Final Project Design for EECS 472

Dajun Yue

# Abstract

In this final project, I want to investigate the role of inventory management in the supply chain operation under external demand uncertainty from the dynamic market. I first develop two single-stage inventory models, which reveals how different inventory policies affect the economic performance of the supply chain. Later, I will develop multi-echelon supply chain networks, involving numerous interacting agents. In the end, a HubNet version of the multi-echelon supply chain model will be established, which allows human clients to participate in the virtual supply chain operation.

# Project description:

In the final project, I will explore the inventory dynamics in the supply chain. Demands from external market can be highly uncertain nowadays. Storing additional safety inventories is a straightforward approach to handle the demand uncertainties. But excess inventory can leads to unnecessary utility and maintenance cost in supply chain management. Therefore, a significant question is ask is: what is the optimal inventory level for each stock keeping unit (SKU), meanwhile the supply chain could provide satisfactory services to the customers. To explore the answer to this problem, we first build two single stage inventory models, which correspond to the continuous review policy and periodic review policy, respectively. In these two models, we fully developed the inventory policies (e.g., when to order, what quantity to order, how often to check the inventory level). Three types of demand patterns are considered in the models, namely deterministic, Poisson distribution, and truncated normal distribution. Followed by the single stages inventory models, multi-echelon models will be developed. The multi-echelon models would involve a number of agents playing different roles in the supply chain, including suppliers, distributors, and retailers. Agents place replenishment orders from upstream echelons, and provide products to downstream echelons, or to external customers. We want to see how the agents, given specific inventory operation rules, would react to the demand uncertainty from the external market. In the end, a HubNet version of the multi-echelon supply chain will be established, which allows human clients to participate in the supply chain operation. Since human clients may not always stick to specific rules, it would be interesting to see how the inventory dynamics evolve over time, as well as the customer performance of the supply chain network.

# Things can be learned:

In fact, lots of factors involved in the supply chain management can be explored using the agent-based modeling framework, including forecasting, vehicle fleet scheduling, inventory management, etc. However, in order not to make things too complicated, I will focus on the study of inventory management under uncertain demands.

As mentioned above, the most critical question to all the companies is: what’s the best inventory policy and inventory level? By using behavior space, we can find out the optimal set of parameters for single-stage inventory problems. In fact, for single-stage inventory problems, analytical solutions can be rigorous developed using probabilistic and statistical methods. However, when multiple agents and more complex supply/demand relationships are involved, things become intractable using analytical approaches.

One thing fascinating in agent-based modeling is that, it allows us to develop from “bottom up”. We can specify same or different properties for each agent, and design communication rules between agents in a very simple way. By running the simulation, macroscopic behaviors may emerge, sometimes even in unexpected patterns.

Analytical methods are prohibitive for studying supply chains with complex networks. But agent-based simulation would enable us to explore the dynamics of complex systems. We can test out the inventory solutions returned by approximate mathematical optimization methods in agent-based modeling. We can also explore the inventory dynamics in the supply chain by varying various parameters. The famous root beer model in the model reveals the impact of a supply delay to the performance of the sequential chain, whereas more phenomena may be observed in a complex network.

One famous phenomenon is the “bullwhip effect”, which states that the orders placed by inventory managers are usually high in one period and low in another; and people find out that the fluctuation would become larger and larger as we observe from the downstream to upstream. An example is the demand for baby diapers. Since the baby diaper is not a seasonal product, the customer demands at retailer markets are usually steady. However, surprisingly, as we observe from downstream to upstream, through wholesalers and distributors, the demand seen at the diaper manufacturers are highly fluctuating. In the literature, researchers mentioned several factors that may result in this phenomenon, and it would be interesting to see if the bullwhip effect can be observed from our agent-based model.

# Implementation:

## Model 1: Single Stage (r,Q) Inventory Policy

This model tries to simulation the inventory dynamics of a SKU operating (r,Q) policy under the uncertain demand. Detailed explanation of the background of the model would be explained later.

### Agents:

There is only one agent at the center of the world, representing the SKU. The size of the circle stands for the on-hand inventory level at the SKU.

The SKU has the following properties:

1) ON-HAND-STOCK: stands for the on-hand inventory level, which is the amount of physical inventory present in the SKU.

2) BACK-ORDER: stands for the amount of inventory that is back-ordered. If the on-hand inventory is not enough to satisfy the demand, the unsatisfied demand is counted as back-orders. Once the replenishment arrives, back-order inventory needs to be filled first. In other words, on-hand inventory level will be zero until back-orders are eliminated.

3) INVENTORY-POSITION: equals to the on-hand inventory level plus the inventory in-transit. Inventory in-transit refers to the replenishment orders that are already placed, but have not arrived yet. Another name for inventory in-transit is “pipeline inventory”.

4) PIPELINE: is a list with the number of elements equal to LEAD-TIME. Once an order is placed, the order quantity will be added as the last element in the pipeline list. If no order posted, zero is put as the last element. Every tick, the list would evolve. New element will be added at the end, and the first element will be removed from the list, which means the order arrives at the SKU.

5) ALPHA-SERVICE-LEVEL and BETA-SERVICE-LEVEL: These two metrics are often used to measure the customer service level of the supply chain. Alpha service level measures the number of times that the demands are satisfied over the total number of demands. In contrast, beta service level measures the quantity of the demands that are satisfied over the total quantity of demand. Therefore, alpha service level measures frequency while beta service level measures the quantity.

6) R: stands for re-order point. Under the (r,Q) policy, when the inventory position is below this level, a replenishment order is placed.

7) Q: stands for the order quantity. Under the (r,Q) policy, the quantity of the replenishment order is a constant, which equals to Q.

### Interactions:

Since there is only one agent (SKU) in the model, there is no interaction with other agents in this single-stage inventory model.

### Setup:

During the setup, an agent is created and placed at the center of the world. The initial inventory level is specified. Elements on the interface are discussed as follows.

RE-ORDER-POINT: lets us specify the re-order point R of the SKU.

ORDER-QUANTITY: lets us specify the order quantity Q of the SKU.

LEAD-TIME: lets us specify the waiting time for the replenishment order to arrive, In this model, once we place a replenishment order, the order won’t arrive immediately but takes some time, which is called “lead time”.

We assume in this model that an uncertain demand would occur in every tick. Three kinds of distribution are considered in this work.

1) Deterministic demand: This is the simplest case, where demand in every tick is constant. In other words, there is no uncertainty. The level of the constant demand can be specified using the MEAN-FOR-DETERMINISTIC slider.

2) Poisson distribution: As the name indicates, this type of demand follows the Poisson distribution. We can specify the mean of the demand by adjusting the MEAN-FOR-POISSON slider.

3) Truncated normal distribution: This type of demand is related to the normal distribution, but slightly different. A normal distribution allows the demand to range from negative infinity to positive infinity. But in practice, that is not true. The truncated normal distribution has a lower bound and an upper bound for the demand. The distribution can be generated in NetLogo following the algorithm:

We first generate a random number which follows the normal distribution (using “random-normal” primitive), with the mean and standard deviation (std.) adjustable via MEAN-FOR-NORMAL and STD-FOR-NORMAL sliders.

If the random number is between the lower bound (LOWER-BOUND-FOR-NORMAL) and upper bound (UPPER-BOUND-FOR-NORMAL), we report as it is.

If the random number is less than the lower bound or greater than the upper bound, we then randomly pick a value between the lower bound and upper bound with equal chance.

### Rules

At every tick, the agent (SKU) will perform the following tasks.

1) Update the inventory policy: Users are allowed to adjust the re-order point and order quantity during the simulation.

2) Receive replenishment order: Take the first element in the PIPELINE list and put the corresponding amount of materials into the inventory. If there is back-order, the back-orders are filled first. The rest is then added to the on-hand stock.

3) Sell: The demand occurs in every tick. If the on-hand inventory level is enough to satisfy the demand, we simply reduce the on-hand stock level by this quantity. If the demand is larger than the on-hand inventory, the unsatisfied demand would be counted as back-orders, meanwhile the on-hand stock level is set to zero.

4) Replenish: At the end of the day, we will check the inventory position (on-hand + in-transit). If the inventory position is below the re-order point R, we place an order with the quantity of Q. If not, do nothing.

5) Calculate the service levels and cost. The methods for calculating the services levels are just as their definitions. The costs considered in this model include: FIXED-ORDER-COST (every time an order is placed, this constant cost occurs), VARIABLE-ORDER-COST (the purchasing cost proportional to the quantity of the order), INVENTORY-HOLDING-COST (the cost for utility and maintenance for keeping the inventory, which is proportional to the on-hand stock level), and BACKORDER-PENALTY (backorders are penalized, proportional to the back-order level).

## Model 2: Single Stage Periodic Review Inventory Policy

This model is similar in many aspects to the previous model, but employs a different kind of inventory policy. To avoid duplication, we only introduce the added or modified features in the periodic review model.

### Agent:

In this model, instead of using re-order point and order quantity, it uses REVIEW-PERIOD and BS (base-stock level) as inventory control parameters.

Another feature is the addition of CHECKLIST, which is a list with the number of elements equal to REVIEW-PERIODS. Only one of the elements equals to 1, whereas the rest elements all equal to 0. At every tick, the first element is moved to the last, while keeping the number of elements constant.

### Rules:

The major difference to the (r,Q) policy is the rule for replenishment. Under the periodic review policy, the replenishment mechanism is as follows.

In the (r,Q) policy, the inventory level is checked at every tick. However, in this model the inventory position is checked every REVIEW-PERIOD. To implement this, we check the first element of the CHECKLIST, if it equals to 1, we will go to check the inventory position. If not, we will simply skip the replenishment.

If we are to check the inventory position, then we compare the inventory position with the base-stock level. If the inventory position is below the base-stock level, we place a replenishment order. The quantity of the order is equal to the difference between the base-stock level and the inventory position. Therefore, unlike in the (r,Q) policy, the order quantity is not constant under the periodic review policy.

## Model 3: Sequential Chain

Model 1 and Model 2 study the inventory dynamics of the single stage SKU. Starting from Model 3, we extend to multi-echelon supply chains. Model 3 tends to reveal the inventory dynamics of the sequential supply chain, which is the simplest multi-echelon supply chain.

### Agent:

In this model, we assume that the turtles (SKUs) are all using base-stock policy, which is the same as in Model 2. As shown in the figure, SKUs have a property called “ROLES” which can be one of “SUPPLIER”, “DISTRIBUTOR”, and “RETAILER”. However, different from previous models we create “links” between the SKUs.

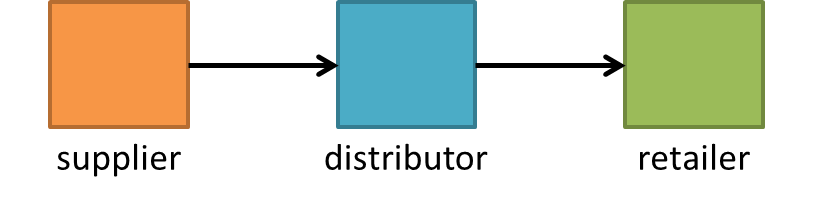
All the SKUs except for the first stage in the chain, create a DEMAND-LINK to its immediate upstream SKU. The demand links have a property called ORDERS-PLACED. On the other hand, all SKUs except for the last stage in the chain, create a SUPPLY-LINK to its immediate downstream SKU. The supply links have a list called ORDERS-FILLED.

### Rules:

At every tick, the SKUs still perform the similar tasks as in Model 2.

External demand (with three choices of distribution) is imposed on the last stage of the chain (usually RETAILER). For other SKUs, they place the order to the DEMAND-LINK to their immediate upstream SKU.

When fulfilling orders, we assume that the first stage of the chain (usually SUPPLIER) has infinite supply, which means no matter how large the demand is, it can satisfy. The SKUs compare the demand from downstream and their on-hand stock. If they have plenty of on-hand stock, they place the ordered quantity into the SUPPLY-LINK to the downstream node, by “lput” a number in the list of ORDERS-FILLED. If the SKUs don’t have enough on-hand stock, they will put whatever they have on-hand into the SUPPLY-LINK. The unsatisfied demand would be counted as backorders.



## Model 4: General Network Supply Chain

With the knowledge and modeling experience of the sequential supply chain, we will extend our model to general supply chain networks, which involves multiple agents in the same echelon. Two critical issues in general supply chain network are: How much to order and From which upstream unit(s)?



Status: Still working on it.

## Model 5: HubNet Implementation of the General Network Supply Chain

Based on Model 4, we will develop a HubNet implementation of the general supply chain, to allow human clients play as inventory managers. It is believed that human strategies may be different from the preset inventory policies, thus generating surprising and interesting results and macroscopic patterns.

Status: Haven’t started yet.