Stock Management System

André Guerra
ist186382

André Ribeiro
ist186384

Tiago Barbosa
ist186519

1. Introduction

In recent years the development of optimization software has grown and is now tackling a wide variety of problems, managing human and non-human resources.

One of the most important and difficult problems in business management for companies which regularly trade physical assets is managing products stocks and inventory. There are two sides of the same coin which frequently are separated by a thin imperceptible line, one consists of having enough stock to maximize sales and profits and the other consists of minimizing the amount of the business' cash withhold on stock.

With this context in mind there are a lot of requirements to be met in order to fully model such a system which includes interaction between agents, each agents maximization of utilities, real-world metrics which include yearly events such as seasons and unforecastable events such as diseases and monetary inflation or deflation that may influence agents decisions and the environment itself.

In order to study this problem we propose a Multi-Agent System that captures the complex scenario that is managing a company's inventory with main focus on a single agent, the stock manager. This manager agent is able to interact with the environment by selling products to clients and negotiate with sellers, effectively altering and manipulating it's stock, money, cash-flow and sellers attitudes. The rest of this paper will be focused on clearly and soundingly explaining the model's environment, agents, how all of these interact and demonstration of results and conclusions.

2. Changes to the Agent Proposal

2.1 Multi-Agent System

In our initial Agent Proposal we suggested that the model for this problem would be designed around a single agent and it's interaction with the environment. After submitting our proposal we studied further multi-agent systems in class and afterwards we reached consensus that it would be more interesting and accurate to model the problem with a multi-agent system. This change allowed us to introduce negotiations between the manager agent and it's product sellers agents which makes the environment dynamic and reflects price fluctuations that occur in real life scenarios.

3. Agent and Environment Properties

Considering that our goal is to characterize the system there are basic building blocks that should and need to be described before diving deep into the environment and agent descriptions.

In a stock management system the most basic units are products, which can be described as what is sold, bought and drives the entire system. Products have a cost associated with them and a profit which is the difference between the cost price and the sell price. They also have an expiry date. After it has passed the products are considered to be eliminated from the system, resulting in a loss for the products' holder. A transaction between agents consists of an exchange of a certain quantity of a product, the price paid for that quantity, the seller and buyer. Our manager agent will manage a warehouse that contains an inventory, an order book and a transaction ledger. The inventory has listings of stock for each product and a record of expired listings. Even though many listings are for the same product they vary in quantity and expiry date. The Order Book maintains a daily history of every client's order and every unfulfilled order due to lack of stock in the inventory. Everyday our agent receives orders from clients and at the end of the day when there are no more orders it can choose to fulfill them in the order he pleases. Although every order expires in the same day they are made, unfulfilled orders cannot be fulfilled in the consecutive days. After this order fulfillment process it can also place orders to its sellers in order to refill its inventory. These orders are assumed to arrive before the next day starts. The transaction ledger maintains a record of every buy and sell transaction for accounting purposes.

The metrics that we chose to evaluate our manager performance are the amount of money made, which can be seen as the initial warehouse money plus the difference between the buy and sell transactions, the cash-flow or net worth of the warehouse, that is the sum of every buy and sell transaction, and the money loss, due to unfulfilled orders and expired stock in inventory. These metrics can be seen as the reward function that the manager tries to maximize and although it is not explicitly coded the behaviour that maximizes this function is implicitly emergent from the managers actions. Overall this system aims to replicate a whole sale supply chain with our agent managing the warehouse right in the middle of said supply chain.

Environment: Our agents perceive the environment as a discrete function of time, observing and interacting with it once a day everyday. Each day is characterized by a series of events that influence agents decisions and provide some cyclical but not deterministic perception of the environment. Some of the events that are cyclical and not random are the seasons of the year, festivities, such as Christmas and Easter. On the other hand, there are non-deterministic events that influence the environment, for example, diseases, euro rate increase or decrease, temperature and rain. All of the cyclical events are discrete, they either are active or inactive however the non-deterministic events are continuous and can vary in pre-defined ranges.

Agents: There are three sets of Agents in our multi-agent system: seller agents, the manager agent and client agents. It will be explained, in order of agent complexity, each agents sensors, actuators, decision making behaviour and how the agents interact.

Client Agents :

These agents sense the current environment events and actuate by placing orders to the manager agent order book. Their decision on amount of product in each order will vary based on the perception they have of the environment. This decision process aims to replicate real behaviour while maintaining some randomization in the process. Intuitively if the product in question is an ice-cream it is natural that there will be more orders with higher amounts during the summer, specially during a heat wave, and less orders during the winter, specially during a rain season.

Seller Agents :

Seller agents abstract themselves from the environment and function as whole sale retailers, where the only motivation for price variations are the negotiations that occur between them and their customers considering that usually the price of manufacturing goods they sell will not vary in a relevant percentage during the year. It is assumed that these agents have an infinite amount of stock for the product they sell and have various discounts depending on the quantity ordered. This list of discounts is communicated to the manager agent as a bulk order table. The bulk order table consists of a list of quantities and price per product in each quantity interval. The presence of a bulk order table works as an incentive for the manager agent to place fewer orders with high amounts. In a real world scenario this incentive is ***very*** accurate because it saves both the seller and the manager various shipping, managing and operational costs. Sellers compete in a one-shot sealed bid auction with their bulk order tables from which the manager agent will pick whom to order stock from. The decision process for reevaluating their bulk order tables comes after being notified if they won the auction. If a seller has won the auction he will show a greedy behavior by increasing his prices even if ever so slightly hoping that with this marginal increase he will still be the seller picked in the next auction but with higher revenues. Sellers that lost the auction will decrease their prices in hopes of making a better deal than the other sellers while still trying to maximize their profits. Sellers have a minimum offer they can make and never go below such threshold because it is assumed that doing so would turn out to be non-profitable for them. Despite of existing a minimum there exists no maximum threshold given that the goal of the sellers is to maximize their own profits, which will lead to the hyper inflation of a product cost if the manager agent only has one seller for a specific product. We believe this approach effectively mimics in simpler terms how the real world sale market behaves.

Manager Agent :

The manager agent is the most complex agent in our system considering our goal in this project was to maximize its profits and minimize its losses. This highly complex agent interacts with the environment by placing orders to its sellers and by fulfilling clients orders. For each of these two interactions our agent has respectively two decision processes which have been studied and analyzed in depth. Throughout our experiments with different decision processes and how different relations between variables in the system could lead to different levels of complexity in its behaviour we used the same baseline environment variables and agents intuitions.

Before the deployment of the manager into the system we used Supervised Learning by training a Neural Network with data from past years. This data consists of a labeled training set in which the inputs are the events described in the environment and the output is a bias that works as an intuition for our agent. In the training set the bias corresponds to the relative amount of a single product that was ordered in comparison to the whole amount of orders for all the products. When using the Neural Network in the simulation, using each days events as input, this bias will then provide our agent an intuition of the percentage of funds it should allocate to each product. This intuition allows our agent to think ahead, be proactive, prevent shortages of stock while still allocating its resources efficiently in consideration of previous experiences.

As the environment surrounding the agents is unpredictable and non-deterministic it may also happen that an unforeseeable series of events leads to a spike or lack of orders. While when a lack of orders happens there is not much room for action by the agent when there is a spike in orders it can show reactive behavior to the sudden change in the environment. When the agent is confronted with this situation where its inventory plummets it has the ability to attribute less importance to it's intuition and other characteristics and react by placing immediately a large order of the product that is missing effectively allocating more resources to such product.

Our agent also exhibits different levels of certainty and determination demonstrating a more cautious or bold behavior by placing higher or smaller orders, which can lead to wise investments that highly pay-off due to the profit margin provided by the bulk order table of the sellers and also smaller orders that lead to less loss in expired products in the inventory. Our agents risk assessment is dynamic and depending on the environment and its interactions with other agents it can choose to increase or decrease, using a stochastic gradient update, its level of boldness throughout the whole run.

All these characteristics that our manager agent possesses implicitly contribute to a maximization of its reward function, higher profit, higher cash flow and less losses.

During our analysis of agents interactions we considered the possibility of introducing *phantom* or *decoy* actions which would introduce deceiving other agents. More precisely our manager agent could deceive its sellers by everyday broadcasting the information that each agent lost the auction even though in reality it did not place any orders. Such a behavior would lead to a much faster decrease in prices and push sellers towards their minimum price. Bearing in mind that we did not encode any behavior by the sellers that would penalize in some form the manager we considered this to be an unfair and unrealistic advantage and therefore decided to leave it out of the system. We believe further work on this topic should include such behaviors and interactions.

Overall we consider our system to be robust and adapted considering we designed it to handle an arbitrary amount of seller and buyer agents and an arbitrary number of products for an arbitrary amount of time.

3.1 Agent Properties

- **Autonomy**

The manager agent makes orders by itself and can manage the stock of the warehouse and the companies funds with his decisions, although these decisions also rely on interaction with the seller agents.

- **Rationality**

The manager agent follows a function of rewards and penalties that tries to maximize our agents money and cash flow by fulfilling every order he gets from his clients while trying to minimize the amount of money loss by avoiding having items in his inventory that expire. Achieving this fine balance is t proof of our manager agent rationality.

- **Adaptivity**

In order to achieve a better result and try to maximize the reward function our manager agents decision process relies on the bias given by its Neural Network and also its risk factor that encourages him to be more careful or bold in his decisions. Both these characteristics constantly change accompanying the evolution of the environment.

- **Reactivity**

Our manager agent is able to react on sudden and abrupt changes to the environment such as a wave of orders that empty its inventory. When deciding on new orders our agent will have these changes in consideration and respond in a timely manner.

- **Proactivity**

Taking initiative in stock management is a direct consequence of our manager agent Neural Network which allows him to exhibit goal-oriented behaviour when trying to predict the next days changes in environment, resulting in a maximization of our reward function.

- **Reactivity vs Proactivity**

Our agent will always try to maximize the rewards, and with this avoid the emergency orders by doing order with lower penalties. This orders have to be done earlier in time to avoid bigger penalties, so the agent will need to have initiative to order the products before he needs them. However, the quantity of a product the agent has in stock may, sometimes, run out, forcing the agent to be reactive and make an emergency order.

- **Personality**

During the planning and architectural decision of how our manager agent should behave we charted and analyzed results with different decision procedures which sum up different personality traits such as boldness, curiosity and the ability to deceive. After comparing results we decided that the best personality with be to fulfill orders sorted in priority by the amount, minimize the number of re-stock buys to increase the benefits of higher volume orders in bulk order tables and to reassess the risk (cautiousness) of our agent every day. We also chose not to include phantom actions, although they rapidly decreased every sellers prices to the minimum, because in a more realistic scenario this would lead to sellers heavily penalizing the manager if this bluff were to be found out.

3.2 Environment properties

- **Semi-Accessible**

Our manager agent has access to all information regarding its stock, pending, fulfilled orders, money and cash flow but it does not possess any of the seller agents price tables.

- **Non-Deterministic**

Our environment is highly complex and there are many non-deterministic factors that make each run of our system different. Considering there are randomization factors that do not start with a seed running the same file to create Data Sets, will lead to a different training of our manager agent Neural Network, running the same file to create a Day Sales file will result in different Day Sales and annual events. Even without changing these files, using the same Neural Network training and the same Day Sales file will lead to different results on different runs because our seller agents product prices fluctuate non-deterministically and our agent uses these to decide on order amounts and prices which in turn will influence every aspect of the outcome in each run.

- **Static**

While the agent is deliberating the environment does not change.

- **Continuous**

Our manager agent may act on the environment an unlimited number times given that it is unpredictable how the environment will change throughout the run.

- **Non-episodic**

The agent's current action will affect a future action.

4. Decision Processes, Charting and Comparative Analysis

In this section we will discuss and compare results of our simulations. In these simulations we chose 2 products, ice-cream (gelado) and tea (chá). These products were chosen because client demand of such products throughout the year is almost symmetrical, with ice-cream being heavily ordered in the summer and tea in the winter. The expiry date we chose after buying these items were 30 and 90 days respectively. Every chart referring to the simulations has a time period of 3 years.

As discussed in previous sections our agents Neural Network requires a large data-set to provide better intuitions. The next chart represents the data-set used to train our agent for the simulations we show next.

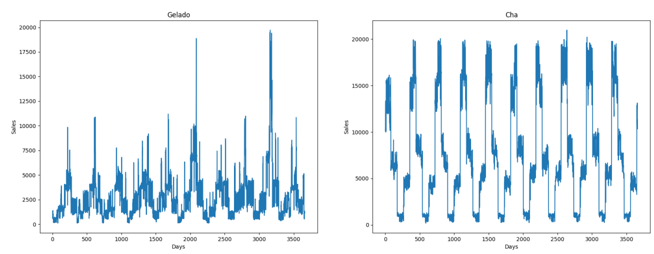

Figure 1: Quantity of product sold per day for each item, from 2010 until 2019.

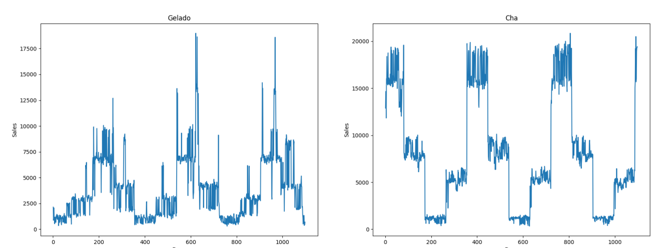

Figure 2: Product demanding from 2020 to 2022 for each item.

Figure 1 represents the data used to train the Neural Network, which is a non-deterministic distribution of events and orders in the past 10 years. Figure 2 represents the data that will be used to simulate client agents orders from now (2020) until 2022. These figures confirm that the product are demand is almost symmetrical for both products. For example, ice-cream during the summer easily reaches 7500 sales per day. On the other hand, in the winter ice-cream sales do not exceed 2500. During the winter, tea easily reaches 15000 sales per day, contrasting with summer when sales do not exceed 1000.

In order to gift our manager agent with intuition we trained and used a Neural Network as explained before. Bellow we have the learning rates of our Neural Network from the data set we've created.

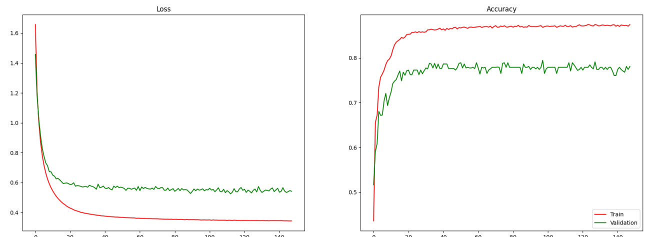

Figure 3: Neural Network training from data of sales of previous years (2010 until 2019).

From Figure 3 we can assume that the intuition bias provided by our neural network will be very accurate, with a low margin for error sometimes returning a bias slightly inferior or superior to what was expected.

After training the Neural Network we conceptualized the manager agents decision processes. As previously explained there are two different kinds of decision processes that the agent possesses, one for fulfilling client orders and another for ordering stock from sellers.

The decision processes for orders fulfillment we designed and experimented were two symmetrical approaches. One prioritized fulfilling orders in a descending fashion, with higher number of products ordered, and the other in a crescent fashion, with lower number of products ordered, tie breaking orders that have an equal number of products by the lowest client id. Although we did not experiment with emotions on our agents we believe that prioritizing a lower client id, which translates to prioritizing older customers, would lead to better results if emotions were present. Since we did not find any major differences in outcomes between these decision processes after comparing results, we decided that the agent should maintain a crescent priority throughout the rest of the executions.

The decisions processes for ordering from sellers we experimented with were different but both attempted to maximize profits. The first was BuyIfXProfit in which the manager agent always ordered a product if the prices offered by the sellers allowed him to have a fixed percentage of marginal profit on the product. The other was MinimizeBuyPrice where he tried to minimize the sellers prices, risking having low inventory but pushing the sellers prices down by ordering less frequently and in larger quantities, hoping that sellers would lower their prices in the sequent days. Although both these decision processes require planning and exhibit pro-active behaviour as explained before, the agent still exhibits reactive behaviour, if the stock is too low and the agent assesses that it will most certainly not be able to fulfill the next days orders he makes an order independent of the price as long as he can still have at least 10% profit on such an order. As explained previously, regardless of which decision process selected we also gifted the agent with a sense of boldness, a risk factor that varies between 0 and 1.

The next charts show the results for different combinations of decision processes and with the agent maintaining the same level of boldness or reassessing it throughout the run.

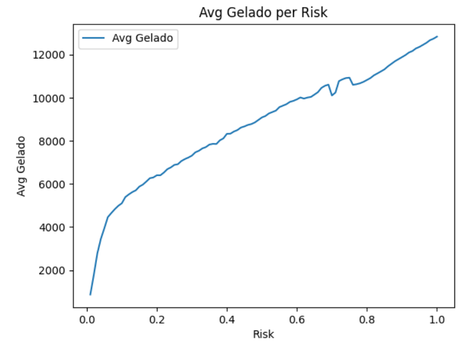

Figure 4: Average quantity of ice-cream ordered by the manager, following Crescent and BuyIfXProfit as decision processes, with variation of the risk.

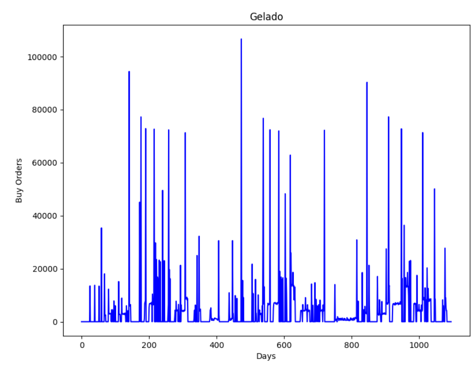

Figure 5: Quantity of ice-cream ordered per day by the manager, following Crescent and BuyIfXProfit as decision processes and risk equal to 0.5.

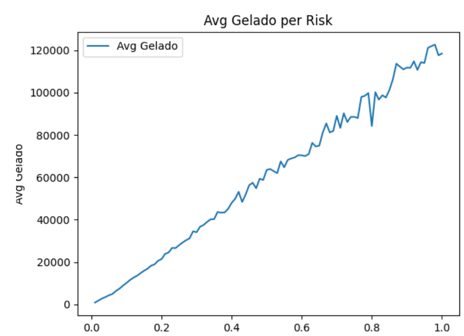

Figure 6: Average quantity of ice-cream ordered by the manager, following Crescent and MinimizeBuyPrice as decision processes, with variation of the risk.

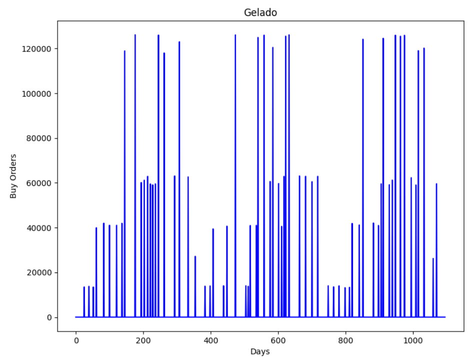

Figure 7: Quantity of ice-cream ordered per day by the manager, following Crescent and MinimizeBuyProfit as decision processes and risk equal to 0.5.

From analysis of Figure 4 and Figure 6 we arrive to the conclusion that the average size of orders done by the agent when following the MinimizeBuyPrice decision process is much higher than when following the other decision process. In Figure 5 and Figure 7 we can see that the manager buys product less often and following much more precisely the ice-cream demand when using MinimizeBuyPrice decision process (we can see the demand of ice-cream in Figure 2). Given that when following MinimizeBuyPrice the size of orders were much higher we expected a higher loss in expired stock. However, this was not the case regardless of the risk, the increase in loss from MinimizeBuyPrice process was slower, as we can see in Figure 8 and Figure 9. We can also observe that when following BuyIfXProfit the manager never loses money in Unfulfilled Orders, which can be explain by the fact that he is almost always ordering product. In contrast, this leads him to losing much more money in expired stock.

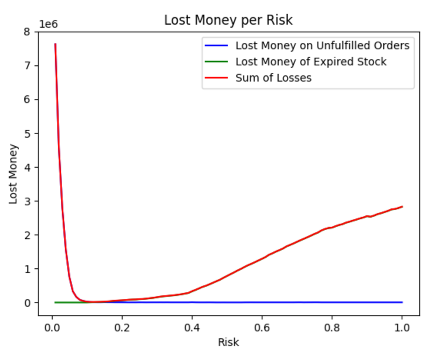

Figure 8: Lost money per risk when manger follows Crescent and BuyIfXProfit decision processes.

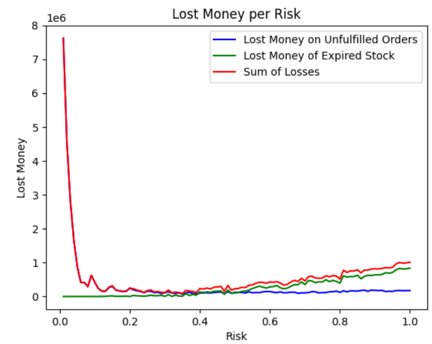

Figure 9: Lost money per risk when manager follows Crescent and MinimizeBuyPrice decision processes.

With this data, it is expected that the money the manager is going to earn is bigger when he follows the MinimizeBuyPrice decision process, as we can see bellow.

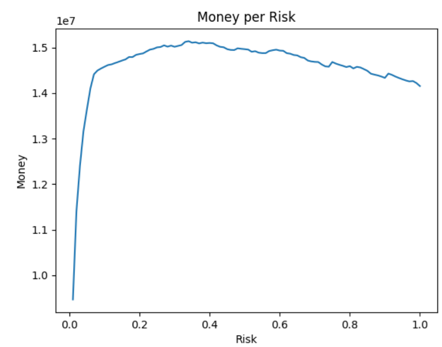

Figure 10: Final money of the warehouse per risk when manager follows Crescent and BuyIfXProfit decision processes.

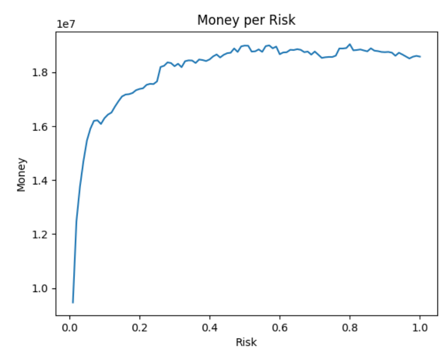

Figure 11: Final money of the warehouse per risk when manager follows Crescent and MinimizeBuyPrice decision processes.

As we've seen before, the manager buys product from sellers who are constantly changing their prices in order to try sell more product to our manager. Every day, the sellers participate in a one-shot sealed bit auction and if the agent makes an order he notifies all the other sellers that their offers were rejected. When the agent reject the offers from a seller there is a chance that the seller reduces the prices he is going to offer in the next auction to the manager. In contrast when the manager accepts the offer of a seller the seller increases his price in hopes to achieve higher profits but still have the lower prices. We can see the variation of prices per seller in the next chart.

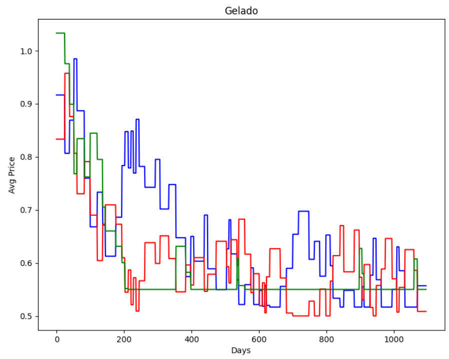

Figure 12: Variation of average price of ice-cream per day for each seller when manager follows Crescent and MinimizeBuyPrice decision processes.

The variation converges to the minimum price the seller can offer. For example, the increase of prices by the blue seller near day 200 is explained by several orders made to the blue seller in that time period. This seller offers the best offer for orders slightly above 50000 in quantity.

With these results in mind we decided to try that our manager changed his risk during execution to see if we was able to reduce all type of losses. Our solution was that the manager should increase risk (bolder) when he loses money in Unfulfilled Orders and reduces it (caution) when he loses money in Expired Stock. This stochastic gradient update is done by comparing the two types of losses and taking into account the average day sales.

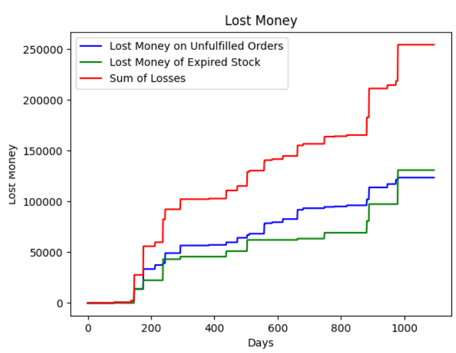

Figure 13: Lost money per day when manger follows Crescent and BuyIfXProfit decision processes and risk equal to 0.5.

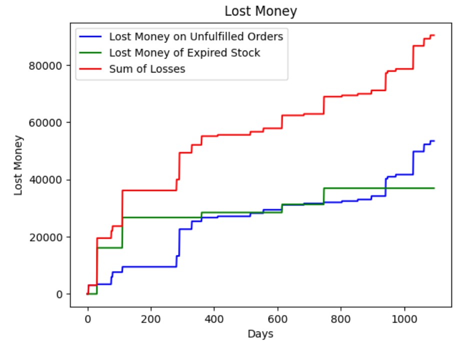

Figure 14: Lost money per day when manger follows Crescent and BuyIfXProfit decision processes and dynamic risk.

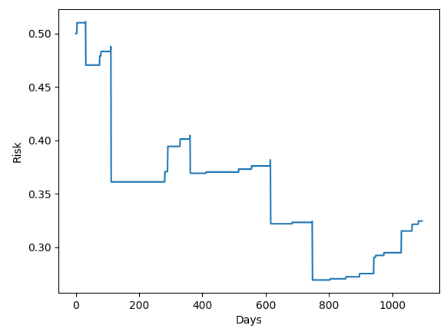

Figure 15: Risk variation per day when manger follows Crescent and BuyIfXProfit decision processes and dynamic risk.

From the charts above we can see that the loss is heavily reduced with dynamic risk, from approximately 250000 to 85000, and that in average the risk variation tends to converge to the lowest loss obeserved in Figure 9, which occurs when risk is near 0,35.

5. Results

From the results presented in the previous section the first thing we realize about our manager is that due to the training of the Neural Network with data from previous years he was able to intuitively predict quite well the product demand for every combination of events that might occur. This intuition soundingly proved that the agent was able to act pro-actively and try to predict demands in certain time frames. Since our sellers offer lower prices for higher amounts it is demonstrated that a decision process that tries to reduce orders in order to place fewer orders with larger amounts will make more profit when selling the products to the clients, this is exactly what we can learn from the differences between the BuyIfXProfit and MinimizeBuyPrice decision processes. The experiments also demonstrated that the manager should not order product with maximum risk. More risk may imply buying a larger amount of a product with a lower price but the money he earns from selling that stock is not enough to cover the money that is lost in expired stock. Our sellers behaviours demonstrated that since they are in a sealed bid auction with other sellers they tend to reduce their price offers to the lowest price they can so that they can increase their chances of getting a order from the manager. In our final test it was proved that when stochastically updating the risk taking into consideration the different types of losses our agent was able to heavily reduce the losses and the average of risk was near the static risk where we had minimum loss in Figure 9.

6. Conclusion

During this paper we explained our approaches to the problem that we introduced, clearly defining goals, problems, agents within the system, the environment on which these agents act and how they sense and interact with it. After researching extensively, each of these topics and comparing results we believe that the problem was indeed studied to in-depth to a level of detail that allows us to affirm with a great level of confidence that our manager agent would succeed in a real world scenario where the environment is more complex and uncertain.

7. Further work

With more time to further model the system and analyze interaction between agents there are some recommendations that we believe are the obvious next steps. Fully adding deceiving behaviours and penalties, such as the phantom actions from the manager agent, emotional features, clients placing higher orders with higher fulfillment rates and levels of satisfaction as well as depreciation of value due to the managers public image degradation if high levels of products in the inventory go to waste.