8-2021

The Food Truck Problem, Supply Chains and Extensions of the Newsvendor Problem

Dennis Quayesam
East Tennessee State University

The Food Truck Problem, Supply Chain and Extensions of the Newsvendor Problem

---

A thesis

presented to

the faculty of the Department of Mathematics and Statistics

East Tennessee State University

In partial fulfillment

of the requirements for the degree

Master of Science in Mathematical Sciences

---

by

Dennis Lartey Quayesam

May 2021

Jeff Knisley, Ph.D., Chair

Michele Joyner, Ph.D.

Robert Price, Ph.D.

Keywords: newsvendor models, stochastic, food truck, simulation, Simpy

ABSTRACT

The Food Truck Problem, Supply Chain and Extensions of the Newsvendor Problem

by

Dennis Lartey Quayesam

Inventory control is important to ensuring sufficient quantities of items are available to meet demands of customers. The Newsvendor problem is a model used in Operations Research to determine optimal inventory levels for fulfilling future demands. Our study extends the newsvendor problem to a food truck problem. We used simulation to show that the food truck does not reduce to a newsvendor problem if demand depends on exogenous factors such temperature, time etc. We formulate the food truck problem as a multi-product multi-period linear program and found the dual for a single item. We use Discrete Event Simulation to solve the stochastic version of the dual and found the optimal order to maximize the food vendors profit.

ACKNOWLEDGMENTS

I would like to express my heartfelt gratitude to my advisor Dr. Jeff Knisley. I would like to thank him for his dedication, inspiration, patience and guidance through out my study. I would also like to thank Dr. Michele Joyner and Dr. Robert Price for accepting to be members of my thesis committee and their valuable suggestions. I would also like to express my appreciation to all faculty members in the Department of Mathematics and Statistics at ETSU for sharing their knowledge.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION AND BACKGROUND

Inventory management is a key contributing factor to the profitability and sustainability of modern business. Inventory control ensures sufficient quantity of items are available to meet future demands of customers. The newsvendor problem is one of the classical problems in inventory management [35]. A newsvendor purchases newspapers in bulk each morning with the intention of selling them during the day. On some days, the newsvendor sells the entire order of papers, while on other days, not all are sold. The newsvendor must decide how many papers to buy to meet future demands [11]. There is a cost per newspaper remaining if he orders too much or a cost per missed sale if he orders too few. A generic setting of this problem is when a decision maker has a single opportunity to decide how much of a single item to order for a single selling period [33]. The newsvendor model is used as a decision tool for the difficult task of filling inventories given uncertain demand [11] and is considered one of the most celebrated models in Operations Research [21].

In nearly all inventory stocking situations, the demand of the customers varies from day to day due to many factors. Some of the factors are price, taste, the number and price of substitute goods, the number and price of complementary goods, income, distribution of income and expectation of future price changes [40]. The newspaper vendor may, for example, have high demands on days when the paper's headlines are interesting and low demands on days when it snows or rains. Ordering too many papers on a day when demand is low incurs a loss. Papers unsold at the end of the day are salvaged at a revenue per item below the cost. Subsequently, ordering too few when future demand is high incurs loss because the newsvendor has to turn away a customer for not having a paper to sell. The newsvendor's objective is the best (optimal) decision he must make to order the appropriate number of newspapers in a given day to maximize profit.

This is a decision many businesses face. In the late 1800's, as financial transactions began to become increasingly complex, the actions of money managers, bankers, and related others became increasingly mysterious. The newsvendor problem was ironically developed in the financial arena by experts to explain to clients how maximizing profit is equivalent to minimizing cost [15].

The newsvendor problem is applied to many real situations across different industries as a decision tool. The model applies to seasonal or perishable products such as fashion apparel, Valentine's day flowers, Christmas trees and cards where retailers must order before a selling season to meet demands [20]. It is used also in the manufacturing industry to predict the optimal quantity of an item to produce to meet future demands of customers [32]. The retail of perishable goods such as bread [16], diary foods and agricultural produce are also framed as newsvendor problems. According to Arikan [2], it also has wide application in service industries such as airlines, hotel rooms, trains and theater seats when the key decision is capacity. In health, it is used to propose a vaccination policy to lower attack rate and vaccine production cost [46]. It can also be used to find the optimal order quantity of immunization vaccines with a short expiration date in a single period. It is used in financial planning to allocate funds, in raising capital for firms on the stock exchange, in insurance and many more [30, 43].

1.1 Classic Single Product Newsvendor Problems

The classical Newsvendor problems deal with a situation faced by a retailer who has a single procurement opportunity to order a single product to be sold to customers during a single sale period. The unit of profit lost because a retailer ordered too few resulting in stockout of inventory to meet customers demand is known as underage cost. The cost lost because a retailer ordered too much exceeding actual customer demand (resulting in excess un-used inventory) is known as overage cost.

We introduce the newsvendor problem as an inventory model described in [39]. Given the formulation of the model, we introduce the following notation:

- Order Quantity (Q): This is the decision variable of a certain product to satisfy demand. We also use x to denote order quantity in some contexts.
- Demand (D): A variable that is defined by the demand distribution.
- Unit Cost (c): This is the unit cost of producing or purchasing each unit.
- Unit Price (p): This is the unit price at which a product is sold during the sale period.
- Holding Cost (h): This is the holding cost per unit at the end of the sale period.
- Backorder Cost (b): This is the backorder cost per unit of unmet demand.

Suppose that a retailer has to decide an order quantity Q of a certain product to satisfy demand D. If the demand is more than the order quantity, then a backorder cost b is incurred. If the demand is less the order quantity, then a holding cost h is incurred and similar to [39], the total cost is

$$C(Q, D) = cQ + b[D - Q]^+ + h[Q - D]^+$$

where $[\cdot]^+ = \max[\cdot, 0]$. The objective is to minimize the total cost function, which can also be written in the form

$$C(Q, D) = cQ + \begin{cases} h(Q - D) \text{ if } D < Q \\ \\ b(D - Q) \text{ if } D > Q \end{cases}$$

If the demand D is deterministic and if Q^* denotes the argmin of the cost for fixed D, then $Q^* = D$ is the optimal order as as shown in Figure 1.

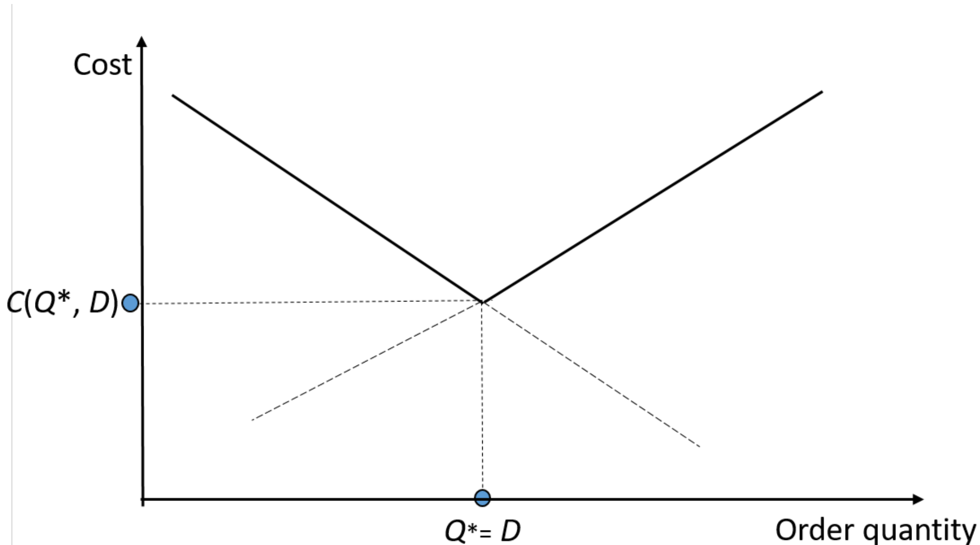

Figure 1: Order quantity equal to demand

In reality, the retailer places an order before the beginning of the sale period and the demand D is uncertain or random. If historical sales show trends in demands, then the demand D may be forecast by forecasting techniques such as moving average, exponential smoothing or time series analysis. In some cases, the demand is assumed to have a specific distribution whose parameters can be estimated [17, 21]. However, in an experimental study, the optimal solution is not necessarily improved even when the demand distribution is assumed to be known by the subjects [6]. Demand distribution estimates in practical situations are often volatile and subject to errors [36]. Demand estimation cannot be done accurately where the history of actual demand observed is insufficient. Demand data is not available for newly introduced products to trace demand patterns [18]. Therefore, the distribution of uncertain demand may be difficult to identify and may change over time.

If demand is random and discrete with probability distribution $\Pr(D)$, then the expected cost is

$$E[C(Q,D)] = \sum_{D=0}^{\infty} C(Q,D)\Pr(D)$$

which is equivalent to

$$E[C(Q,D)] = cQ + \sum_{D=0}^{Q-1} h(Q-D)^+ \Pr(D) + \sum_{D=Q}^{\infty} b(D-Q)^+ \Pr(D)$$

According to [11], “in some situations, a discrete distribution function provides the best representation of demand, whereas in other situations a continuous distribution function works best”. The discrete random demand is often approximated to be a continuous random variable because when a discrete demand is used, the resulting expressions and optimal order quantity may become more difficult to solve analytically [20]. If $f(D)$ is the probability density function of the continuous random demand

D, then the expected cost function is

$$E[C(Q,D)] = \int_{D=-\infty}^{\infty} C(Q,D)f(D)dD$$

Since $D \geq 0$, the cost function for the continuous case is equivalent to

$$E[C(Q,D)] = cQ + \int_{D=0}^{Q} h(Q-D)^+ f(D)dD + \int_{D=Q}^{\infty} b(D-Q)^+ f(D)dD$$

The goal of the newsvendor problem is either to maximize the expected profit or to minimize the expected cost. The profit function P of a retailer who decides to order quantity Q to satisfy demand D is

$$P(Q,D) = \begin{cases} (p-c)D - c(Q-D) & \text{if } D < Q \\ \\ (p-c)Q & \text{if } D > Q \end{cases}$$

The expected profit maximization is equivalent to the expected cost minimization because the optimal order quantity Q^* is

$$Q^* = \text{argmax } P(Q,D) = \text{argmin } C(Q,D)$$

Therefore, without loss of generality, we use loss minimization as the objective function throughout this thesis.

1.2 Extensions of Newsvendor Problems

In the larger context of operations research, extensions of the newsvendor problem have been developed to model a large variety of important real life problems. The multi-product newsvendor model is one of the extensions of the classical newsvendor problem and was first studied by Hadley and Whitin [19]. The multi-product newsvendor problem is framed as a retailer who has to order inventories of several products at the beginning of a sale period and sell each product at a distinct fixed price [12]. Unsold products after the sale period become worthless and are therefore discounted or discarded [41]. The retailer has to determine the optimal order quantities for all the different products that will minimize the expected cost.

Following the same notations in the classical newsvendor problem, let n be the total number of product such that $i = 1, 2, ..., n$. The expected cost minimization for a discrete demand of products is

$$\sum_{i=1}^{n} E[C(Q_i, D_i)] = \sum_{i=1}^{n} \left(c_i Q_i + \sum_{D_i=0}^{Q_i-1} h_i (Q_i - D_i)^+ \Pr(D_i) + \sum_{D_i=Q_i}^{\infty} b_i (D_i - Q_i)^+ \Pr(D_i) \right)$$

and the expected cost minimization for a continuous demand is

$$\sum_{i=1}^{n} E[C(Q_i, D_i)] = \sum_{i=1}^{n} \left(c_i Q_i + \int_{D_i=0}^{Q_i} h_i (Q_i - D_i)^+ f(D_i) dD_i + \int_{D_i=Q_i}^{\infty} b_i (D_i - Q_i)^+ f(D_i) dD_i \right)$$

In the multi-product newsvendor problem some studies suggest quadratic programming and a notable example can be found in [1]. A Langrangian method is also proposed in a related study [48].

Extensions to the classical newsvendor problem can be classified into 11 categories in terms of new assumptions and constraints [26]. Many seasonal products in certain cases of retail businesses are sold across multiple periods [44]. Some portion of unsold products from previous periods are stored and sold in later periods. The amount of inventory carried over into the next period is also random due to the fluctuations in demand. This type of newsvendor problem is called a multi-period problem. In a single product multi-period and multi-product multi-period problems, the retailer's goal is to determine a sequence of order quantities that minimizes the expected cost.

The multi-product newsvendor and multi-period newsvendor problems can be formulated as linear programming problems or mixed integer programming problems. A linear programming problem is a problem of minimizing or maximizing a linear function in the presence of a set of linear equality or inequality constraints [5]. The standard form of every linear program is [34]

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

where $c \in R^n$, $b \in R^m$, $A \in R^{m \times n}$ and $x \in R^n$ is the vector of n decision variables to be determined. According to Bertsimas [7], the linear programs version of the multi-product remains tractable if integrality constraints are imposed on the orders.

The primal is the original linear programming problem, but for each primal there is an associated linear programming problem which is called the dual. Every linear program has its associated dual. The dual program for the standard linear program described above is

$$\begin{aligned} \max \quad & b^T y \\ \text{s.t} \quad & A^T y \leq c \\ & y \geq 0 \end{aligned}$$

Dual programs are important in linear programming for the interpretation and implementation of sensitivity analysis [20]. The idea of duality is useful in modeling since it provides economic interpretations of solutions to linear programs [28]. Some-times management wants to know the effect on the optimal solutions if conditions cause small changes in parameters. In certain applications, getting the solution to the dual is faster compared to the solution to the primal [13]. Duality theorems have shown that the solution to the dual is equal to the solution to the primal if the optimal solution exists in both the primal and dual linear programs and the duality gap is zero.

Turning away a customer because a firm has run out of stock causes damages for any business. In this era of online sales, customers will simply go to different stores to find what they are looking for in a situation where their regular store is out of stock. These customers may switch to buying from the new store that always has what the customer needs [10]. This situation is not different from offline stores. Excess inventory is often not a problem but lost sales can be a big problem [11]. Businesses therefore want to avoid lost sales and thus will not allow the inventory to drop below what is required for that sale period. Extra inventory held to protect against stock-out is known as the safety stock. Maintaining safety stock involves re-ordering within multiple periods. The decision maker observes the state of demand and based on that, chooses the order quantity.

In the multi-period newsvendor problem, if the demands for each period are deterministic or known, then the expected profit occurs where the sum of the order quantities for each distinct period is equal to the demands of those period respectively. The optimal order quantities are complicated by one period extending to multiple periods in the sale horizon of the products since the next period ordering quantity is determined from previous period unsold products. The system operates continuously with random demands for each period. Suppose D_i $(i = 1, 2, ..., n)$ are demands for each specific period. Then the vendor has to reorder Q_j $(j = 1, 2, ..., k)$, to satisfy demands at all times of the sale period and minimize the total expected cost at the end.

In the multi-period description, a sequence of decisions must be made with each decision affecting future decisions. This form of recursion is a principal property of Dynamic Programming. Richard Bellman invented the name Dynamic programming at RAND Corporation [14] and showed how the principle of mathematical induction is applied to solve multi-stage decision problems under uncertainty. Dynamic programming divides such problems into similar overlapping sub-problems so that their results can be re-used.

The multi-period newsvendor problems are dynamic programs because the excess inventory overlaps in subsequent sale periods. The dynamic problem of keeping inventories above safety stock is quite involved and its complex nature has to be decomposed into a sequence of simpler problems. Introducing certain assumptions, the overall order quantity can be found by solving each period at a time sequentially until all the sale periods are included. Dynamic programs tends to be immensely difficult to solve in general and are often NP [45]. This motivates us to introduce a generalization of newsvendor problem that does not require dynamic programs but still incorporates multiple periods.

In this study, we use simulation to approximate a system or operation that evolves over time. A simulation is developed to answer specific questions and also to see what happens when the objects interact with each other. According to Holden [23], there are many options when it comes to running simulations.

A Discrete Event Simulation is used to model a real world system that can be decomposed into a set of logically separate processes that evolve through time [4]. Aside from its elegant formulation, Discrete Event simulation is widely used compared to other operational research techniques because of its flexibility and ability to deal with variability and uncertainty in complex systems [9]. It basically models systems as a network of queues with changes in activities in each state of the system at discrete points in time.

1.3 Stochastic Demand and Connection to Supply Chains

A supply chain consists of a network of manufacturers who process raw materials into finished products and transport them to warehouses, distribution centers and retail outlets to be sold to customers. The supply chain involves not only the flow of material from the supplier to the customers but also the exchange of funds and information at different stages. Efficiency in supply chain coordination requires proper management of goods and services across all the various stages.

Inventory management is the control of inventory levels at different locations in the supply chain and the management of operations of products and services. For example, consider a supply chain that has only two firms – a supplier and a retailer. The retailer is modeled as newsvendor problem [47], and therefore supply chains with trivial networks are considered to be newsvendor problems.

Inventory management problems are often regarded as newsvendor problems, and correspondingly, the newsvendor problem is considered a fundamental concept in the study of supply chains management. In the real world, the operation of supply chains is complex with lots of uncertainties such as demand uncertainty, cost uncertainty and supply uncertainty. These parameters are random over time but the stochastic demand is arguably the most critical.

Order quantities for each cycle in a supply chain are decided before demand is observed at the end of each sale cycle. The average of past demands may not predict the future demand of a given product. Predicting how many to order so as to minimize cost and not allowing inventory levels to drop below safety stock levels over multiple periods is a pivotal problem in supply chain management.

1.4 A New Extension – The Food Truck Problem

As already mentioned, our discussion to this point motivates us to introduce a generalization of the newsvendor problem that does not require dynamic programs but still incorporates multiple periods. Our generalization is based on the scenario of a *food truck* such as those found around a university campus. This generalization also can incorporate demand which does not have a fixed distribution.

A food truck is a licensed vehicle stationed at a temporary location from which food is sold. In recent years, it is gaining popularity around the world. The food truck business has its own peculiar set of problems due to laws and regulations that differ for each city, county and state.

A generic food truck problem is when the owner of the food truck must purchase Q units of a food product to sell from a given location in order to meet a daily demand of D_i for that week or period. Suppose a unit of the product cost c dollars and there is a holding cost h dollars for each item in excess demand for that day. Also, suppose that the truck holds at most k items of the food product and there is a backorder cost $b = c + r$ dollars where r dollars is the cost per item of a trip to get more supplies during the week.

The demand D_i on day i is possibly dependent on the other days and on exogenous factors which are not accounted for in the daily demand distributions such as temperature, how much at the beginning of each week including backorder should be ordered in order not to only guarantee an adequate supply each day but also minimize the weekly cost of food product supply?

The food truck problem is not necessarily only related to the food truck business but to all those problems that share the characteristics of how a food trucks operates. A special case of this kind of problem is where a sandwich shop located on a college campus site orders sandwiches at a cost of c dollars each and sells them for p dollars each. The shop is likely to sell more on days when there are many classes – and thus many students on campus – as compared to days with fewer classes. How many should the shop owner order for the week so that it won't stockout and still minimize total cost? Demand distribution in this case depends on several variables apart from the number of classes on each day. At some universities, most college students go home on Fridays for the weekend and return either on a Sunday or Monday, thereby demand distribution may be dependent on a particular day.

Another special case of the food truck problem is the ice cream vendor. Suppose a vendor orders ice cream at cost c dollars per unit and sells it at p dollars per unit. Suppose also that the vendor is located at one of the dormitories of a state university or at an office complex. On a hot day, the vendor is likely to sell more and on a cold day, the vendor is likely to sell less. Thus the question is as follows: If demand is uncertain and known to vary with temperature throughout the day, then how many should the ice cream vendor order so that he/she meets customers demands at all times and still minimizes cost? The ice cream vendor cannot predict order quantity independent of a weather forecast if the demand distribution is dependent on the temperature.

The food truck problem and its special cases described above are difficult to solve and analytically intractable. The presence of exogenous variables further complicates them. Formulating a model to determine how many the vendor should order before the beginning of the multiple period, when to reorder and how many to reorder at any time of the day is not easy to develop. Our goal is to identify special cases which can be solved via well-established stochastic linear programming methods. We develop a simulation to answer specific questions such as what conditions or constraints should be included so that the food truck problems become solvable in a multi-period case.

The Simpy package written in Python [31] is used to simulate the food truck problem and the ice-cream problem as a discrete-event simulation. The newsvendor model and the simulation results can be used as decision tools for the vendors to predict optimal order quantity for the week while factoring in the presence of exogenous variables.

2 STOCHASTIC LINEAR PROGRAMMING

We first introduce the concepts of probability and randomness in order to understand stochastic linear programming.

2.1 Probability and Randomness

Probability allows us to quantify the variability in the outcome of an experiment whose exact outcome cannot be predicted with certainty [25]. The theory of probability helps to construct mathematical models for random outcomes and to make inference. To understand our methods used to solve the food truck problem, we first introduce definitions from probability theory.

Definition 2.0 *Let S denote a sample space and A be the set of events. If P is a real-valued function defined on A, then P is a probability measure if it satisfies*

1. $P(A) \geq 0$

2. $P(S) = 1$

3. If A_i is a sequence of events in A and $A_m \cap A_n = \phi$ for all $m \neq n$, then
$$P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i)$$

Definition 2.1 *A random variable X is a discrete random variable if its space is either finite or countable.*

A probability function tells us how the probability is distributed over the set of events in A [22].

Definition 2.2 *The probability mass function (pmf) of a discrete random variable X is*

$$p_X(x) = \Pr(X = x), \qquad \text{for } x \in X$$

For any value x, the probability mass function of a discrete random variable X satisfies [22],

1. $0 \leq \Pr(x) \leq 1$

2. $\sum_x \Pr(x) = 1$

Definition 2.3 *A random variable X is said to be a continuous random variable if its cumulative distribution function $F_X(x)$ is a continuous function for all $x \in \mathbb{R}$.*

Definition 2.4 *The cumulative distribution function for a random variable X is denoted as $F(x)$ and its defined as*

$$F(x) = Pr(X \leq x)$$

The probability that X takes on the value less than or equal x is denoted $F_X(x)$ and it is called the cumulative probability distribution of X. The derivative of $F(x)$ – if it exists – is the probability f_X density function of X. It satisfies the following

1. $f_X(x) \geq 0$

2. $\int_{-\infty}^{\infty} f_X(x)dx = 1$

3. $P(a \leq X \leq b) = \int_a^b f_X(x)dx$

The cumulative distribution for a discrete random variable X with probability function $\Pr(x)$ is of the form

$$F(x) = \Pr(X \leq x) = \sum_{j=-\infty}^{x} \Pr(x_j)$$

A continuous random variable X with probability density function $f(x)$ is of the form

$$F(x) = \Pr(X \leq x) = \int_{-\infty}^{x} f(t)dt$$

It follows from the fundamental Theorem of Calculus that if $F(x)$ is the cumulative density of a continuous random variable of X, then

$$\frac{d}{dx}F_X(x) = f_X(x)$$

whenever the derivative exists [25, 29].

The cumulative distribution function $F_X(x)$ is non-decreasing, which means that for all a and b, if $a < b$, then $F(a) \leq F(b)$ [22]. The probability that a random variable takes on a value from a to b is

$$\Pr(a \leq X \leq b) = \int_{a}^{b} f(t)dt = F_X(b) - F_X(a)$$

Definition 2.5 *The expected value of a random variable X is the mean or average of the probability distribution and its define as. If X is a discrete random variable with probability function $\Pr(x)$, then*

$$E(X) = \sum_{x} x \Pr(x)$$

provided $E(\mid X \mid) < \infty$.

If X is a continuous random variable with probability density function $f(x)$, then

$$E(X) = \int_{-\infty}^{\infty} x f(x)dx$$

provided $\int_{-\infty}^{\infty} \mid x \mid f(x) < \infty$.

According to Larsen [29], $E(X)$ is said to exist if the the integral in Definition 2.5 absolutely converges.

Theorem 2.1 *The expected value of a function g of a discrete random variable X with probability distribution* $\Pr(x)$ *for any real valued function $g(x)$ is*

$$E[g(X)] = \sum_x g(x) \Pr(x)$$

provided $\sum_x \mid g(x) \mid \Pr(x) < \infty$.

In the case of continuous random variable X with probability density function $f(x)$,

$$E[g(X)] = \int_{-\infty}^{\infty} g(x) f(x) dx$$

provided $\int_{-\infty}^{\infty} \mid g(x) \mid f(x) dx < \infty$

Definition 2.6 *A discrete random variable X is said to have a Poisson distribution if*

$$\Pr(x) = \frac{\lambda^x}{x!} e^{-\lambda} \qquad , \qquad x = 0, 1, 2, ...$$

where $\lambda > 0$ is the mean number events in an interval of time, Δt.

The Poisson distribution provides a good model of random events that occur infrequently in a fixed interval of time, space or any other dimension [42].

Definition 2.7 *A random variable X is said to have a exponential distribution if*

$$f(x) = \frac{1}{\theta} e^{\frac{-x}{\theta}} \qquad , \qquad 0 < x < \infty$$

where $\theta > 0$ is the scale parameter.

The time interval between consecutive occurring events is an important random variable in some situations [29]. The waiting time from the occurrence of any one event until the occurrence of the next event have an exponential distribution occurring according to a Poisson distribution [37]. If λ is the mean number of customers arriving during a time interval Δt, and θ is the mean waiting time for the next customer to arrive, then $\theta = \frac{\Delta t}{\lambda}$. The exponential distribution can also stated as

$$f(x) = \beta\, e^{-\beta x} \qquad , \qquad 0 < x < \infty$$

where $\beta = \frac{1}{\theta}$.

2.1.1 Linear Programming Theorems

Theorem 2.2 *Strong Duality : If x and y^T are feasible for the primal and dual problems respectively, then*

$$y^T b = c^T x$$

If either the primal or the dual has a finite optimal solution, then so does the other and the corresponding objective function values are equal.

Theorem 2.3 *Complementary slackness: Let x and y^T be feasible for the primal and its dual respectively of the standard program and let ϵ be the slack variable of the dual. Then x and y^T are optimal if and only if for each $j = 1, ..., n$, the following hold:*

1. If $x_j > 0$, then $\epsilon_j = 0$

2. If $\epsilon_j > 0$, then $x_j = 0$

Complementary slackness theorem is useful because it helps to interpret dual problems and dual variables.

2.2 Theoretical Solutions

2.2.1 Discrete Demand

Suppose Q_i are the order quantities and D_i are the demands for each day. Suppose also that the demand D_i for each day follows a discrete probability function such as the Poisson distribution. The cost of each item for a particular day of the week is

$$C(Q_i, D_i) = c_i Q_i + \begin{cases} b_i(D_i - Q_i) \text{ if } D_i > Q_i \\ \\ h_i(Q_i - D_i) \text{ if } D_i < Q_i \end{cases}$$

Since Q_i takes on integer values, the discrete demand D_i is defined in terms of the probability mass function (pmf). The expected cost on the i^{th} day is

$$E[C(Q_i, D_i)] = \sum_{D_i=0}^{\infty} C(Q_i, D_i) \Pr(D_i)$$

which explicitly is given by

$$E[C(Q_i, D_i)] = \sum_{D_i=0}^{\infty} \left(c_i Q_i + \begin{cases} b_i(D_i - Q_i) \text{ if } D_i > Q_i \\ \\ h_i(Q_i - D_i) \text{ if } D_i < Q_i \end{cases} \right) \Pr(D_i)$$

This in turn simplifies to

$$E[C(Q_i, D_i)] = c_i Q_i + \sum_{D_i=0}^{Q_{i-1}} h_i(Q_i - D_i)^+ \Pr(D_i) + \sum_{D_i=Q_i}^{\infty} b_i(D_i - Q_i)^+ \Pr(D_i)$$

The expected holding cost is

$$c_i Q_i + \sum_{D_i=0}^{Q_{i-1}} h_i(Q_i - D_i)^+ \Pr(D_i)$$

and the expected back-order cost is

$$c_i Q_i + \sum_{D_i=Q_i}^{\infty} b_i(D_i - Q_i)^+ \Pr(D_i)$$

Similar to the first order condition for profit maximization for quantity (Q^*) in Economics where marginal revenue is equal to marginal cost, the optimal order quantity (Q_i^*) for the item on i^{th} day can be determined at the point where the expected cost for Q and $Q+1$ are equal.

For Q_i, recall that the expected cost is

$$E[C(Q_i, D_i)] = c_i Q_i + \sum_{D_i=0}^{Q_{i-1}} h_i (Q_i - D_i)^+ \Pr(D_i) + \sum_{D_i=Q_i}^{\infty} b_i (D_i - Q_i)^+ \Pr(D_i) \tag{1}$$

For each additional quantity $Q_i + 1$, the cost of each item on i^{th} day is

$$C(Q_i + 1, D_i) = c_i(Q_i + 1) + \begin{cases} b_i(D_i - (Q_i + 1)) & \text{if } D_i > Q_i + 1 \\ h_i((Q_i + 1) - D_i) & \text{if } D_i < Q_i + 1 \end{cases}$$

The expected cost for each additional quantity $Q_i + 1$ in the discrete case is

$$E[C(Q_i + 1, D_i)] = \sum_{D_{i=0}}^{\infty} \left(c_i(Q_i + 1) + \begin{cases} b_i(D_i - Q_i - 1)^+ \\ h_i(Q_i + 1 - D_i)^+ \end{cases} \right) \Pr(D_i)$$

which simplifies to

$$E[C(Q_i{+}1, D_i)] = c_i(Q_i{+}1) + \sum_{D_i=0}^{Q_i} h_i(Q_i{+}1{-}D_i)^+ \Pr(D_i) + \sum_{D_i=Q_i+1}^{\infty} b_i(D_i{-}Q_i)^+ \Pr(D_i) \tag{2}$$

For optimal order Q_i^*,

$$E[C(Q_i, D_i)] = E[C(Q_i + 1, D_i)]$$

which implies equation (1) = equation (2). This simplifies to

$$c_i + h_i \sum_{D_i=0}^{Q_i} Pr(D_i) - b_i \sum_{D_i=Q_i+1}^{\infty} Pr(D_i) = 0 \tag{3}$$

By definition 2.4, the cumulative distribution function $F(Q_i)$ satisfies

$$1 - F(Q_i) = \sum_{D_i=Q_i+1}^{\infty} Pr(D_i)$$

which combined with Equation(3) yields

$$c_i + h_i F(Q_i) - b_i(1 - F(Q_i)) = 0$$

$$c_i + h_i F(Q_i) - b_i + b_i F(Q_i) = 0$$

$$(b_i + h_i)F(Q_i) = b_i - c_i$$

which results in our desired result of

$$F(Q_i^*) = \frac{b_i - c_i}{b_i + h_i} \tag{4}$$

2.2.2 Continuous Demand

Suppose Q_i takes on continuous values and the demand D_i is also continuous with pdf of f. The expectation of continuous distribution by Theorem 2.1 is

$$E[C(Q_i, D_i)] = \int_{-\infty}^{\infty} C(Q_i, D_i) f(D) dD$$

Since $Q_i \geq 0$

$$E[C(Q_i, D_i)] = \int_{0}^{\infty} \left(c_i Q_i + \begin{cases} b_i(D_i - Q_i) \text{ if } D_i > Q_i \\ \\ h_i(Q_i - D_i) \text{ if } D_i < Q_i \end{cases} \right) f(D) dD$$

$$E[C(Q_i, D_i)] = c_i Q_i + \left[\int_{0}^{Q_i} h_i(Q_i - D_i)^+ f(D) dD + \int_{Q_i}^{\infty} b_i(D_i - Q_i)^+ f(D) dD \right]$$

The expected holding cost is

$$c_i Q_i + \int_{D_i=0}^{Q_i} h_i(Q_i - D_i)^+ f(D) dD$$

and the expected backorder cost is

$$c_iQ_i + \int_{D_i=Q_i}^{\infty} b_i(D_i - Q_i)^+ f(D)dD)$$

We can write the expected cost as

$$E[C(Q_i, D_i)] = c_iQ_i + h_iQ_i \int_{D_i=0}^{Q_i} f(D)dD - h_i \int_{D_i=0}^{Q_i} D_i f(D)dD \\ + b_i \int_{D_i=Q_i}^{\infty} D_i f(D)dD - b_iQ_i \int_{D_i=Q_i}^{\infty} f(D)dD$$

The distribution function satisfies

$$1 - F(Q_i) = \int_{Q_i}^{\infty} f(D)dD$$

We define the mean demand as

$$\mu_i = \int_{0}^{\infty} D_i f(D)dD$$

and let define the partial expectation of demand to be

$$H(Q_i) = \int_{0}^{Q_i} D_i f(D)dD$$

It follows that

$$E[C(Q_i, D_i)] = c_iQ_i + [h_iQ_iF(Q_i) - h_iH(Q_i) + b_i(\mu_i - H(Q_i) - b_iQ_i(1 - F(Q_i))]$$

$$E[C(Q_i, D_i)] = c_iQ_i + [h_iQ_iF(Q_i) - h_iH(Q_i) + b_i\mu_i - b_iH(Q_i) - b_iQ_i + b_iQ_iF(Q_i)]$$

$$E[C(Q_i, D_i)] = c_iQ_i + h_iQ_iF(Q_i) + b_iQ_iF(Q_i) - h_iH(Q_i) - b_iH(Q_i) + b_i\mu_i - b_iQ_i$$

$$E[C(Q_i, D_i)] = c_iQ_i + (h_i + b_i)Q_iF(Q_i) - (h_i + b_i)H(Q_i) + b_i(\mu_i - Q_i)$$

$$E[C(Q_i, D_i)] = c_iQ_i + (h_i + b_i)(Q_iF(Q_i) - H(Q_i)) + b_i(\mu_i - Q_i)$$

Note that $F'(Q_i) = f(Q_i)$ and by Leibniz rule in Calculus $H'(Q_i) = Q_i f(Q_i)$. The derivative of the expected cost with respect to Q_i is

$$E'[C(Q_i, D_i)] = c_i + (h_i + b_i)(Q_i F'(Q_i) + F(Q_i) - H'(Q_i) - b_i$$

which we differentiate and set to 0 to solve for Q_i^*. Solving and simplifying leads to

$$c_i + (h_i + b_i)(Q_i^* f(Q_i^*) + F(Q_i^*) - Q_i^* f(Q_i^*)) - b_i = 0$$

$$c_i + (b_i + h_i)(F(Q_i^*)) - b_i = 0$$

$$(b_i + h_i)(F(Q_i^*)) = b_i - c_i$$

which again produces our desired results of

$$F(Q_i^*) = \frac{b_i - c_i}{b_i + h_i} \tag{5}$$

The right hand side of equation(4) and equation(5) is called the critical fractile and is the probability that the order quantity satisfies the demand in the likelihood of overage or underage. If the demand distribution is known, the optimal order quantity is

$$Q_i^* = F^{-1}\left(\frac{b_i - c_i}{b_i + h_i}\right)$$

This means that, order a quantity of Q^* for the i^{th} day to minimize the expected cost.

2.3 Approach 1 to Stochastic Linear Programs

A stochastic program is a class of optimization problems where either some or all the parameters of the problem are random at the time a decision is made [39]. In the two-stage stochastic programming model, a decision is made at the first stage before the realization of a random event. A recourse action is taken after the realization at the second stage to correct the action taken in the first stage. The two-stage stochastic linear program in a standard form is [38]

$$\begin{aligned} \text{Min} \quad & c^T x + E_\xi[Q(x,\xi)] \\ \text{s.t} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

where $Q(x,\xi)$ is the second-stage's optimal value

$$\begin{aligned} \text{Min} \quad & q^T y \\ \text{s.t} \quad & T(\xi)x + W(\xi)y = h(\xi) \\ & y \geq 0 \end{aligned}$$

The expectation of $Q(x,\xi)$ in the objective function of the first stage is taken with respect to the probability distribution of the random vector $\xi = (h, q, W, T)$. If the random vector ξ has finitely many realizations, then the possible outcomes that are observed are called scenarios [38]. If we have K scenarios with specific probability distributions $p_1, p_2, ..., p_K$ then the expectation of $Q(x,\xi)$ in the first stage is

$$E[Q(x,\xi)] = \sum_{k=1}^{K} Q(x,\xi)p_k$$

If we have a random array $\xi_k = (h_k, q_k, T_k, W_k)$ each with a probability p_k, then the two-stage stochastic linear program becomes

$$\text{Min} \quad c^T x + \sum_{k=1}^{K} (q_k)^T y_k p_k$$

$$\text{s.t} \quad Ax = b$$

$$T_k x + W_k y_k = h_k$$

$$x \geq 0, \quad y_k \geq 0, \quad k = 1, 2, ..., K$$

which can be solved via standard techniques and algorithms in linear programming.

2.4 Approach 2 to Stochastic Linear Programs

Stochastic programs are convex but often do not have differentiability [8]. Consequently, the objective function can be difficult to calculate or even write in closed form. The Sample Average Approximation (SAA) is an approach that uses Monte Carlo simulation to solve stochastic problems. According to [8], Monte Carlo simulation seems to be a natural choice for use in stochastic programs because of its applicability to higher dimensional problems.

A Sample Average Approximation begins with a random sample $s_1, s_2, ..., s_N$ from the possible realizations of the random vector ξ. Each sample is independently and identically distributed, and thus has an equal probability of occurrence compared to the other samples. The expectation function $E[Q(x, \xi)]$ is approximated by the sample average

$$E[Q(x, \xi)] = \frac{1}{N} \sum_{k=1}^{N} Q(x, s_k)$$

In terms of the Monte Carlo estimate, the two-stage stochastic problem becomes a deterministic equivalent in a form

$$\text{Min} \quad c^T x + \frac{1}{N} \sum_{k=1}^{N} Q(x, s_k)$$

$$\text{s.t} \quad Ax = b$$

$$T_k x + W_k y_k = h_k$$

$$x \geq 0, \quad y_k \geq 0, \quad k = 1, 2, ..., K$$

The SAA guarantees a near optimal solution of the true problem as the sample size increases which follows from the theorem of strong Law of Large Numbers [27].

2.5 Approach 3 to Stochastic Linear Programs

Machine Learning methods are able to process large data sets and provide good estimates to many forecasting problems in the real world [24]. Companies in this era monitor demands of their products at the various sales point and the outcomes are used in decision making. Suppose demands data $D_1, D_2, ..., D_n$ are available and they are dependent on exogenous variables or features $Z_1, Z_2, ..., Z_n$ respectively. Such features may include temperature, location, day of the week, time that predicts the demands and so on.

It follows from the Sample Average Approximation that, a good estimate of the vendor's expected cost is minimizing the average of the samples of the cost.

$$\min_{Q_i} \quad \frac{1}{n}\sum_{i=1}^{n} c_i Q_i + C_{o_i}[Q_i - D_i]^+ + C_{u_i}[D_i - Q_i]^+$$

Suppose that the order quantity depends on the features and some parameters. Let $Q_i = Z_i^T\beta$ where Z_i^T is the transpose of Z_i and β represents a vector of parameters. The objective function of the vendor after substitution of Q_i is

$$\min_{\beta} \quad \frac{1}{n}\sum_{i=1}^{n} c_i Z_i^T\beta + C_{o_i}[Z_i^T\beta - D_i]^+ + C_{u_i}[D_i - Z_i^T\beta]^+$$

A regularization term is added to the cost function to avoid over-fitting, typically the L^k norm of the vector parameter. The choice of L^k norm is based on a belief of the number of features involve in predicting the demand [3]. We add L^k regularization and the model becomes

$$\min_{\beta} \quad \frac{1}{n}\sum_{i=1}^{n} c_i Z_i^T \beta + C_{o_i}[Z_i^T\beta - D_i]^+ + C_{u_i}[D_i - Z_i^T\beta]^+ + \lambda \mid\mid \beta \mid\mid_k$$

where $\lambda \geq 0$ is called a tuning or regularization parameter.

If $S_i = Z_i^T\beta - D_i$ and $t_i = D_i - Z_i^T\beta$, then machine learning model is transformed into a linear programming problem

$$\min_{\beta} \quad \frac{1}{n}\sum_{i=1}^{n} c_i Z_i^T \beta + C_{o_i} S_i + C_{u_i} t_i + \lambda \mid\mid \beta \mid\mid_k$$

$$\text{s.t} \quad S_i + Z_i^T\beta \geq D_i$$

$$Z_i^T\beta - t_i \leq D_i$$

$$S_i \geq 0,\ t_i \geq 0$$

The solution to the Linear Program is

$$\beta^* = \text{argmin} \quad \frac{1}{n}\sum_{i=1}^{n} c_i Z_i^T \beta + C_{o_i} S_i + C_{u_i} t_i + \lambda \mid\mid \beta \mid\mid_k$$

The optimal order quantity given features Z_i^T is

$$Q_i^* = Z_i^T \beta^*$$

The decision rule depends upon the most recent data available, therefore $Q_i^* = Z_{\text{new}}^T\beta^*$

3 THE FOOD TRUCK PROBLEM

In this thesis, the food truck problem is an extension of the newsvendor problem, one that we model with Discrete event simulation.

3.1 Derivation and Explanation of all the Deterministic Variations

Our goal is to estimate the order quantities that minimizes the cost in the food truck problem for multiple periods. The owner of a food truck must purchase Q_{ij} units each of M food products to sell from a given location on campus in order to meet a daily demand of D_{ij} units of the jth product for that week, where $i = 1, 2, 3, 4, 5$ corresponds to Monday through Friday. Suppose a unit of the j^{th} food product costs c_j dollars and suppose that there is a holding cost h_j for each item in excess of the demand for that day. Also, suppose that the truck holds at most K items of the food product and that there is a backorder cost for the j^{th} product, $b_j = c_j + r_j$ where r_j is the cost per item of a trip to get more supplies of the j^{th} product during the week. If the demand D_{ij} on day i is possibly dependent on the other days and on exogenous factors not accounted for in the daily distributions, how much should be ordered each week (at the beginning and including the backorder) in order to not only guarantee an adequate supply each day, but also to minimize the weekly cost of the food products supply?

3.2 Reduction to Classical Newsvendor

The food truck problem described above has characteristics of the newsvendor problem. Similar to the newsvendor, the food vendor has to make initial order decision once before the week starts. The demand is random since he does not know exact number of people who may purchase the food products. The food vendor incurs a cost to preserve the food product if he orders too much. The unsold food products for a particular day are carried over to next days. The food vendor also incurs a cost if he orders too few because of missed sales. If the food truck is so large that capacity is not an issue, then the food vendor may order the food products once per week on a fixed day before sale begins. Also if the demands for each day are independent and identically distributed, then the food truck problem reduces to the classical newsvendor problem with multiple periods.

3.2.1 Reduction to Classical Newsvendor – Simulation

We first demonstrate that the Food Truck problem can be reduced to a classical newsvendor. Consider a single food product which has the following parameters fixed for the week:

- $c = 3$ for cost of each food item.
- $p = 1$ for profit for each food item.
- $h = 0.5$ for holding cost per unit.
- $b = 4$ for backorder cost per unit.
- $r = 1$ for return trip cost per item.

If demands for each day of the week are independently and identically distributed, then the total demand for the week is

$$D_T = D_1 + D_2 + D_3 + D_4 + D_5$$

If the optimal order to meet each independent day's demand is $Q_1^*, Q_2^*, ..., Q_5^*$ respectively, then the optimal order quantity for the week is

$$Q_T^* = Q_1^* + Q_2^* + Q_3^* + Q_4^* + Q_5^*$$

The critical fractal is $F(Q^*) = 0.2222$.

We assume that customer arrival is a Poisson process and the inter-arrival rate to purchase the food product follows an exponential distribution with $\theta = 1$. This implies that the time between customer arrivals has a mean of 1 minute. If the food truck operates for 5 hours daily, then the demand follows a Poisson distribution with mean rate $\lambda = 300$ per a day. Figure 2 shows a randomly generated data of means of demands for 100 different samples.

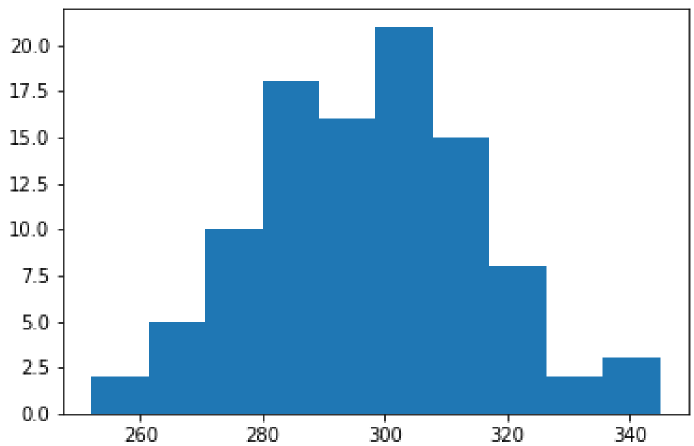

Figure 2: Means of demands across this range of demands

We constructed the cumulative distribution of the corresponding Poisson distribution for a range of 200 to 400, and then we searched for the critical fractile within this range as described in Section 2. A theoretical optimal order for each day of $Q_i^* = 287$ minimizes the expected total cost.

Simpy is used to model the food truck problem and the code is listed in Appendix A.1. The model parameters are defined in a Python class and assigned attributes. The main components of the food truck are order quantity, demand and cost. Customer inter-arrival times are modeled by a Discrete Event Simulation. Similar to the theoretical solution, we assume that the arrival of customers follow a Poisson process and use the exponential distribution to generate customer inter-arrival time with a mean $\theta = 1$. The food truck operates for 5 hours, so the simulation time for the day is 300. SAA approach is used to compute the expected cost for each corresponding order quantity from a sample size of 100. Figure 3 is a plot of the expected cost and the order quantity.

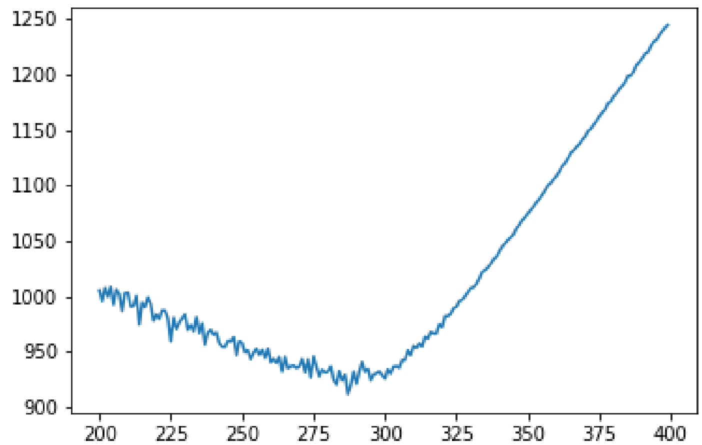

Figure 3: Order quantity and expected cost

The results for the simulation show that the optimal order quantity for the day yields a minimum cost when $Q^* = 287$. This is similar to the results we had in the theoretical solution.

Different parameters of cost values yields different critical fractiles. The holding cost per unit, back order cost per unit and the cost per unit of the product are varied over a range of values. The theoretical procedure is repeated for these different cost parameters and the results of the order quantity are compared to that of the simulation when these cost values held constant for theoretical approach and simulation.

Table 1: Samples From Parameter Sweeps

c, h, r	Critical Fractile	Theoretical Q^*	Simulation Q^*
c=4, h=1, r=2	0.2857	290	290
c=4, h=0.5, r=1	0.1818	284	285
c=2, h=1, r=2	0.4000	295	295
c=3, h=2, r=2	0.2857	290	289
c=1, h=2, r=2.5	0.4545	298	298

From Table 1, we observe that the simulation result is either the same or close to the theoretical solution over a range of parameters in the newsvendor problem and therefore assert that the food truck problem can be reduced to a classical newsvendor problem.

3.3 Simulation of the Full Food Truck Problem

The Food truck problem deviates from a typical newsvendor problem when we consider certain variations. Demand of the food product depends on many features which are not identically distributed, especially in the case of the food truck problem. The demand distribution may be dependent on the number of classes on each day. Intuitively, it is expected that the more classes on each day, the higher the demand. Also, the demand may be influenced by the temperature of the day. If the temperature is cold, many customers may prefer to stay indoors and if the temperature is high, many customers may want to come outside. The location of the food truck is a vital feature to consider since it affects the demand. The day of the week also tends to affect the demand. These variations do not reduce the food truck problem to a newsvendor problem.

The code listed in Appendix A.2 is for a multi-period simulation of a modified food truck problem where the vendor reorders only once in midweek. Two cases of the simulation are conducted. We simulated a case for the modified food truck problem when demand is independent and identically distributed (newsvendor case) and when demand is dependent on time (not newsvendor case). A Simpy environment is created for each day and the demands and missed sales are monitored. The optimal order quantity for the two cases are shown in Table 2.

Table 2: Demand Dependent on Time and Independent Demand

c, h, r	Critical Fractile	Reorder	Demand–time Q^*	Demand–No time Q^*
c=3, h=0, r=1	0.2500	200	1450	1480
c=4, h=1, r=1	0.1667	200	1420	1423
c=3, h=0,5, r=2	0.3636	500	1422	1450
c=4, h=1, r=2	0.2857	100	1420	1426

In Table 2, the simulated optimal order of the week if demand is time dependent (not newsvendor problem) and the optimal order for the week if demand is independent and identically distributed (newsvendor problem) for different parameters varies when the mid-week order is held fixed for the two cases. The variability in the optimal value indicates that the two cases are not the same. The food truck problem does not reduce to the classical newsvendor problem in the presence of an exogenous variable – time. The simulation results show that in the presence of exogenous variables such as time, location, temperature and so on, the food truck problem does not reduce to the newsvendor problem.

3.4 Theoretically Solvable Variation of the Food Truck Problem

The Full Food Truck problem is intractable and combinatorially complex. Note that the objective is to find both how much to order and *when* to reorder to minimize the expected cost. If we fix the reorder time to a single day (Wednesday) midweek, then it is possible to reduce the Food truck problem to a theoretically solvable problem which does not reduce to a classical newsvendor.

Suppose the Food truck vendor orders quantity Q_{0j} for the j^{th} product before the week starts and reorders quantity Q_{1j} on mid-week. The realization of demands on each day of the week are $D_{1j}, D_{2j}, D_{3j}, D_{4j}, D_{5j}$. The goal of the vendor is how much of the j^{th} food product should be ordered before the week starts and how much of product j should be ordered in midweek to minimize the expected cost for the week. Let $C_{oj} = c_j + h_j$ be overage cost and $C_{uj} = c_j + b_j$ be the underage cost for the j^{th} product. The vendor can run-out on any day or carry-over items to next days, but the vendor only reorders on midweek.

If the vendor does not run-out on Monday, then $Q_{0j} - D_{1j}$ of the j^{th} items will be left and carry over to Tuesday. The vendor may also run-out on Monday and there is nothing to sell on Tuesday. Thus, cost on Monday is

$$c_j Q_{0j} + C_{oj}[Q_{0j} - D_{1j}]^+ + C_{uj}[D_{1j} - Q_{0j}]^+$$

Let $S_{1j} = Q_{0j} - D_{1j}$ denote the j^{th} food item left to sell on Tuesday. If there is nothing to sell on Tuesday, then the vendor incurs a cost $C_{uj}D_{2j}$. Thus, cost on Tuesday is

$$C_{uj}D_{2j} + C_{oj}[S_{1j} - D_{2j}]^+ + C_{uj}[D_{2j} - S_{1j}]^+$$

Let $S_2 = S_1 - D_2$ denote the food item left to sell on Wednesday. If there is nothing left to sell on Wednesday, the vendor reorders and the cost is $C_{uj}D_{3j} + r_jQ_{1j}$. The vendor still reorders irrespective of whether there is overage or underage. Thus, cost on Wednesday is

$$C_{uj}D_{3j} + r_jQ_{1j} + C_{oj}[S_{2j} - D_{3j}]^+ + r_jQ_{1j} + C_{uj}[D_{3j} - S_{2j}]^+ + r_jQ_{1j}$$

If the vendor sells all old stock of the j^{th} food items and is left with the new stock reordered, then the cost on Thursday is $C_{oj}[Q_{1j} - D_{4j}]^+ + C_{uj}[D_{4j} - Q_{1j}]^+$. There may be old stock of j^{th} item which carry over in addition to the new reorder quantity. Thus $(S_{2j} - D_{3j}) + Q_{1j}$ is the quantity to carry over into Thursday. The cost on Thursday is

$$C_{oj}[Q_{1j} - D_{4j}]^+ + C_{uj}[D_{4j} - Q_{1j}]^+ + C_{oj}[(S_{3j} + Q_{1j}) - D_{4j}]^+ + C_{uj}[D_{4j} - (S_{3j} + Q_{1j})]^+$$

If the vendor has Q_{1j} to sell on Thursday and couldn't sell all, then $Q_{1j} - D_{4j}$ will be left to sell on Friday. Let $S_{4j} = Q_{1j} - D_{4j}$. The cost of this scenario on Friday is $C_{oj}[S_{4j} - D_{5j}]^+ + C_{uj}[D_{5j} - S_{4j}]^+$. If also the vendor runs-out of the j^{th} item on Thursday, then there is nothing to sell for that item on Friday and the cost is $C_{uj}D_{5j}$. If $S_{5j} = (S_{3j} + Q_{1j}) - D_{4j}$ is left on Thursday and carries over to Friday, then the cost on Friday is

$$C_{uj}D_{5j} + +C_{oj}[S_{4j} - D_{5j}]^+ + C_{uj}[D_{5j} - S_{4j}]^+ + C_{oj}[S_{5j} - D_{5j}]^+ + C_{uj}[D_{5j} - S_{5j}]^+$$

We sum the cost on each day to get the total cost for the week for the j^{th} item

$$c_jQ_{0j} + C_{oj}[Q_{0j} - D_{1j}]^+ + C_{uj}[D_{1j} - Q_{0j}]^+ + C_{uj}D_{2j} + C_{oj}[S_{1j} - D_{2j}]^+ + C_{uj}[D_{2j} - S_{1j}]^+$$
$$+C_{uj}D_{3j} + rQ_{1j} + C_{oj}[S_{2j} - D_{3j}]^+ + rQ_{1j} + C_{uj}[D_{3j} - S_{2j}]^+ + r_jQ_{1j} + C_{oj}[Q_{1j} - D_{4j}]^+$$
$$+C_{uj}[D_{4j} - Q_{1j}]^+ + C_{oj}[(S_{3j} + Q_{1j}) - D_{4j}]^+ + C_{uj}[D_{4j} - (S_{3j} + Q_{1j})]^+ + C_{uj}D_{5j}+$$
$$C_{oj}[S_{4j} - D_{5j}]^+ + C_{uj}[D_{5j} - S_{4j}]^+ + C_{oj}[S_{5j} - D_{5j}]^+ + C_{uj}[D_{5j} - S_{5j}]^+$$

To simplify the nonlinear expressions, we introduce S_{ij} and t_{ij} variables and define corresponding constraints. The total cost function is

$$c_1Q_{0j} + C_{oj}S_{1j} + C_{uj}t_{1j} + C_{oj}S_{2j} + C_{uj}t_{2j} + C_{oj}S_{3j} + C_{uj}t_{3j} + C_{oj}S_{4j} + C_{uj}t_{4j}$$
$$+C_{oj}S_{5j} + C_{uj}t_{5j} + C_{oj}S_{6j} + C_{uj}t_{6j} + C_{oj}S_{7j} + C_{uj}t_{7j} + 3r_jQ_{1j} + C_{uj}D_{2j} + C_{uj}D_{3j} + C_{uj}D_{5j}$$

where $S_{1j} = Q_{0j} - D_{1j}$, $S_{2j} = S_{1j} - D_{2j}$, $S_{3j} = S_{2j} - D_{3j}$, $S_{4j} = Q_{1j} - D_{4j}$, $S_{5j} = (S_{3j} + Q_{1j}) - D_{4j}$, $S_{6j} = S_{4j} - D_{5j}$, $S_{7j} = S_{5j} - D_{5j}$, $t_{1j} = D_{1j} - Q_{0j}$, $t_{2j} = D_{2j} - S_{1j}$, $t_{3j} = D_{3j} - S_{2j}$, $t_{4j} = D_{4j} - Q_{1j}$, $t_{5j} = D_{4j} - (S_{3j} + Q_{1j})$, $t_{6j} = D_{5j} - S_{4j}$, $t_{7j} = D_{5j} - S_{5j}$.

The terms $C_{uj}D_{2j}$, $C_{uj}D_{3j}$, $C_{uj}D_{5j}$ are constants. If K is a constant and $f(x)$ is a function, then

$$\text{argmin } \{K + f(x)\} = \text{argmin } \{f(x)\}$$

Consequently, minimizing the total sum of the costs of M products for the various days is formulated as a linear program of the form

$$\min \qquad \sum_{j=1}^{M} c_j Q_{0j} + 3r_j Q_{1j} + C_{oj}\mathbf{1^T S_j} + \mathbf{C_{uj} 1^T t_j}$$

$$\text{s.t} \qquad Q_{0j} - S_{1j} \leq D_{1j}$$

$$Q_{1j} - S_{4j} \leq D_{4j}$$

$$Q_{1j} - S_{3j} - S_{6j} \leq D_{4j}$$

$$S_{ij} - S_{i+1,j} \leq D_{i+1,j}$$

$$Q_{0j} + t_{1j} \geq D_{1j}$$

$$Q_{1j} + S_{3j} + t_{6j} \geq D_{4j}$$

$$S_{ij} + t_{i+1,j} \geq D_{i,j}$$

$$S_{ij} \geq 0, t_{ij} \geq 0, \qquad i = 1, ..., 7 \text{ and } j = 1, ..., M$$

This is the formulation of the food truck problem for M multi-products with a mid-week order. We are going to focus on $M = 1$ product and obtain the dual. The dual program of the Food truck linear program in a vector form is

$$\max \qquad D^T(\mathbf{z}-\mathbf{v})$$

$$\text{s.t} \qquad -v_1+z_6 \le c$$

$$-v_6-v_4+z_7+z_3 \le 3r$$

$$\mathbf{v}-\mathbf{v_{i+1}}+\mathbf{z} \le C_o$$

$$v_5 \ge C_o$$

$$0 \le \mathbf{z} \le \mathbf{C_u}$$

$$v_i \ge 0 \qquad i=1,...,7$$

If we let $\mathbf{u}=\mathbf{z}-\mathbf{v}$, then the dual program is equivalent to

$$\max \qquad D^T\mathbf{u}$$

$$\text{s.t} \qquad -v_1+v_6+u_6 \le c$$

$$v_3-v_4-v_6+v_7+u_3+u_7 \le 3r$$

$$2\mathbf{v}-\mathbf{v_{i+1}}+\mathbf{u} \le C_o$$

$$v_5 \ge C_o$$

$$0 \le \mathbf{u}+\mathbf{v} \le \mathbf{C_u}$$

$$v_i \ge 0,\;\; u_i \ge 0 \qquad i=1,...,7$$

3.5 Interpretation and Implementation of the Dual Program

The dual formulated has objective to maximize the cost function with deterministic demand and a vector **u** while the primal's objective is the minimization of the cost function. These two objectives are equivalent base on the theorem of strong duality. There is a relationship between the slackness in a primal constraints and the slackness of the associated dual variable according to complementary slackness. To maximize the objective function in the dual, the variable u must be made large as possible. Some of $v_i = 0$ for u to be large, This implies that, some of the constraints in the primal are not binding. Variations of these constraints would not change the optimal solution.

Demand is random and therefore from the stochastic version of the dual is

$$\max \qquad D^T u + E_D[Q(u, D_i)]$$

$$\text{s.t} \qquad -v_1 + v_6 + u_6 \leq c$$

$$v_3 - v_4 - v_6 + v_7 + u_3 + u_7 \leq 3r$$

$$2\mathbf{v} - \mathbf{v_{i+1}} + \mathbf{u} \leq C_o$$

$$v_5 \geq C_o$$

$$0 \leq \mathbf{u} + \mathbf{v} \leq \mathbf{C_u}$$

$$v_i \geq 0, \ u_i \geq \qquad i = 1, ..., 7$$

where $E_D[Q(u, D_i)]$ is the optimal value of the second stage

$$\begin{aligned} \max \quad & q^T y \\ \text{s.t} \quad & T(D_i)u + W(D_i)y = h(D_i) \\ & y \geq 0 \end{aligned}$$

A decision is made at the first stage in the stochastic version of the dual on the initial order. A recourse action is taken at the second stage. The demand is random and has finite realization. The optimal value is approximated using SAA and thus

$$E[Q(u, D)] = \frac{1}{N} \sum_{k=1}^{N} Q(u, D)$$

The result gives the sample average approximation of the demand. This implies that to maximize the expected profit, we should order a quantity equal to the approximated demand. We then simulate the demand in the stochastic descriptions of the dual in the food truck problem.

The modification of the Food truck problem does not reduce to the newsvendor problem but can be solved. We have formulated the special case of food truck problem as a linear program and derived its dual. The Duality Theorem and the associated Theorem on Complementary Slackness give us reason to believe that the primal and dual are related problems. We observe that the demands D_i appear in the objective function of the dual. The constraints are deterministic. Instead of simulating the entire process to get the solution to the food truck problem, simulating the dual only requires simulating the demand.

Simulating the demand and using the SAA approach, we get the estimated demand for the particular day and the optimal quantity orders to minimize the expected cost is equal to the simulated estimated demands.

For the special case of the food truck problem with mid-week order, the food truck dual is created as Python class. The code listed in Appendix A.3 have random demands which are dependent on time generated in Simpy. An environment is created for the simulation week. The demand for each day is simulated and the demand for the entire week is displayed as results. From the results of the dual stochastic linear program, the optimal order quantity of the j^{th} product that minimizes the vendor's expected cost is equivalent to the simulated demand for the week.

4 FUTURE DIRECTIONS

In future analysis, a Full Food truck problem with multi-products and multi-period should be formulated as a stochastic linear program without any modification to help the vendor decide on how much to order initially and how much to re-order at any given time of the day. The model should incorporate features like temperature, time, location etc. and be extended to supply chains. Dynamic programming and Deep learning approaches should also be considered in a data driven case of the Food Truck Problem.

BIBLIOGRAPHY

[1] Layek L Abdel-Malek and Nathapol Areeratchakul. A quadratic programming approach to the multi-product newsvendor problem with side constraints. *European Journal of Operational Research*, 176(3):1607–1619, 2007.

[2] Emel Arikan. *Single Period Inventory Control and Pricing: An Empirical and Analytical Study of a Generalized Model*. Peter Lang International Academic Publishers, 2018.

[3] Gah-Yi Ban and Cynthia Rudin. The big data newsvendor: Practical insights from machine learning. *Operations Research*, 67(1):90–108, 2019.

[4] Jeffrey S Barrett, Bhuvana Jayaraman, Dimple Patel, and Jeffrey M Skolnik. A sas-based solution to evaluate study design efficiency of phase i pediatric oncology trials via discrete event simulation. *Computer methods and programs in biomedicine*, 90(3):240–250, 2008.

[5] Mokhtar S Bazaraa, John J Jarvis, and Hanis D Sherali. *Linear programming and network flows*. John Wiley & Sons, 2008.

[6] Uri Benzion, Yuval Cohen, and Tal Shavit. The newsvendor problem with unknown distribution. *Journal of the Operational Research Society*, 61(6):1022–1031, 2010.

[7] Dimitris Bertsimas and Aurélie Thiele. A data-driven approach to newsvendor problems. *Working Paper, Massachusetts Institute of Technology*, 2005.

[8] John R Birge and Francois Louveaux. *Introduction to stochastic programming*. Springer Science & Business Media, 2011.

[9] SC Brailsford and NA Hilton. A comparison of discrete event simulation and system dynamics for modelling health care systems. 2001.

[10] Erik Brynjolfsson, Yu Hu, and Duncan Simester. Goodbye pareto principle, hello long tail: The effect of search costs on the concentration of product sales. *Management Science*, 57(8):1373–1386, 2011.

[11] Gerard Cachon and Christian Terwiesch. *Matching supply with demand*. McGraw-Hill Publishing, 2008.

[12] Tatyana Chernonog and Noam Goldberg. On the multi-product newsvendor with bounded demand distributions. *International Journal of Production Economics*, 203:38–47, 2018.

[13] Gerard Cornuejols and Reha Tütüncü. *Optimization methods in finance*, volume 5. Cambridge University Press, 2006.

[14] Stuart Dreyfus. Richard bellman on the birth of dynamic programming. *Operations Research*, 50(1):48–51, 2002.

[15] Francis Y Edgeworth. The mathematical theory of banking. *Journal of the Royal Statistical Society*, 51(1):113–127, 1888.

[16] Ijeoma W Ekwegh. Newsvendor models with monte carlo sampling. 2016.

[17] Guillermo Gallego. Production management. *lecture notes, Columbia University, New York, NY*, 1995.

[18] Helena Gaspars-Wieloch. Newsvendor problem under complete uncertainty: a case of innovative products. *Central European Journal of Operations Research*, 25(3):561–585, 2017.

[19] George Hadley and Thomson M Whitin. Analysis of inventory systems. Technical report, 1963.

[20] F Heller and GJ Liberman. Introduction to operation research 7th ed. 2000.

[21] Arthur V Hill. The newsvendor problem. *White Paper*, pages 57–23, 2011.

[22] Robert V Hogg, Joseph McKean, and Allen T Craig. *Introduction to mathematical statistics*. Pearson Education, 2005.

[23] Lauren Holden. Inventory optimization using a simpy simulation model. 2017.

[24] Jakob Huber, Sebastian Müller, Moritz Fleischmann, and Heiner Stuckenschmidt. A data-driven newsvendor problem: From data to decision. *European Journal of Operational Research*, 278(3):904–915, 2019.

[25] Richard A Johnson, Irwin Miller, and John E Freund. *Probability and statistics for engineers*, volume 2000. Pearson Education London, 2000.

[26] Moutaz Khouja. The single-period (news-vendor) problem: literature review and suggestions for future research. *Omega*, 27(5):537–553, 1999.

[27] Anton J Kleywegt, Alexander Shapiro, and Tito Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2002.

[28] Bernard Kolman and Robert E Beck. *Elementary linear programming with applications*. Gulf Professional Publishing, 1995.

[29] Richard J Larsen and Morris L Marx. *An introduction to mathematical statistics*. Prentice Hall, 2005.

[30] Abdalla Mansur, Jack Brimberg, and Ghaleb El-Refae. Financial planning in the public sector and the news vendor problem. *Yugoslav Journal of Operations Research*, 28(2):265–274, 2018.

[31] Norm Matloff. Introduction to discrete-event simulation and the simpy language. *Davis, CA. Dept of Computer Science. University of California at Davis. Retrieved on August*, 2(2009):1–33, 2008.

[32] Péter Mileff and Károly Nehéz. An extended newsvendor model for customized mass production. *Advanced Modeling and Optimization*, 8(2):169–186, 2006.

[33] H Kaan Okyay, Fikri Karaesmen, and Süleyman Özekici. Newsvendor models with dependent random supply and demand. *Optimization Letters*, 8(3):983–999, 2014.

[34] Marcel Oliver. Practical guide to the simplex method of linear programming, 2012.

[35] Yan Qin, Ruoxuan Wang, Asoo J Vakharia, Yuwen Chen, and Michelle MH Seref. The newsvendor problem: Review and directions for future research. *European Journal of Operational Research*, 213(2):361–374, 2011.

[36] Ruozhen Qiu, Minghe Sun, and Yun Fong Lim. Optimizing (s, s) policies for multi-period inventory models with demand distribution uncertainty: Robust dynamic programing approaches. *European Journal of Operational Research*, 261(3):880–892, 2017.

[37] Richard Scheaffer and Linda Young. *Introduction to Probability and its Applications*. Nelson Education, 2009.

[38] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. *Lectures on stochastic programming: modeling and theory*. SIAM, 2014.

[39] Alexander Shapiro and Andy Philpott. A tutorial on stochastic programming. *Manuscript. Available at www2. isye. gatech. edu/ashapiro/publications. html*, 17, 2007.

[40] John Sloman and A Wride. *Economics*. Pearson Education Limited, 2004.

[41] Nazli Turken, Yinliang Tan, Asoo J Vakharia, Lan Wang, Ruoxuan Wang, and Arda Yenipazarli. The multi-product newsvendor problem: Review, extensions, and directions for future research. *Handbook of newsvendor problems*, pages 3–39, 2012.

[42] Dennis Wackerly, William Mendenhall, and Richard L Scheaffer. *Mathematical statistics with applications*. Cengage Learning, 2014.

[43] Charles X Wang and Scott Webster. The loss-averse newsvendor problem. *Omega*, 37(1):93–105, 2009.

[44] Chun-Chin Wei and Liang-Tu Chen. Supply chain replenishment for multi-period newsvendor products.

[45] Gerhard J Woeginger. Exact algorithms for np-hard problems: A survey. In *Combinatorial optimization—eureka, you shrink!*, pages 185–207. Springer, 2003.

[46] Hamed Yarmand, Julie S Ivy, Brian Denton, and Alun L Lloyd. Optimal two-phase vaccine allocation to geographically different regions under uncertainty. *European Journal of Operational Research*, 233(1):208–219, 2014.

[47] Fuqiang Zhang. Supply chain coordination. *Wiley Encyclopedia of Operations Research and Management Science*, 2010.

[48] Guoqing Zhang, Jianmai Shi, Sohail S Chaudhry, and Xindan Li. Multi-period multi-product acquisition planning with uncertain demands and supplier quantity discounts. *Transportation Research Part E: Logistics and Transportation Review*, 132:117–140, 2019.

APPENDICES

A Discrete Event Simulation – SimPy Code

A.1 Reduction to the Classical Newsvendor Problem

The code listing simulates the solution of the food truck problem reduced to a newsvendor problem.

```
#!/usr/bin/env python
# coding: utf-8

import matplotlib.pyplot as plt
import numpy as np
import collections
import simpy as sim

## Defining parameters
c = 3      # cost of food item
p = 1      # profit (excess of cost) of each item
h = 0.0    # holding cost in dollars
r = 1     # return trip cost per item
b = c + r # back order cost per unit
K = 1000  # Truck Capacity

CF = (b - c)/(b+h)

from scipy.stats import poisson

   = 300
NumOfSamples = 100  #Number of samples

print(poisson.rvs(  ,size = NumOfSamples))

#Histogram
list = poisson.rvs(  ,size = NumOfSamples)
plt.hist(list);

P_cdf = np.array( [ poisson.cdf(k,  ) for k in range
   (200,400) ] )
ind = np.searchsorted(P_cdf, CF)

Qstar = 200 + ind # Optimal order quantity (Theoretical)

## Foodtruck
```

```
## order quantity Q
## sell items

#@dataclass
class Foodtruck(object):

    def __init__(self, environment, order_quantity  = 0):
        ##  Q is the order quantity
        self.Q = order_quantity
        self._Q = order_quantity
        self.env = environment
        self.NumberSold = 0
        self.MissedSales = 0

    def sell_item(self, quantity = 1):
        ## inventory descreases by quantity
        ## inventory = 0 ==> generate underage cost
        while True:
            if( self._Q > 0 ):
                self.NumberSold += quantity
                self._Q -= quantity
                self.MissedSales = max( -self._Q, 0)
            else:
                self.MissedSales += quantity
            t = self.env.now
            interarrival_time = np.random.exponential(1) #
                1 min between sales, on average
            yield self.env.timeout(interarrival_time)


def SimDay( order_quantity, verbose=False):
    ## Food Truck open 5 hours per day
    env = sim.Environment()
    DailySimTime = 5*60  #time in minutes

    ## Create an instance of Food truck
    vendor = Foodtruck(env, order_quantity =
       order_quantity)
    env.process(vendor.sell_item())
    env.run(until = env.now + DailySimTime)
    return vendor

def CostCalculation(ASimulatedDay):
    Q=ASimulatedDay.Q
    vendorResult= ASimulatedDay
    D=vendorResult.NumberSold+vendorResult.MissedSales
    Underage=  (D-Q) ; Overage= (Q-D)
    #print(c,b,h,Underage, Overage)
    Cost = c*Q + b*max(Underage,0) + h*max(Overage,0)
```

```
    return (Cost)

#Order Quantity of 141
ExpectedCost = np.array( [ CostCalculation(SimDay(141,
   verbose = False)) for _ in range(100) ]).mean()

ExpectedCost = []
for i in range(200,400):
    ExpectedCost.append(np.array([CostCalculation(SimDay(i
       , verbose = False)) for _ in range(100)]).mean())

orderquantity =(range(200,400))

plt.plot(orderquantity, ExpectedCost);
plt.savefig("simulate")

print(orderquantity[np.argmin(ExpectedCost)])# Optimal
   order quantity (Simulation)

print(Qstar) # Optimal order quantity (Theoretical)
```

A.2 Full Simulation of the Food Truck Problem with Mid-week Reorder

The code listing performs a simulation of two cases of the Food truck problem. One case is when the demands are independently and identically distributed (Newsvendor) and the other case is when demands are dependent on time. The code generates demands using inter-arrival times and simulates the optimal order quantity for the two cases.

```
#!/usr/bin/env python
# coding: utf-8

import matplotlib.pyplot as plt
import numpy as np
import collections
import simpy as sim
import tqdm

## Defining parameters
c = 1     # cost of food item
p = 1     # profit (excess of cost) of each item
h = 2   # holding cost in dollars
```

```python
r = 2.5     # return trip cost per item
b = c + r # back order cost per unit
K = 1000  # Truck Capacity

CF = (b - c)/(b+h)# Critical Fractile

from scipy.stats import poisson

  = 300
NumOfSamples = 100  ## Number of samples

print(poisson.rvs(  ,size = NumOfSamples))

Qrange = range(1400,1600) #range of values for optimal order
P_cdf = np.array( [ poisson.cdf(k,  ) for k in Qrange ] )
ind = np.searchsorted(P_cdf, CF)

## Foodtruck
## order quantity Q
## sell items

#@dataclass
class Foodtruck(object):

    def __init__(self, environment, order_quantities):
        ##  Q is the order quantity
        self.Q1 = order_quantities[0]  #First order
        self.Q2 = order_quantities[1]
        self.Q = order_quantities[0]
        self.TotalCost = 0
        self.Demand = 0
        self.env = environment
        self.NumberSold =  0
        self.MissedSales = 0

    def sell_item(self, quantity = 1):
        ## inventory descreases by quantity
        ## inventory = 0 ==> generate underage cost
        while True:
            if( self.Q > 0 ):
                self.NumberSold += quantity
                self.Q -= quantity
                self.MissedSales = max( -self.Q, 0)
            else:
                self.MissedSales += quantity
            self.Demand += 1

            ## time dependent demand
            t = self.env.now
```

```
            Demand = 60.0 + 20*np.sin(np.pi*(t/
               DailySimTime -1/2))  ## expected demand
               varies with time
            interarrival_time = np.random.exponential(60/
               Demand) # 1 min between sales, on average
            yield self.env.timeout(interarrival_time)

    def calculate_cost(self):
        Overage = self.Q  #Stock Remaining at end of day
        Underage = self.MissedSales  #Unrealized Sales
        DailyCost = b*Underage + h*Overage
        return DailyCost

## Food Truck open 5 hours per day
DailySimTime = 5*60  #time in minutes

def SimDay(vendor, verbose=False):
    vendor.env = sim.Environment()
    vendor.MissedSales = 0
    vendor.NumberSold = 0
    vendor.env.process(vendor.sell_item())
    vendor.env.run(until = vendor.env.now + DailySimTime)
    vendor.TotalCost += vendor.calculate_cost()
    return vendor

def SimWeek( order_quantities, verbose=False):
    """First is initial order quantity, and second is the
       midweek order"""
    ## Food Truck open 5 hours per day
    env = sim.Environment()

    ## Create an instance of Food truck
    vendor = Foodtruck(env, order_quantities =
       order_quantities)
    vendor.TotalCost += c*vendor.Q1
    if(verbose):
        print('Day      Q    Missed   Sold ')
    PrintString = ' %s  %7d %7d %7d'
    SimDay(vendor) #Monday
    if(verbose):
        print(PrintString % ('M', vendor.Q, vendor.
           MissedSales, vendor.NumberSold))
    SimDay(vendor) #Tuesday
    if(verbose):
        print(PrintString % ('T', vendor.Q, vendor.
           MissedSales, vendor.NumberSold) )
    SimDay(vendor) #Wednesday
    if(verbose):
```

```python
        print(PrintString % ('W', vendor.Q, vendor.MissedSales, vendor.NumberSold) )
    vendor.Q += vendor.Q2
    vendor.TotalCost += c*vendor.Q2
    SimDay(vendor) #Thursday
    if(verbose):
        print(PrintString % ('R', vendor.Q, vendor.MissedSales, vendor.NumberSold) )
    SimDay(vendor) #Friday
    if(verbose):
        print(PrintString % ('F', vendor.Q, vendor.MissedSales, vendor.NumberSold) )
    return vendor

vendor = SimWeek([1500,0], verbose = True)#.TotalCost

print(vendor.TotalCost)

ExpectedCost = np.zeros(len(Qrange))
print("         1         2         3         4         5")
print("12345678901234567890123456789012345678901234567890")
for i in Qrange:
    if( (i+1) % 50 ):
        print('.', end='')
    else:
        print('.')
    ExpectedCost[i-Qrange[0]] += np.array([SimWeek([i-150,150]).TotalCost for _ in range(20)]).mean()
## rolling mean
HalfWindow = 20
RolledCost = HalfWindow*ExpectedCost[HalfWindow:-HalfWindow].copy()
for i in range(HalfWindow):
    RolledCost += ExpectedCost[i:i-2*HalfWindow]
    RolledCost += ExpectedCost[HalfWindow+i:i-HalfWindow]
RolledCost /= 3*HalfWindow
plt.plot(Qrange[HalfWindow:-HalfWindow], RolledCost );

print(Qrange[HalfWindow+np.argmin(RolledCost)])
```

A.3 Simulation of Demands of the Food Truck Problem with Mid-week Order

The code listing simulates the demands of the food truck problem for each day and displays the demands for the entire week.

```
#!/usr/bin/env python
# coding: utf-8

import matplotlib.pyplot as plt
import numpy as np
import collections
import simpy as sim
import tqdm

## Defining parameters
c = 3     # cost of food item
p = 1     # profit (excess of cost) of each item
h = 0.0   # holding cost in dollars
r = 1     # return trip cost per item
b = c + r # back order cost per unit
K = 1000  # Truck Capacity

CF = (b - c)/(b+h)  # Critical fractile

## Foodtruck
## order quantity Q
## sell items

#@dataclass
class FoodtruckDual(object):

    def __init__(self, environment):
        ##  Q is the order quantity
        self.objective = 0
        self.Demand = 0
        self.env = environment

    def Generate_demand(self, quantity = 1):
        ## inventory descreases by quantity
        ## inventory = 0 ==> generate underage cost
        while True:
            self.Demand += 1

            ## time dependent demand
            t = self.env.now
            Demand = 60.0 #+ 20*np.sin(np.pi*(t/
               DailySimTime -1/2))  ## expected demand
               varies with time
```

```
            interarrival_time = np.random.exponential(60/
               Demand) # 1 min between sales, on average
            yield self.env.timeout(interarrival_time)

## Food Truck open 5 hours per day
DailySimTime = 5*60  #time in minutes

def SimDay(vendor, verbose=False):
    vendor.Demand = 0
    vendor.env = sim.Environment()
    vendor.env.process(vendor.Generate_demand())
    vendor.env.run(until = vendor.env.now + DailySimTime)
    return vendor.Demand

def SimWeek(  verbose=False):
    """First is initial order quantity, and second is the
       midweek order"""
    ## Food Truck open 5 hours per day
    env = sim.Environment()
    TotalDemand = 0

    ## Create an instance of Food truck
    vendor = FoodtruckDual(env)
    TotalDemand += SimDay(vendor) #Monday
    TotalDemand += SimDay(vendor) #Tuesday
    TotalDemand += SimDay(vendor) #Wednesday
    TotalDemand += SimDay(vendor) #Thursday
    TotalDemand += SimDay(vendor) #Friday
    return TotalDemand

print(SimWeek())
```

VITA

DENNIS LARTEY QUAYESAM

Education:	Bsc. Mathematics, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana 2011
	MPhil. Actuarial Science, University of Ghana Accra, Ghana 2016
	M.S. Mathematical Sciences, East Tennessee State University, Johnson City, Tennessee 2021
Professional Experience:	High School Teacher, St. Mary's Senior High, Ghana, 2013–2019
	Teaching Assistant, Methodist University College, Ghana, 2017–2018
	Graduate Assistant, East Tennessee State University Johnson City, Tennessee, 2019–2021