Logistics modeled with agents

David Calhas
José Mota

***Abstract*—This is a project that modules autonomous agents and a multi agent system. The environment simulated is a logistics one, where companies and clients exist and interact between each other. The main purpose is to see how different metrics evolve through time and with different techniques applied, such as auctions, RL and so on. This is useful to see how markets behave in a particular way, and check the evolution of monopolies and competition between companies through time. Human preferences is also a topic that we take into consideration at the client level.**

I. Introduction

The project was developed as a component of the evaluation in a course called, Autonomous Agents and Multi-Agent Systems.

The main purpose is to module the behaviour of different agents according to multiple values of parameters (e.g., the profit that an agent wants to get in the future, etc).

A. Explanation of the world

This world is composed by two different types of agents: clients and companies.

A client is an agent that makes offers and has preferences as the world evolves, the preferences relate to the company the client has a greater affection to.

A company is an agent that makes deliveries according to the offers made by the client. each time a client proposes an offer, there is an auction for that offer, where the companies make their evaluations and the client chooses an offer.

Each company has a set of vehicles, that do the transportation. This vehicles can be divided into two types: Trucks and Buses. So that, the client can transport either goods or people. For each transportation the company shall have a profit and a cost, the money received through the client and the cost of each trip taken.

These are the two identities that make the world evolve through time, the evolution of the time is represented with a step function, that simulates the different interactions that are made through the world components and agents.

B. Problem explanation

The problem resumes in simulating a multi agent system and make this simulation as real as possible, in order to take conclusions and explore how the world evolves through time.

As it was said in A, the world evolves through steps, and one of the problems is to see what shall be done in each step (time step, to be more precise).

How should the negotiation of each offer be made? How should the transportations be done? How much time is a realistic time for the transportation?

These are all questions that we try to solve with this project and analyze its results.

C. Techniques incorporated in the problem solution

To build the system we used different techniques, with the objective to make the system realistic.

Auctions was introduced to tackle the problem of how each offer is assigned to a company. Cooperation between companies, so that they can benefit between each other and have better results, than if they were to work strictly individually, was also introduced.

Different parameters are to be used, so that the world is not static, and changes through time. Parameters such as: the risk a company is willing to take at each negotiation to make more money, versus having more probability of winning an auction, but gaining less per offer; the number of trucks or buses each company possesses, this can differentiate the companies, because a company can have more resources and consequently win future auctions, because of their availability; each client shall have preferences, so that they don't always choose the cheapest offer that comes to their "mailing box", in the real world, you, (the reader) as normal person, don't go drink the cheapest coffee in the world (or maybe you do), you a preference for a certain type of coffee, this can be because you like the service in a certain shop better than in the other, or just because of the simple taste of the coffee in a particular place; more parameters were introduced and they will be covered in more detail in the following sections.

As an extension, we also integrate learning by reinforcement in this project. We think it is that learning is in the project, because learning through the years gives results that are closer to the real world than simple algorithms. The algorithm integrated was the SARSA algorithm.

II. Background

In this sections, the techniques are explained with detail and how they are applied in the world.

A. Auctions

At each time step of the world, an auction is made. The type of auction to be implement was a first price sealed bid auction with some minor details that give some kind of enlightenment to the auction.

The auction starts by a base offer made by the client, and its from that value that the companies start bidding. This base offer is related to the distance and utility of the goods and people. Each company makes its bid, calculating it based on the cost (impact made by the distance) and the company risk.

In addition to the bid value of each company, companies that do not have the resources to proceed with the offer (do not have enough trucks or buses, depending on the offer) do not participate in the auction.

After all companies proposed the price, the client will choose the best offer, evaluating the price and preference for each company. The preference factor makes this auction not just dependable of the price, but on simulated human behavior as well.

B. Parameters

The world geography is represented as a graph, that is similar to the map of Portugal, when dividing it by different districts (18 districts in total). These representation is fixed and neither the locations or the distances between them varies, as different executions start or the world evolves (static map).

When a company is created (this is done during the setup of the world), a location is assigned as the home of the company. This location, as said in the paragraph above, is one of the 18 districts. The location is an important factor, because some districts are located in a more concentrated area (north area of Portugal), than others, like the south districts (Faro is pretty far away from the area of concentration referred).

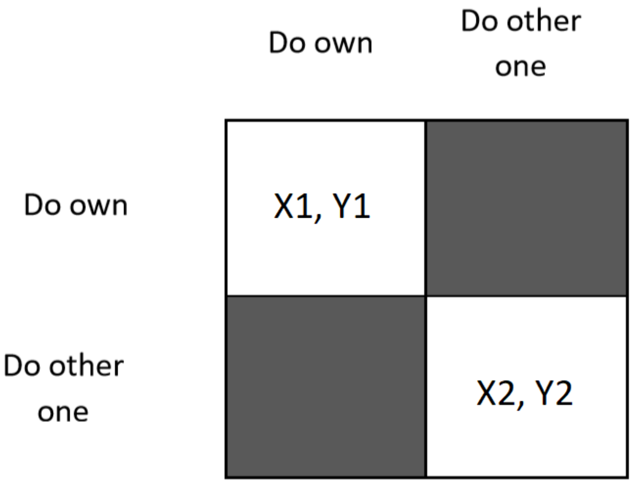
To add to the location factor, we integrated the variable that we call the risk of the company. It is called risk, because it is the risk a company is willing to take at each auction offer that is made, the bigger the risk, the smaller the probability of the offer being accepted, but the revenue of a transportation increases along with the risk from each one. On the other hand, the smaller the risk, the smaller the profit gained from each auction that is accepted, but in this case the probability of an offer being accepted is higher. This risk parameter, may define the survival or not of a company, and defines its future, its in this factor that we, further ahead, implement the learning by reinforcement.

There is given an initial budget for each company, so that they can invest on trucks and/or buses. The bigger the budget, the bigger the resources in the beginning of the world. This can be a critical parameter (if the other companies have a lower initial budget), because once a company has the preference of its clients, the more probable is the company of becoming a monopoly.

In the beginning of the set up of the world, the number of clients and companies, that are to integrate the world is defined, and the creation of them is done. This two parameters are crucial. First, the number of clients is important, not because the more clients, the more offers that are made, no that's not the case in the world that we modeled, because at each time step an offer is always made, regardless of the number of clients. The only thing, that the number of clients has an impact in the world, is in the preferences of each one of them, and this can create a market centralization. Secondly, the number of companies is important, because as the number of companies increase, the competition also does, and that can be crucial, either for the survival of the smaller companies or for the stagnation of the market (the set of companies may not have a monopoly).

C. Cooperation

The cooperation between companies was implemented in the project. We analyze interaction among a group of companies who behave strategically (make own delivery or switch the delivery with other company). We model this problem in a matrix of game theory. Two players that are represented for two companeis, and two strategies. We have just analyzed the entries of the matrices that both make their deliveries or both of them exchange. The main goal is reduce the cost of both with the exchange of deliveries. We keep a pool with deliveries already accepted. When a number of companies is reached in a poll, we verify if we can reach a nash equilibrium between companies, described above.



	Do own	Do other one
Do own	X1, Y1	■
Do other one	■	X2, Y2

Fig. 1. For the cooperation engage the following condition must be verified $(x_2 > x_1) \wedge (y_2 > y_1)$

D. Reinforcement Learning

Reinforcement Learning (RL) was introduced to see the impact it had on a company. The objective is for the agent (company) learn what value (value of the risk) to use in each situation.

The state of the company is based on the ranking of the company among the others, this ranking is made by the profit of each company. $S = 0, ..., N-1$. A company is in state 0, if it is the company with the highest balance among the companies in the world.

The cost is impacted at each state, so it will be similar to the values in the state set. $C = 0, ..., N-1$. If a company is the richest one in the world, the cost is 0, on the contrary, if the company is the poorest one, then the cost is $N-1$.

The set of actions is to maintain the value of the risk, or change it to another value. $A = Maintain, 0.1, 0.25, 0.5, 0.75, 0.9$.

There is no transition probability matrix, as the next state in the world is obtained by calling the step function that was defined for the world.

The algorithm chosen for the learning process was the SARSA algorithm, that follows the equation: $Q_{t+1}(s_t, a_t) =$

$Q_t(s_t, a_t) + \alpha * (c_t + \gamma * Q_t(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t)$. Where s_t is the state at time step t, a_t is the action at time step t, s_{t+1} is the state at time step $t+1$, a_{t+1} is the action at time step $t+1$, and Q_t and Q_{t+1} is the quality function for time step t and $t+1$, respectively.

The γ was set to 0.95 and α was set to 0.3.

E. Investment Mid-Simulation

During the simulation of the world, in other words, as the world evolves, the companies also evolve (if they have the monetary power to do so). Companies are able to buy trucks, so that they can invest in their future and have an edge on the smaller companies.

With this feature, the smaller companies have their lives more complicated, because it will be difficult for them to compete in auctions with the big sharks that have more resources than them.

III. Solution (explanation of the step function)

At first, the world is set up. A fixed number of clients and companies are created, with some parameters, such as, localization, initial budget, as well as the risk of the company. The companies use the initial budget to buy trucks and/or buses, based on a buying policy.

After the world is set, the step function is called throughout the execution, this function consists of: each company is hit by a tax; the state of the company is verified (e.g., it can be bankrupt, or it may be avoiding bankruptcy); each company has the opportunity to invest, this is if they have the means to do so; the trucks that are on the move to make a delivery, are updated by position; after this an auction is started, a client is chosen randomly to participate in the auction, as an auctioneer.

This function is defined as a step function, this means that there is a cycle and for every iteration of that cycle the step function is called. The step function basically gives a sense of time evolution to the world.

A. Reinforcement Learning

The RL part is integrated separately on the program, so there is a special version of the program that runs the SARSA algorithm.

On the setup function of the world, there is one company which ”learns” and only one company, this is because we want to see the impact of it in the world. The company executes the SARSA algorithm with 400 iterations, and after the learning is done, the Q function is stored in that respective company, and each action throughout the simulation, which consists of maintaining, increasing or decreasing the risk, is chosen based on the state of the company (the ranking of the company, based on the balance of the companies that are in the world). With this, we expect for the company to adapt on the world based on its characteristics.

IV. Execution and Simulations

In this section, we will show results of particular simulations. This will help us see and understand how the system behaves, and what particular parameters are vital for the system evolution.

We will start by analyzing the following situation: a world where there are only 5 companies and 5 clients, the companies have the same risk (which is 0.5) and are all located in the district of Lisbon. The graph below shows the execution of this setup.

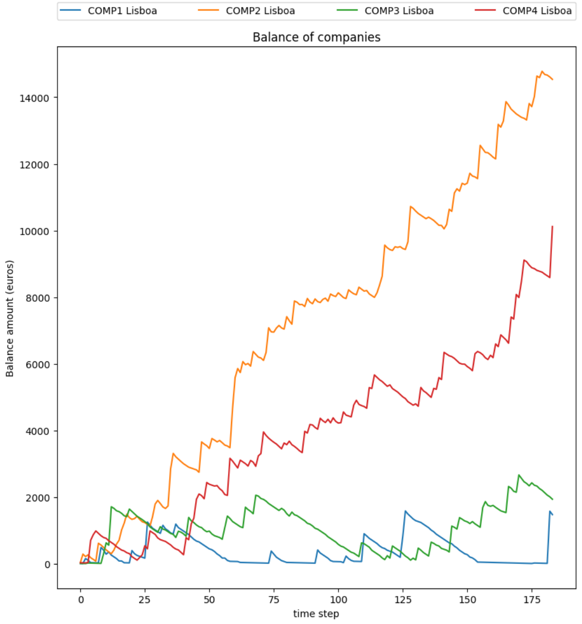


Fig. 2. This graphic shows the balance variance through time. The time of the simulation goes until time step 175 approximately.

As we can see the execution, in Fig. 2, there was one company that went bankrupt. Because there are only 4 companies plotted. This happens, because there is competition between the companies, and for a company to survive, it has to get some deals done. That was not the case for the company that went bankrupt. Because of the more parameters that are integrated in the system, such as, client preferences, and a big thing that impacts the world execution is that, if some company by luck, or any other reason, wins the first bids, that company might take control and the balance might spike from the others.

Next, lets compare the previous Fig. 2, with the following one, Fig. 9.

The two plots are really similar, both had a company that went bankrupt, comparing the risks of the companies, we might get some help to analyze the graph:

As we can see the company that went bankrupt was COMP2, and it was by no coincidence the one that had the highest Risk at 0.994, this meant that the company may not have won any auction, because of the high Risk. On the other hand, the company that had the highest balance at the end of the execution, was COMP1, and it, again by no

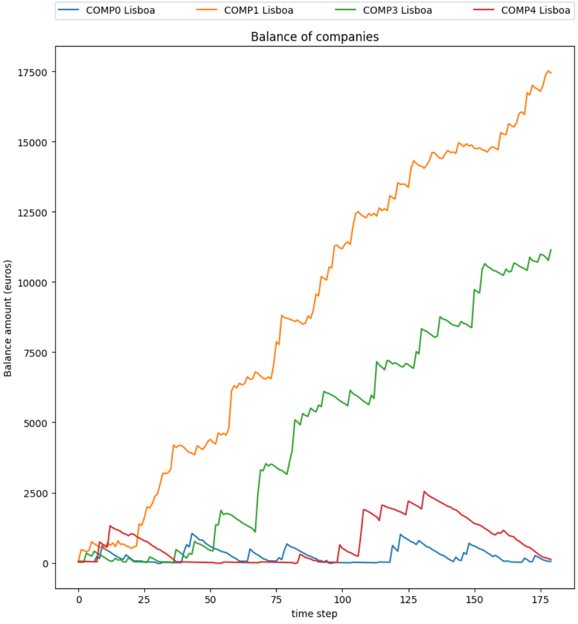


Fig. 3. This graphic shows the balance variance through time. The time of the simulation goes until time step 175 approximately.

Id	Local	Risk
COMP0	Lisbon	0.590
COMP1	Lisbon	0.118
COMP2	Lisbon	0.994
COMP3	Lisbon	0.352
COMP4	Lisbon	0.761

TABLE I
COMPANIES WITH THE RESPECTIVE, LOCAL AND RISK, PLOTTED IN FIG. 9

coincidence, was the one with the lowest Risk, at 0.118. The second company, the one plotted with the green color, had the second lowest Risk, at 0.352. So we can see a pattern here, the companies that have the highest Risk, can not win auctions, and consequently they won't survive, and enter bankruptcy. On the other hand, the ones that succeed, are the ones with lowest Risk, this is all taking into account that they are all in the same location (in this case, the location is the Lisbon district for all of them).

After this, we simulated 200 times a world, that was set up with 10 companies, none of them had learning by reinforcement; 5 clients; and the locations of the companies were random, but the risk is fixed at 0.5 for every company. With this we ran different worlds 100 times, and plotted the companies that had the highest profit in a histogram, which had the districts in the bins. Each world was simulated for 200 time steps. The goal of this plot is to see if there is any local, that brings advantage to the company.

As we can see in Fig. 4, there are districts that have a more advantageous locality. According to the plot, the two districts that had the higher amount of wins were $Santarem$ and $Viseu$, with a total of 30 and 25+ wins, out of 200 simulations.

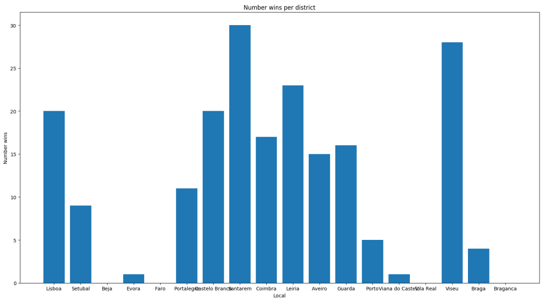


Fig. 4. Histogram showing the number of times a company located at L had the highest balance after 200 timesteps. There were 200 simulations done.

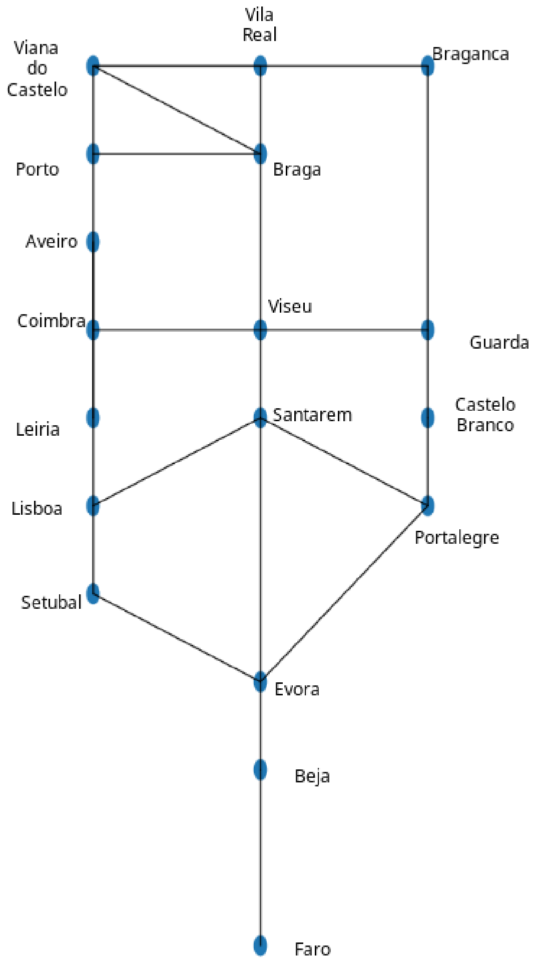


Fig. 5. This is the graph structure of the fixed map, where the companies make their transportation. Each node has a label, which corresponds to the district name, and they have connections between each other

With these results we can come to the conclusion that the districts, that have a higher amount of wins, are the more

”centralized” ones, in other words the ones which are located in a higher density area of districts.

$Faro$ and $Beja$ did not win once through the 200 simulations, this is because they are both ”isolated” districts (they are in the extreme south of Portugal) and the connection to the other ones are difficult. This is also the case of $Braganca$ and $VilaReal$, which are the extreme north of the graph (Portugal).

The things said above, can be better justified with the next image, that represents the graph of Portugal used. The nodes/districts that won 20+ times in the 200 simulations, were Lisboa, Castelo Branco, Santarem, Leiria e Viseu. These are all districts with a lot of neighbors and are located in the central part of the graph.

Apart from risk and localization, one of the basic parameters that influences the evolution of the companies on the world, is the number of competitors one has. For example, the majority of the graphs plotted were with 5-10 companies on the world, but if the number of companies increases, will there be a monopoly? It is very unlikely, due to competition being more fierce, and also because every company start from the same situation (initial budget is the same for every company in the beginning).

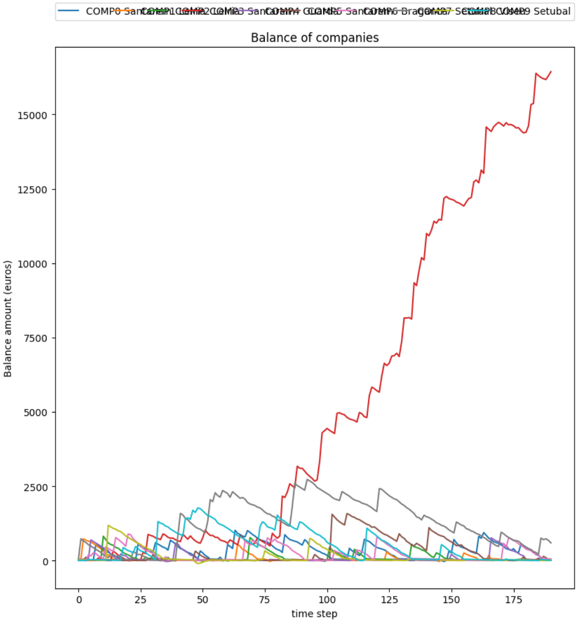


Fig. 6. Simulation with 10 companies in the world.

Fig. 6 demonstrates that one company became a monopoly after 175 steps, approximately. So, there was not a lot of competitiveness in the world from time step 100, which was the time step the company started to take off.

We simulated the world with 20 companies, and the results were really different, maybe by luck or not. The Fig. 7 shows a high level of competitiveness between the companies, no one can take off from the main herd, that is all concentrated between 0 (minimal survival value, some go below zero, but that is because of special cases of avoiding failure) and 2000, these values are the balance of each company.

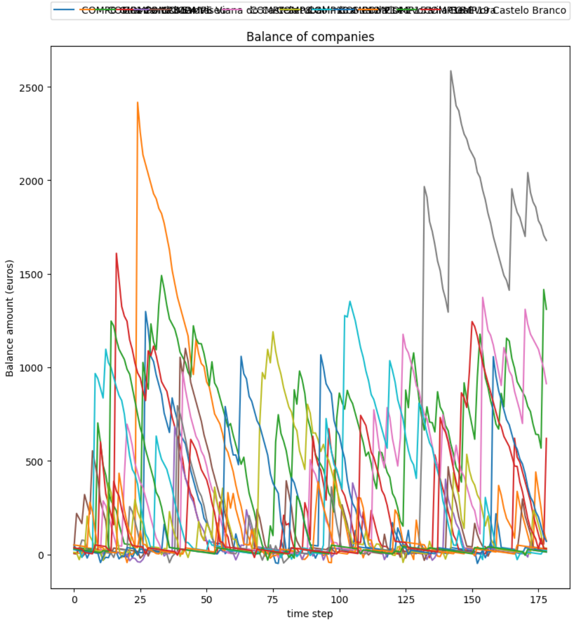


Fig. 7. Simulation with 20 companies in the world.

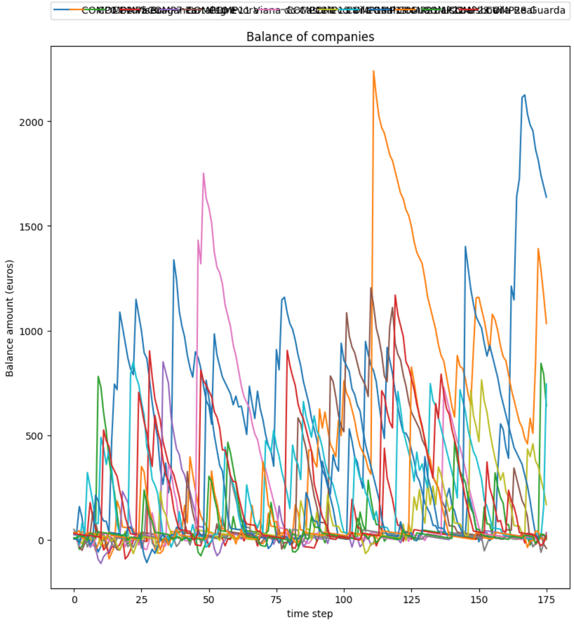


Fig. 8. Simulation with 30 companies in the world.

In Fig. 8, the behaviour is basically the same as in Fig. 7, and that proves our point, that the bigger the number of companies, the harder it is for one company to become a monopoly or evolve through time. Because of this, we think that these simulations are enough to demonstrate our point.

This competitiveness exists, because each time the step function is called, only one offer is made, and only one client makes that offer, so per step, there is only one deal between a certain company and client. This makes other companies starving, if they do not win an auction, and winning an auction is harder as the number of competitors increases.

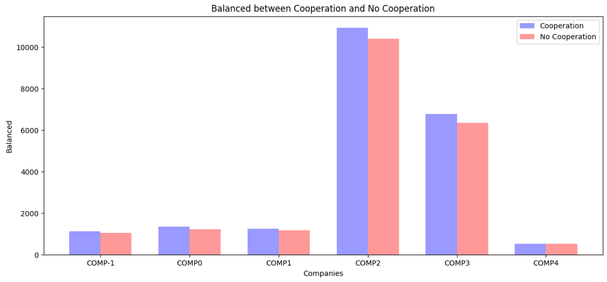


Fig. 9. This graphic shows the value raised by the company using cooperation policy or not.

Fig. 9 shows the comparison of the balances in two cases: the blue one, is when the company uses the cooperation policy; the red one, is when the company does not use that policy.

As said above, there are trades of deliveries previously accepted, and these trades are only made if there is a reduction of the cost for both companies. So, as expected, the value when the cooperation is used is higher compared with the value, when the cooperation is not used.

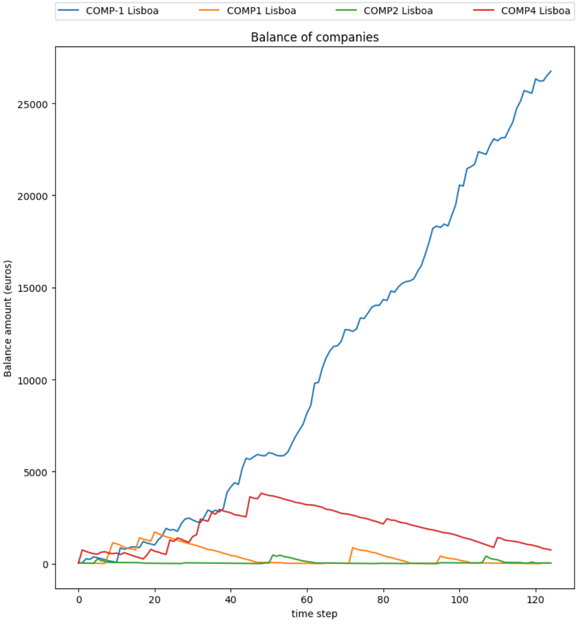


Fig. 10. Simulation of the world, with one company learning

To demonstrate how the learning by reinforcement behaves in this world, we did a simple simulation, where all the companies were located at the Lisboa district, so that locality does not become an issue, as it was already demonstrated it can become one; and the risks of the companies were fixed at 0.5, for the same reason of the locality case. The company that did the learning can change the risk throughout the simulation, that is exactly the purpose of the learning, for the company to adjust itself to the world and have a better performance when compared to the others.

In Fig 10, we can see that there is one company, with Id ”COMP-1”, which is the Id of the learning company, it becomes a monopoly at the end. In the beginning the company does not start in the best of condition, but it adapts itself as time goes by, and ultimately exceeds all the other ones. This was already expected, as the number of competitors is not high. If we change the number of companies, and make the risks and localizations of the companies random, the learning company does not become the best one anymore, this is also because of the reasons we already discussed (these parameters are very important for the evolution of the world).

To test the success rate of the RL company, we simulated 200 world simulation, with 200 time steps in each one.

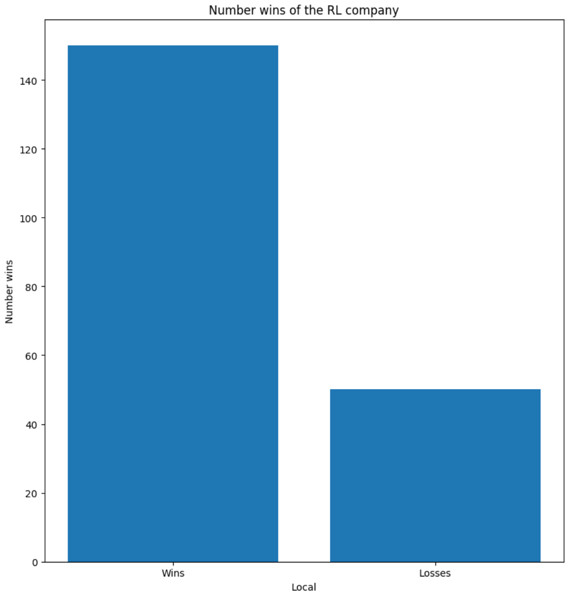


Fig. 11. Number of times the RL company, ended with the highest value of actives plus balance. This was learned with 1000 iterations in the SARSA algorithm, and the world resets after 100 iterations, so it visits every state.

In Fig. 11, we can see that the company that learns prior to the execution, has a good success rate. With around 150 times, ending with the highest value out of the companies. 200 simulations were ran to prove the viability of the values.

With this, we tried out different parameters on the learning iteration, like, increasing the number of iterations and increasing the interval in which the world resets, during the learning.

Fig. 12, shows the same type of results as Fig 11, but this time with the world reseting after 200 iterations, instead 100.

The results were not what we expected. The expectation was that the ratio of wins would increase, because in reality

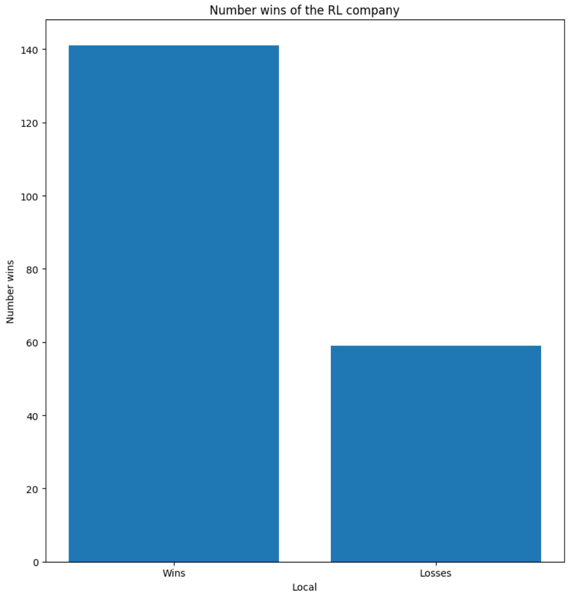


Fig. 12. Number of times the RL company, ended with the highest value of actives plus balance. This was learned with 1000 iterations in the SARSA algorithm, and the world resets after 200 iterations, so it visits every state.

200 is the number of the time steps, that the world simulates.

With this deception results, we tried another approach, in order to see if the results would get better.

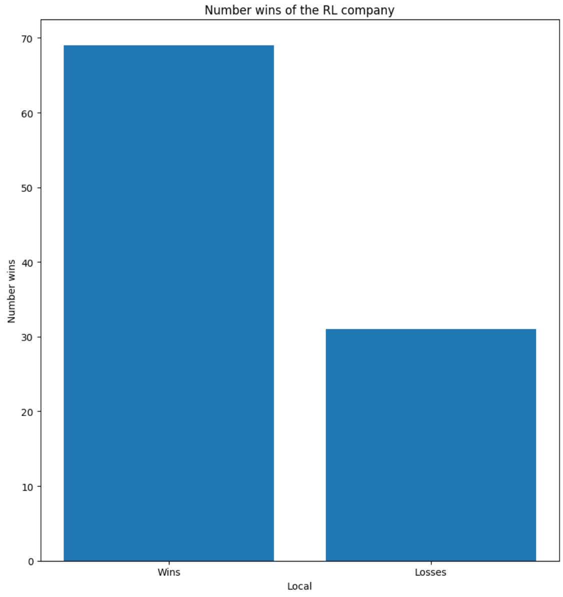


Fig. 13. Number of times the RL company, ended with the highest value of actives plus balance. This was learned with 2000 iterations in the SARSA algorithm, and the world resets after 100 iterations, so it visits every state.

In Fig. 13, the learning algorithm had 2000 iterations and the γ and α, were set to 0.99 and 0.5, respectively. The results are surprisingly worse than the simulations with 1000 iterations in the learning algorithm.

These results are not so bad, when you look at the performance of the same company, but with no prior learning.

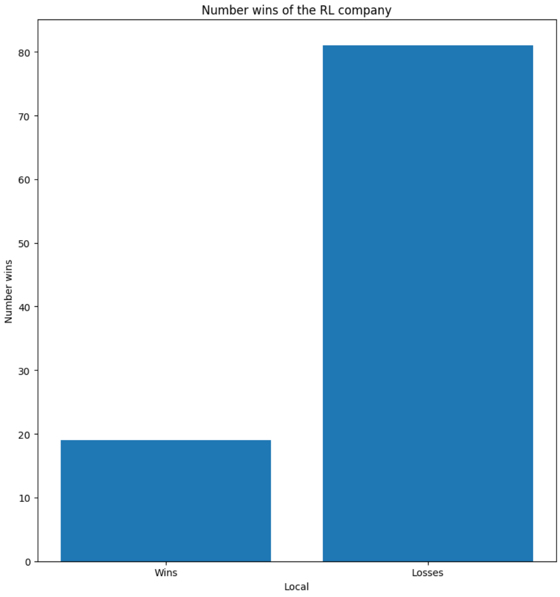


Fig. 14. Number of times the RL company, ended with the highest value of actives plus balance. There is no learning performed in this simulations

As we can see, in Fig. 14, the results are way worse than the ones shown with prior learning. We can conclude that the learning algorithm is giving an advantage to the company, although it sometimes fails. This may be due to the training, not exploring all the states and action pairs a sufficient amount of times, and because of the low amount of iterations performed.

V. Conclusion

The system built, represents a real world pretty well. The features implemented give the execution a smoother behaviour.

The parameters integrated give variability and prove that things like location can be a key factor for the success of a company.

The learning gives indeed an advantage, and makes a company perform with success in the majority of the time.

About future work, things like coordination between the agents could be implemented with more detail, and the learning could be explored with more exhaustion, in order to obtain better results.

VI. Acknowledgements

We thank Filipe Azevedo for motivating us. At times it was hard for us to continue developing our project, but he was always there for us. We thank Dominga, for being a support on our daily life at IST. We also thank Professor Rui Henriques, for giving us his insights during the development of the project, and for always being available.

References

[1] D. Ariely. *Predictably Irrational: The Hidden Forces that shape our Decisions*. Harper Collins, 1st edition, 2008.