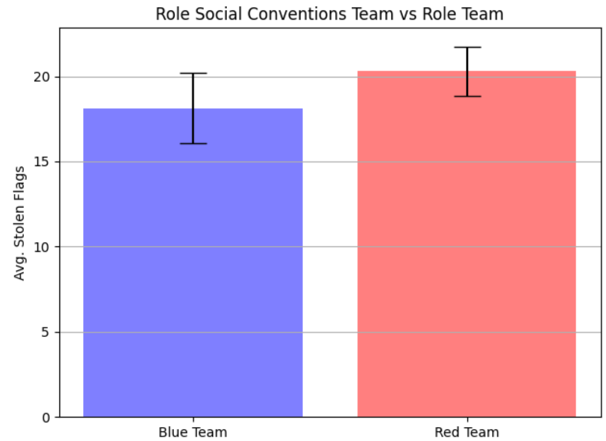


Graph 5. Role Team vs Fixed Role Team
Average Frozen Agents

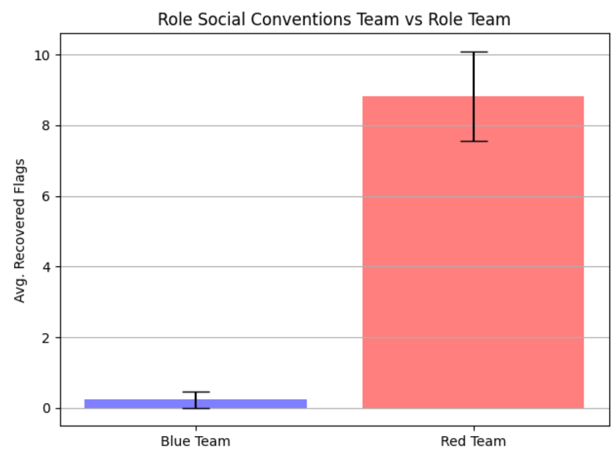
Dynamic Role with Social Conventions Team vs Role Team



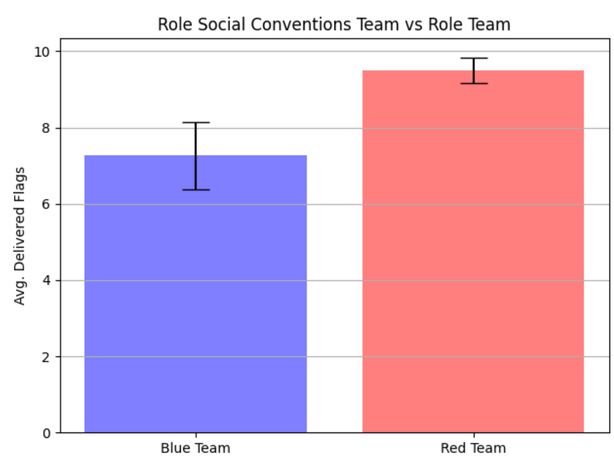
Graph 6. Average Wins Agents



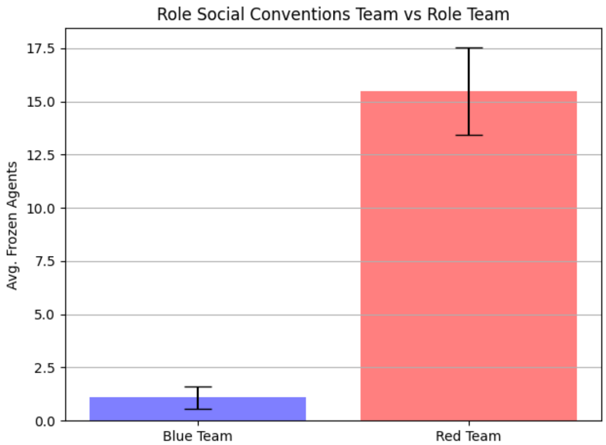
Graph 7. Average Stolen Agents



Graph 8. Average Recovered Agents

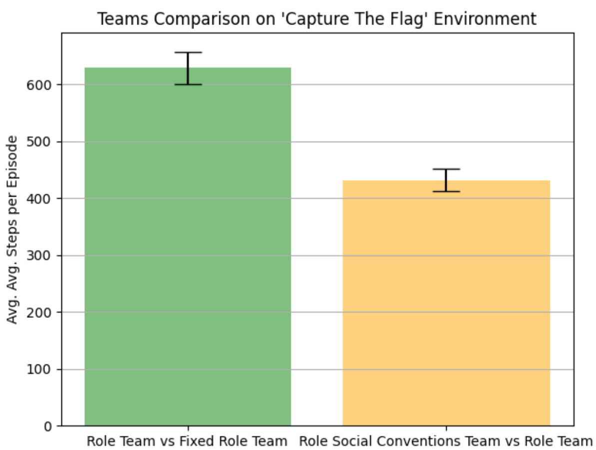


Graph 9. Average Delivered Agents



Graph 10. Average Recovered Agents

Average Steps Comparision between the 2 games (Role and Fixed Role vs Social Convention and Dynamic Role)



Graph 11. Games Comparison
Average Steps Per Episode

3.2.2 Discussion

The metrics consider the following games:

- Role Team vs Fixed Role Team
- Dynamic Role with Social Conventions Team vs Role Team

Let's consider that the first team of each game is assigned as the Blue Team and the second team is assigned as the Red Team

We choose as a base line the Fixed Role Team, since it's the simpler one.

We started by analyzing its behaviour against the Role Team. As shown in graph 1, the Role Team won nearly 60% of the games. This can be explained by the fact that the Role Team calculates the role based on the spawn of the Agent, which can give a slight advantage to that team. However, the Fixed Role Team wins sometimes because, luckly, the Agent spawn in a place where it fits the role given to it, and also the Defensive Agents sometimes reach their Defensive Area ahead of the enemy, meaning that the enemy needs to take a longer route to a different defensive area to steal flags. The difference between wins and losses is totally acceptable since the algorithms aren't very different, they just differ in the first step for the role assignment.

We can also observe similar results for the number of stolen flags, which was expected. The lower number of delivered flags by the Attackers of the Fixed Role Team can be explained by the higher number of recovered flags in the Role Team. This can happen due to the fact that, besides the Fixed Role Team being able to steal as many flags, the Defenders of the Role Team are able to steal and recover flags as well as freezing their enemies efficiently. This can happen due to inadequate spawns for the assigned role of the Fixed Role Team, as explained above.

We then move to the analysis of the Dynamic Role Team with Social Conventions against the Role Team. This time we saw no improvement. The Role Team won nearly 75% of the time. The explanation to this sits on the fact that the Dynamic Role Agent needs to compute their role at each timestep, which can delay their selection of action and arrival at the target position. Also, these agents can have dilemmas and so, it needs to change the mode type to Social Conventions. This can result in a loss of performance in relation to the Role Agent that already has a Role suitable to its objective.

Although the average stolen flags per game are slightly different, the difference regarding the delivered flags can be explained by the huge difference in both recovered flags and frozen agents. Since Social Convention actions have a predefined order set as Steal, Steal, Freeze, most of the agents are prone to perform steals in the enemy defensive area and, therefore, perform a role similar to an Attacker Role of Role Agents, resulting in a huge lack of recovered flags and frozen agents. Adding to this, the Role Team assigns one Attacker and two Defenders. Therefore, the Role Social Convention Team has a slightly large number of stolen flags.

Finally, looking at last graph 11, we can observe that the first game, Role Team vs Fixed Role Team required, in average, per episode, more 100 steps than the second game, Role Social Conventions Team vs Role Team. The first game, Role Team and Fixed Role Team was a closed one due to the fact that both architectures are extremely similar so agents require more actions to finish the game. However, for the second game, Role Social Conventions Team vs Role Team, the steps required to finish the game are much smaller since the Role Team performance is much better, as seen above.

3.2.3 Findings

Afterall the analysis, we can conclude that the Role Team is the most efficient. The algorithm is very complex but a Role is Assigned to an Agent according to its position in relation to its target, and the Agent goes ahead to its objective without any dilemmas or mode changings.

3.2.4 Future Work

The built agent architectures used for the evaluation don't take into account the coordination of the agents within a team. Therefore, we propose a new system that aims at solving this problem: Coordination Graph Agents. This system is expected to be useful in a large game with more than 3 agents per team. By decomposing the game into small sub tasks, agents could know how to interact with their teammates to improve the efficiency of the team.

ACKNOWLEDGMENTS

A special thanks to Prof. José Alberto Rodrigues Pereira Sardinha and Prof. Guilherme Sant'Anna Varela for all the availability and all the teachings that allowed us to get here.

REFERENCES

[1] Kudenko, Daniel & Kazakov, Dimitar & Alonso, E.. (2003). Machine Learning for Agents and Multi-Agent Systems. 10.4018/9781591400462.ch001.

[2] Sardinha, José. Autonomous Agents and Multi-Agent Systems DEI IST. Lecture 2 - Deliberative Agents. (2022).

[3] Sardinha, José. Autonomous Agents and Multi-Agent Systems DEI IST. Lecture 6 - Coordination Games (1), Social Conventions Agents. (2022).

[4] Sardinha, José. Autonomous Agents and Multi-Agent Systems DEI IST. Lecture 6 - Coordination Games (2), Role Agents. (2022).