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 to 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 is withhold on stock.

Although it is part of our knowledge that this problem is continuously addressed by many of the biggest corporations on the planet, and that these companies probably have better solutions to this problem than ours we still consider this problem to be very interesting and worth deepening our understanding of it.

2. Agent Proposal

While investigating this complex problem we arise to the conclusion that we also need a complex agent to solve this problem.

The goal of our agent is to manage the stock of a company, being capable to answer to all of the clients demands but at the same time not wasting resources on buying an unnecessary amount of stock. The agent will build he's beliefs for each days demand, using reinforcement learning, through a data set consisting of observations of demand for that given day on previous years (experience), meaning that our agent will consume a lot of previous data in order to try and establish a pattern among randomness. At the end of each time period, the agent will check the current stock, in-flows and out-flows, and for each product decide if is necessary to order some quantity based on the beliefs he has for the next time period demand and the pending orders.

This agent is completely autonomous and needs only sensors that give him information about requests, stock and cash-flow, which in a practical point-of-view would already be integrated into the company's Point-Of-Sale system. Since we do not have any 'real' warehouse to manage feeding this information will be replicated by us in test models we establish.

3. Agent and Environment Properties

3.1 Environment properties

- **Accessible**
    - The agent has access to all information regarding the stock, pending and fulfilled orders and cash flow.
- **Deterministic**
    - The environment is deterministic because we assume that each order that the agents makes will always arrive at the supposed day and with correct number of products (if there is time after doing all the prerequisites listed here we might try to include concepts such as faulty deliveries, a random number of stock being spoiled before it's expiry date and even robberies, changing the environment to non-deterministic).
- **Static**
    - While the agent is deliberating the environment doesn't change. (if there is time after doing all the prerequisites listed here, we might try to include dynamism on the environment)

- **Continuous**
  - The agent can have a unlimited number of actions on the environment given that we can't predict how much he will need to order for each product at a given time.
- **Non-episodic**
  - The agent's current action will affect a future action.

The agent will have available three types of orders. If the stock is too low and he believes that the next day will have high demand, the agent can submit an order for next day but will get a higher delivery fee for the products arrive that quickly. The second type, is the option to make an order for the beginning of the next week and the delivery fee will be lower than the first type. Finally the agent can make a large scale order that arrives only one month later, but will pay no delivery fee for making that order thus obtaining a bigger profit (reward).

3.2 Agent Properties

- **Autonomy**
  - The agent makes orders by itself and can manage the stock of the warehouse and the companies funds.
- **Rationality**
  - The agent follows a function of rewards and penalties with the goal of avoiding to run out of a product while avoiding having products in stock that are damaged by being in stock longer than they could and trying to maximize the profit he can get from an order.
- **Adaptivity**
  - The agent is able to learn the quantity needed in stock for a product for every day of the year, considering for this the data collected from that day in the previous years.
- **Reactivity**
  - If a product in our warehouse runs out, our agent will be able to make an emergency order to restore the stock and ensure that the product is available.
- **Proactivity**
  - The agent can order a product anytime he wants if he thinks that ordering it will maximize the rewards and prevent a product run out. With this the agent is able to prevent penalties generated, for example, when he makes an emergency order.
- **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**
  - We want to make several agents to test in the warehouse, by changing the way they react to a run out of a product, and how they are going to take risks and place orders on a larger scale that can make bigger profits, but if the products of that order stay in the warehouse for a long time they can become an unnecessary expense to the warehouse for being spoiled.