OR7240: Integer and Nonlinear Optimization

Professor Mehdi Behroozi

Project:

1. Robust Inventory Optimization
2. Charlie Card Kiosk Visits

Team Members: Prerit Samria

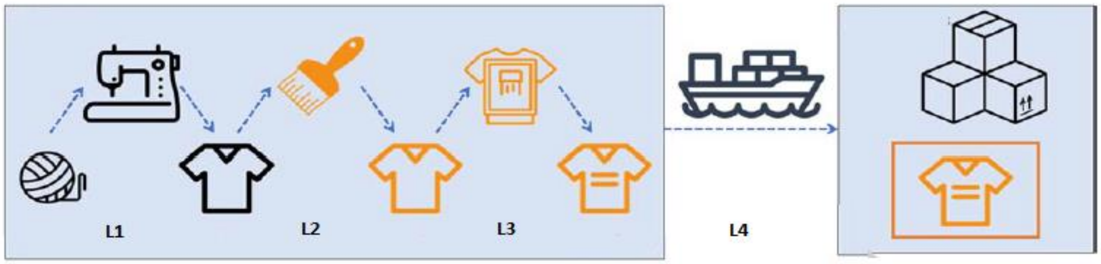
Soumyakant Padhee

1. Robust Inventory Optimization

1.1. Motivation

In the current global scenario, production sites in high wage countries such as the USA are greatly competing with low wage countries. It is considerably cheaper to make products in low wage countries and ship it to demand centers to fulfill global demand. Competition resulting from continuously improving cost pressure and current political norms endanger the remote production sites as more and more companies are moving their production locally. That said, firms are still designing their supply chain network starting from low wage countries and in order to decrease the production costs and counter such trends, these firms combine the strengths of the low wage countries; low manufacturing cost and high-wage country; efficient service delivery network, control, and highly efficient forecasting. Especially, for non-durable goods such as fashion products where demand changes ethereally, even if the ultra-efficient supply chain is reckoned, a lower cost of manufacturing at lower-wage countries provides an edge to firms to increase their margins, which might not be replaced soon. Hence, the focus on robust optimal inventory control is essential to fundamentally reinforce the above strategic supply chain. Due to fluctuating factors; such as a large variety of products, demand asymmetry and change in behavioral taste of customers; and inherent variance in the production lead times, the number of constraints is extremely high before even firms try to fix the level of inventory is optimized. So, we attempt to create a synthetic model to simulate various scenarios of the supply chain and provide bounding policies to govern inventory decisions under various conditions.

We develop a toy model with manufacturing centers in China and demand fulfillment centers in Boston. Conditional on various plausible bounding conditions of the supply chain we try to find the optimal level of inventory. Our firm produces T-shirts in China, then ships them to their warehouse in the USA for distribution to their customers, mostly smaller retailers with shops in high-density urban centers like Boston and New York.



The firm makes shirts of various sizes and types. Each shirt is made from a standard yarn readily available in the region. We have 5 standard sizes, three types (long sleeves, short sleeves, and half sleeves), 5 colors, and 5 caption types providing a shopper menu size of 750 variants both male and female. The shirts undergo multiple manufacturing processes and pass through various job centers. Each job center has its specific gestation period and lead times in between processes. Once the shirts are made, they are packed and shipped to Boston, where it could be stored at a demand delivery center that serves various local retail shops. We build our model as a one-stage delivery model and, we assume production and transit times as deterministic. Despite the fact, the number of constraints makes the model considerably complex. The multi-stage delivery system and adding stochasticity to the model are extensions and merely increase the complexity. That said while determining sensitivity of various variables we consider stochastic transit times, various demand structures and more such things.

Firstly, we present the basic model; objective, constraints, and then elucidate our method of finding optimal solutions. Secondly, we consider various plausible cases to test the robustness of the optimal solution. Lastly, we try to state the optimal policy to plan inventory decisions.

1.2. Model

Objective function: $\max margin$

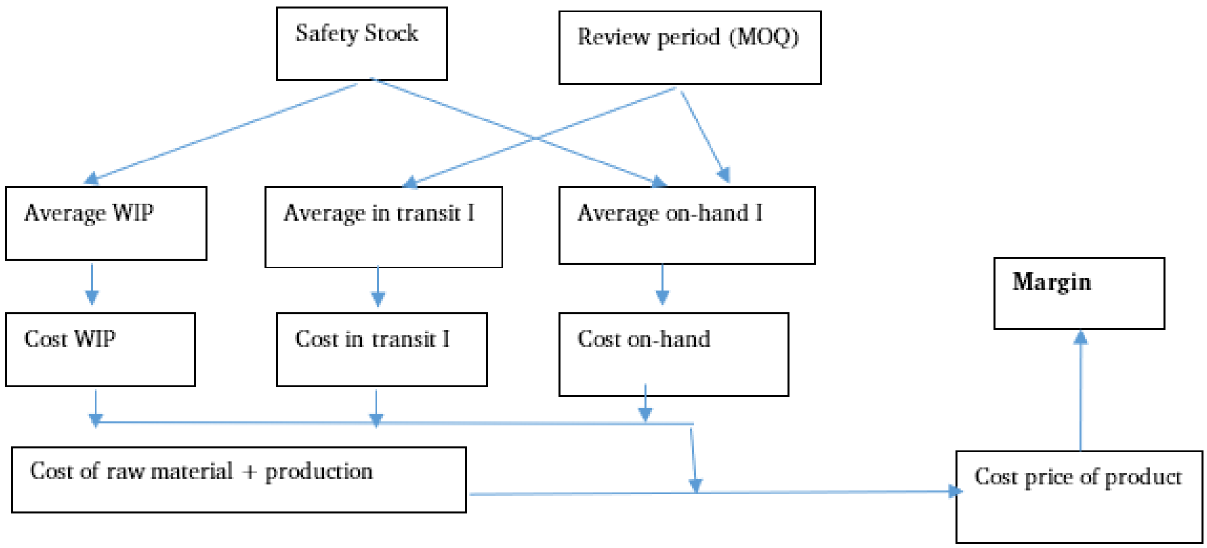
Constraints: $Replenishment\ lead\ times,$
$Various\ inventory\ costs, transit\ cost$
$demand\ distribution.$
$Shirts \in \boldsymbol{Z}^{+}$

Decision variables: $fill\ rate, Review\ period\ \ - T$

The objective function is to maximize the profit margin. The total cost of each shirt has three components; manufacturing costs, inventory cost, transit cost, and freight charges. We assume the selling price of shirts to be deterministic and uniformly distributed depending on product variants. Even though manufacturing costs, transit costs, and import transactions sum up to a major portion of the total cost, we assume that these costs are already standardized, and the process flow of the production is optimized to achievable limits. As per the choice of products we have in our model, the demands of shirts are highly ethereal and hence, the inventory cost of maintaining a large inventory makes up a large portion of the cost. We presume that the type of industry that we are considering makes the assumption reasonable.

There are inventory control policies that differ in terms of the type of review, order quantity limits, type of service level and others. In terms of the type of review, even if the storage and sorting facility is technologically developed in high wage countries, given that the unit cost of the product under consideration being very small makes the periodic review a better choice than the continuous review. The transit options available for transferring finished products are cost-sensitive, with marine transport option much cheaper than premium air freights. However, the high lead time for the marine transport option decreases the flexibility of the repeated changes in the order quantity. Therefore, among (Q, R), (S, T), (S, s, T) models we choose to use (S, T) model where S represents a safety stock and T represents the review period. The inventory is checked at the start of every review period and if the level of inventory is less than safety stock, an order is placed with minimum order quantity, otherwise, no order is placed. Since, there is a practical limit to the number of shipments that can be made in a month, comparing the cost of transit, the marginal cost of production, the ordering and order processing cost is minimal. We assume that the impact of ordering cost is amortized in the marginal cost hence, we do not consider ordering cost as a component of the total cost.

The complete calculation of objective function adheres to the following schematic.



The choice of safety stock level impacts the Average Work in progress and Average on-hand inventory, and the review period determined by the minimum order quantity impacts the average transit inventory and the required amount of on-hand inventory level. Once various levels of inventory are calculated, the respective costs are computed, and the final objective value of the margin is found with the help of the cost of inputs and selling price of the products.

Before proceeding further, we want to define the standardized loss function which we use profusely while developing the model.

$$Total\ cost\ at\ the\ end\ of\ a\ period\ with\ the\ order\ quantity\ of\ Q = T(Q)$$

$$T(Q) = c_0 \max((Q - D, 0) + c_u Max(0, D - Q)$$

where $c_0 = cost\ of\ average, c_u = cost\ of\ underage, D = demand\ in\ the\ period.$

$$E_D[T(Q)] = \int_0^\infty T(Q)\, f(x) dx = c_0 \int_0^Q (Q - x) f(x)\, dx + c_u \int_Q^\infty (x - Q) f(x) dx$$

$$E_D[T(Q)] = c_0[z + L(z)]\sigma_D + c_u L(z)\sigma_D$$

where, $L(z)$ is the loss function to estimate standard deviation z.

We estimate $L(z)$ with various numerical methods explained in (Claudia Sikorski, 05/2016) however, the percentage change in values of L(z) is so minute that the effect on the final margin is insignificant irrespective of whichever numerical method we use. Hence, we use the simplest approximation which is as follows:

$$z - value = 4.85 - L(z)^{1.3} \cdot 0.3924 - L(z)^{0.135} \cdot 5.359$$

Moving ahead with the calculation, firstly, the MOQ level is used to calculate the review period. The safety level is calculated using the distribution of demand, which in turn is used to compute the average on-hand inventory. Depending on the service level, intrinsic lead time of systems, the average in-transit inventory and average work on progress is calculated. Once, levels of inventory are calculated the cost of each inventory is computed, which in addition to 18% of the opportunity cost of money and selling price, gives the margin of sales.

Parameters	**Equation/Explanation**
T	$= MOQ \div \mu_D$, rounded up to the next integer.
Replenishment Lead Time	μ_L = Cut & Sew Time + Dye Time + Embellish Time + Transit Time
Mean demand - DLTR	$= \mu_D * (\mu_L + T)$, where μ_D is average weekly demand
Std. dev demand - DLTR	$= \sigma_D * \sqrt{(\mu_L + T)}$, where σ_D is standard deviation of weekly demand
S	$= \mu_{DLTR} + z * \sigma_{DLTR}$, where $z = L^{-1}\left((1-\beta)\frac{\mu_D * T}{\sigma_D * \sqrt{(\mu_L + T)}}\right)$
Average WIP	= μ_D * (Cut & Sew Time + Dye Time + Embellish Time) For total average WIP, sum over all products.
Average In-transit inventory	= μ_D * Transit Time to Boston. For total average in-transit inventory, sum over all products.
Average On-hand inventory	$= \left[\frac{\mu_D * T}{2} + (L(z) + z) * \sigma_{DLTR}\right] + z * \sigma_{DLTR}$, which is average inventory plus safety stock. For total average on-hand inventory, sum over all products.
Cost of WIP inventory	The cost of WIP inventory is the sum of the costs of WIP for the stages cut & sew, dye, and embellish. These stage WIP values are calculated by multiplying the cost of raw materials and production up to that stage by the average WIP at that stage. For total cost of WIP inventory, sum over all products.
Cost of In-transit Inventory	= (Raw Material Cost + Production Cost) * Average In-transit Inventory, where Production Cost = Cut & Sew Cost + Dye Cost + Embellish Cost For total cost of in-transit inventory, sum over all products.
Cost of On-hand Inventory	= Cost of a Single Shirt * Average On-hand Inventory

	For total cost of on-hand inventory, sum over all products.
Working Capital	= Average On-hand Inventory + Average In-transit Inventory + Average WIP
Financial Opportunity Cost	= 18% * (Cost of On-hand Inventory + Cost of In-transit Inventory + Cost of WIP Inventory)
Average Storage Cost	= Storage Cost * Average On-hand Inventory
Annual Inventory Carrying Cost	= Average Storage Cost + Financial Opportunity Cost
Cost of a Single Shirt	= Raw Material Cost + Production Cost + Freight Cost + Tariff Cost, where Tariff Cost = Tariff Rate * (Raw Material Cost + Production Cost + Freight Cost)
Annual Cost of Goods Sold	= Annual Demand * Cost of a Single Shirt, where Annual Demand = $52 * \mu_D$ For total annual cost of goods sold, sum over all products.
Annual Revenue	= Annual Demand * Selling Price of a Single Shirt For total annual revenue, sum over all products.
Adjusted Gross Margin	= (Total Annual Revenue – Annual Cost of Goods Sold – Annual Inventory Carrying Costs) / Total Annual Revenue
Inventory Turns	= (Annual Cost of Goods Sold) / (Annual Cost of On-hand Inventory + Annual Cost of In-transit Inventory + Annual Cost of WIP)

1.2.1. Computation of optimal margin

Before we optimize the service level and MOQ to calculate the value of objective function -sales margin, we try to plot the change in margin % with respect to the service level across different MOQ level. As you can notice from Figure 3: Margin % vs Service level below, the maximum value of margin is non-concave and Euclidian space of MOQ, service level and the margin is not a convex set. This happens due to the nonlinearity of the standardized loss function used in safety stock computation. Moreover, it can be observed that the margin monotonously decreases as the service level increases. This is because to provide a higher level of service and lower stock-outs, a higher level of inventory has to be maintained. Secondly, since the order is a bulk order, the transit rate is proportional to the volume of the order, the number of turn-around time increases decreasing the margin. To validate this, we split the transit cost into two parts; a fixed ordering cost and a variable transit cost depending on the volume of the cargo. Eventually, it is observed that the fixed ordering cost only offsets the profit margin, but the trend of the plots does not change. Thus, verifying higher inflated costs due to a higher level of inventory. Due to the non-convexity of the feasible space, we fix the minimum level of service and solve for the optimum. Although we might get a local minimum, it is robust to the fluctuation in demand distributions and change in lead time.

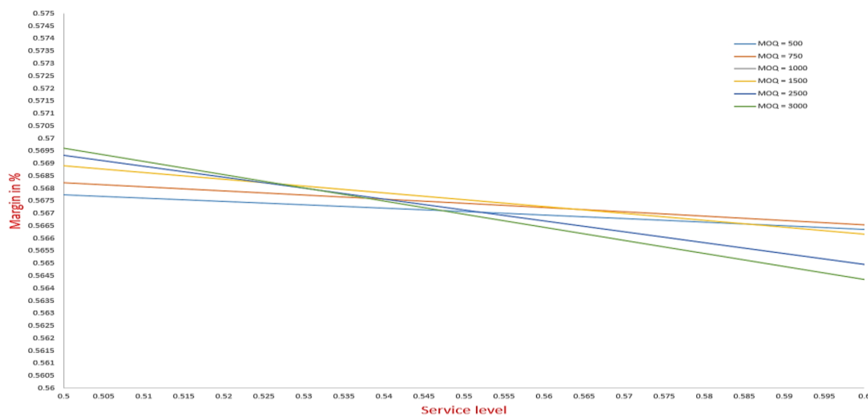


Figure 1 Part of Figure III

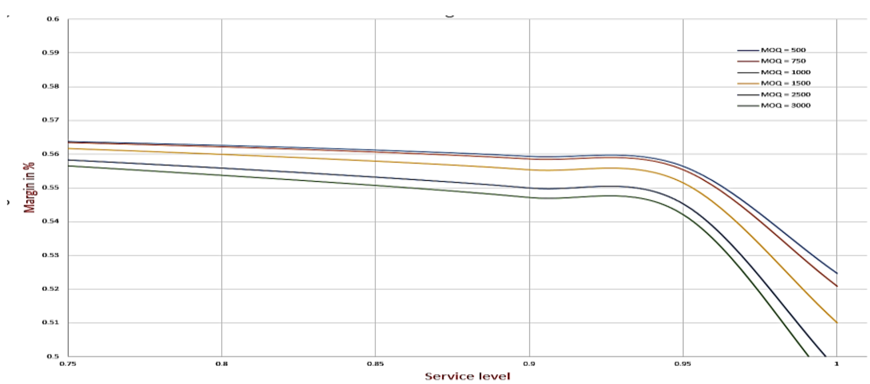


Figure 2 tail end of figure III

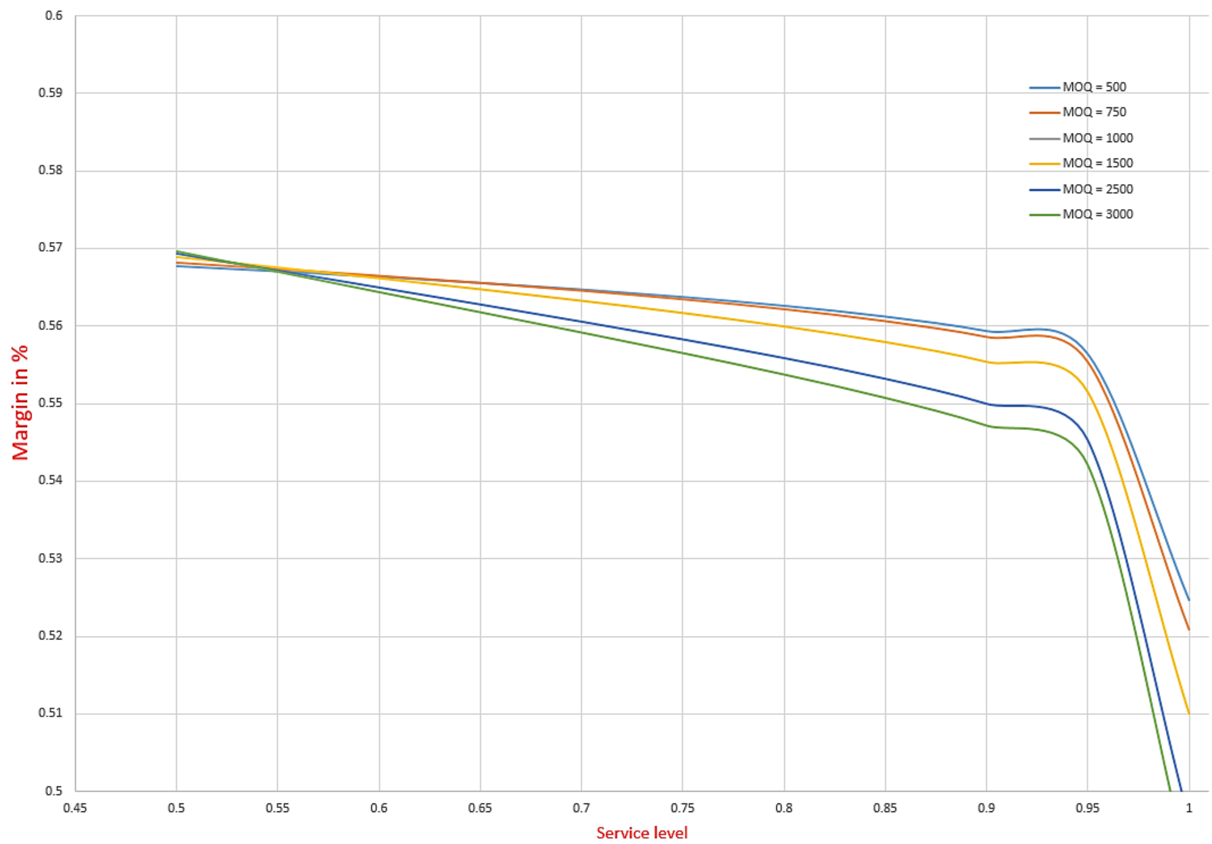


Figure 3: Margin % vs Service level

1.3. Sensitivity Analysis

1.3.1. Demand distribution

The optimal value of the solution depends on the demand distribution. We simulate a random demand synthetically and maximize the margin. The simulated demand has an underlying base normal distribution along with seasonality, positive trend and random shocks which represent promotional events and festive buying season. We have three types of mean and variance; high, medium, low. Once normalized, the variants of products with median sizes have higher mean and lower variance, the variants at the tail end of sizes, the mean and variance are low. Furthermore, we introduce a uniform distribution of mean across various colors of each variant, which we term as medium mean and variance.

To test the robustness of our solution, we increase the variance of underlying normal distribution and compute for optimum.

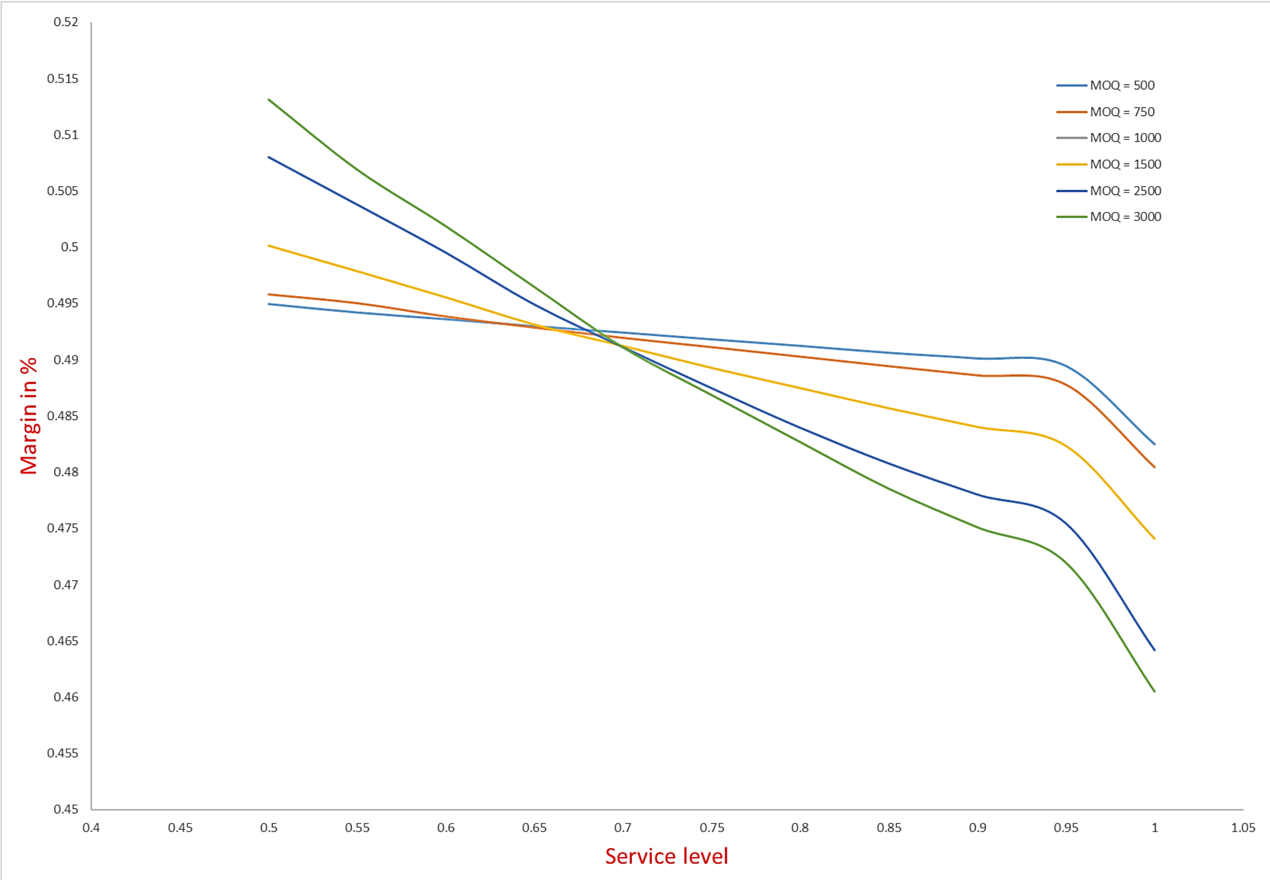


Figure 4 Higher variance

1.3.2. Stochastic Lead times and transit time, and change in overage and underage cost

When we introduce stochasticity in the lead and transit time, the Std. dev of demand during lead time changes as follows.

$$Std.dev\ demand\ -\ DLTR = \sqrt{\mu_L \sigma_D^2 + \mu_D \sigma_L^2 + \sigma_D^2 T}$$

We assume σ_L^2- variance of the lead time as 7 days/ 1 week and compute for the optimal.

The proportion of cost that inventory control contribute to the total cost, is an integral part of the one stage inventory planning problem. A slight reduction in the difference between selling price and cost price make the margin very sensitive to the service level. As a result, the slope of the plot decreases sharply as the service level increase. Since, now when a considerable proportion of total cost comes from the expense due to inventory, an increase in service level decrease the margin very sharply. However, the general trend of the curve still remains nonconvex and similar to the demand sensitivity.

Introducing stochasticity in the model, especially in the transit and lead times has similar effect, and additionally the margin drops.

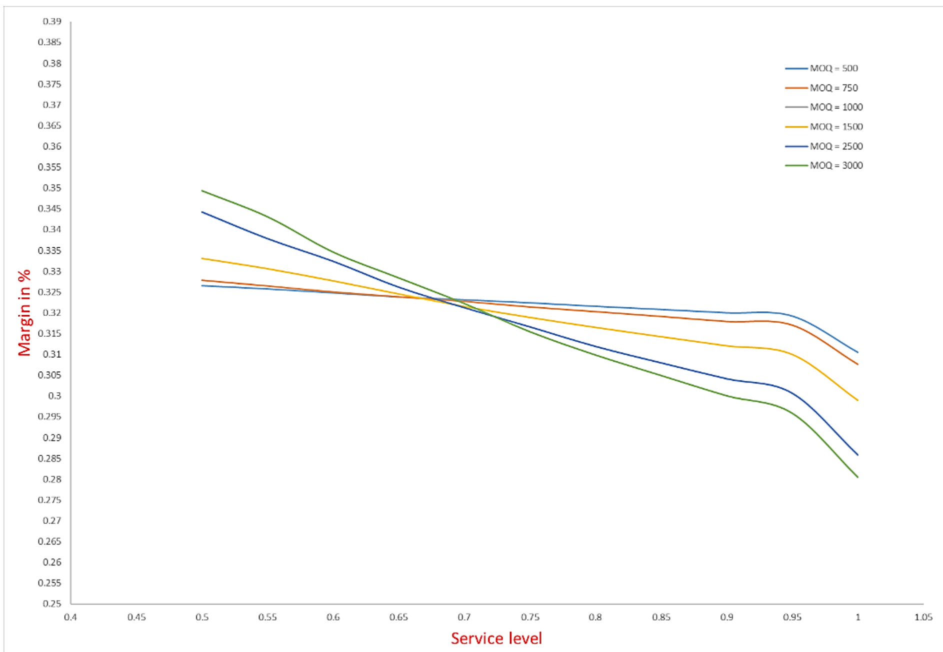


Figure 5 decreasing the difference between overage and underage cost

1.4. Conclusion

Modeling a one stage stochastic inventory model where longer transit time reduces the flexibility of reacting to demand shock is complex. However, the characteristic nonconvexity that is intrinsic to the model, determines the zone of operation. Once decision makers decide on the range of service level that they want to operate, then a choice of minimum order quantity and service level can be determined. Additionally, the monotonicity of margin function serves as a bounding characteristic for MOQ and service level.

1.5. Appendix

Service level in inventory systems are generally defined in two ways across literature. α, β

$$\alpha = probability\ of\ no\ stock\ out = 1 - \int_{s}^{\infty} f(x)\,dx = F(s)$$

$$\beta = propertion\ of\ order\ filled\ from\ stock = 1 - \frac{L(z)\sigma_{DLTR}}{\mu_D T},$$

$$where\ s\ is\ the\ safety\ stock\ level$$

$L(z)$= is the standardized loss function

$$\beta = 1 - \frac{L(z)\sigma_{DLTR}}{\mu_D T\ (\ average\ demand\ per\ cycle)}$$

$$= 1 - \frac{\sigma_{DLTR}}{\mu_D T} L\left(\frac{s - \mu_{DLTR}}{\sigma_{DLTR}}\right)$$

$s = \mu_{DLTR} + z * \sigma_{DLTR}$, where $z = L^{-1}\left((1-\beta)\frac{\mu_D * T}{\sigma_D * \sqrt{(\mu_L + T)}}\right)$

We use the above formulation for calculating the safety stock based on service level β and review period T is determined by

$$T = \left\lceil \frac{MOQ}{\mu_D} \right\rceil$$

1.6. Bibliography

1. Claudia Sikorski, A. A. (05/2016). 1. Numerical Approximation of the Inverse Standardized Loss Function for Inventory Control Subject to Uncertain Demand. *Canadian Operations Research Society Conference 2016*.

2. Charlie Card Kiosk Visits

2.1. Motivation

The vending machines Kiosks at station allow us to load the following pre-specified amounts onto your Charlie card: $\$10, \$20, \$40 \,\&\, \$50$. Given, an exponential distribution of arrival time of a person, although the probability that person will miss the train when she is stuck in Kiosk Queue to reload her card remains the same, but from a behavioral perspective, it always seems like you missed the train because of the Kiosk Queue. Thus, the goal is to minimize the number of visits to the Kiosk machine to load Charlie card with money for N trips.

Let the kiosk offer a menu of refill values: Menu, $M = \{1{:}\,\$10, 2{:}\,\$20, 3{:}\,\$40, 4{:}\,\$50\}$

Let each ride in the subway cost \$2.40.

Let after the end of each month you want to keep a residual amount of money, R in the card.

2.2. Goal

Given the menu of refill values available to subway riders, what is the minimum number of visits to vending machine Kiosks?

Further, we want to determine given the distribution of types of riders in terms of different values of $N \text{ and } R$ (robustness test), what menu of refill values will decrease the number of visits to the Kiosk and decrease congestion for travelers.

2.3. Model

Decision Variables: xi denotes the number of times one visits a Kiosk machine to refill on i^{th} value from the menu M.

It is an integer linear programming problem with linear constraints.

$$Min\; x_1 + x_2 + x_3 + x_4$$

Subject to

$$10x_1 + 20x_2 + 40x_3 + 50x_4 \geq 2.40N$$

$$10x_1 + 20x_2 + 40x_3 + 50x_4 - 2.40N \geq R$$

$$x_i \;=\; non - negative\ Integer$$

2.4. Solution

The integer programming problem is solved using the *Intlinprog* in the optimization toolbox in MATLAB. The integrality gap between the integer programming problem and linear program relaxation is 10^{-5}. The MATLAB code is in the appendix.

Values of N is changed from 1 to 120 with increments of 5 and values of R is changed from $0.25 to $100 with increments of $5.

N/R	0.25	5.25	10.25	15.25	20.25	25.25	30.25	35.25	40.25	45.25	50.25	55.25	60.25	65.25	70.25	75.25	80.25	85.25	90.25	95.25
1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2
6	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	3	3
11	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3
16	1	1	1	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3
21	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3
26	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	4	4
31	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	4	4	4	4
36	2	2	2	3	3	3	3	3	3	3	3	3	3	4	4	4	4	4	4	4
41	2	3	3	3	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4	4
46	3	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4	4	4	5	5
51	3	3	3	3	3	3	4	4	4	4	4	4	4	4	4	4	5	5	5	5
56	3	3	3	3	4	4	4	4	4	4	4	4	4	4	5	5	5	5	5	5
61	3	4	4	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5	5
66	4	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5	5	5	6
71	4	4	4	4	4	4	5	5	5	5	5	5	5	5	5	5	6	6	6	6
76	4	4	4	4	5	5	5	5	5	5	5	5	5	5	6	6	6	6	6	6
81	4	4	5	5	5	5	5	5	5	5	5	5	6	6	6	6	6	6	6	6
86	5	5	5	5	5	5	5	5	5	6	6	6	6	6	6	6	6	6	6	7
91	5	5	5	5	5	5	5	6	6	6	6	6	6	6	6	6	6	7	7	7
96	5	5	5	5	6	6	6	6	6	6	6	6	6	6	7	7	7	7	7	7
101	5	5	6	6	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7
106	6	6	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7	7	7
111	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7	7	7	8	8	8
116	6	6	6	6	6	7	7	7	7	7	7	7	7	7	7	8	8	8	8	8
121	6	6	7	7	7	7	7	7	7	7	7	7	8	8	8	8	8	8	8	8

Figure-6: Optimal number of visits for different values of N and R

Each cell of the matrix represents the optimal number of visits for different values of $N\ and\ R$.

For a user with N = 121 and R = $10.25, optimal solution is: $x_1 = 0,\ x_2 = 0,\ x_3 = 0,\ x_4 = 7$. Thus, optimal number of visits = 7.

For a user with N = 11 and R = $10.25, optimal solution is: $x_1 = 0,\ x_2 = 0,\ x_3 = 0,\ x_4 = 1$. Thus, optimal number of visits = 1.

2.5. Sensitivity Analysis

Changing the refill values in the menu M.

1. Menu, $M = \{1: \$10, 2: \$20, 3: \$30, 4: \$40\}$

Model:

$$Min\ x_1 + x_2 + x_3 + x_4$$

Subject to

$$10x_1 + 20x_2 + 30x_3 + 40x_4 \geq 2.40N$$
$$10x_1 + 20x_2 + 30x_3 + 40x_4 - 2.40N \geq R$$
$$x_i = non-negative\ Integer$$

Solution:

N/R	0.25	5.25	10.25	15.25	20.25	25.25	30.25	35.25	40.25	45.25	50.25	55.25	60.25	65.25	70.25	75.25	80.25	85.25	90.25	95.25
1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	3	3	3	3
6	1	1	1	1	1	1	2	2	2	2	2	2	2	2	3	3	3	3	3	3
11	1	1	1	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	4
16	1	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	4	4	4
21	2	2	2	2	2	2	3	3	3	3	3	3	3	3	4	4	4	4	4	4
26	2	2	2	2	3	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4
31	2	2	3	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4	5	5
36	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4	5	5	5	5	5
41	3	3	3	3	3	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5
46	3	3	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5	6	6
51	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5	6	6	6	6
56	4	4	4	4	4	4	5	5	5	5	5	5	5	5	6	6	6	6	6	6
61	4	4	4	5	5	5	5	5	5	5	5	6	6	6	6	6	6	6	6	7
66	4	5	5	5	5	5	5	5	5	6	6	6	6	6	6	6	6	7	7	7
71	5	5	5	5	5	5	6	6	6	6	6	6	6	6	7	7	7	7	7	7
76	5	5	5	5	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7
81	5	5	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7	8	8
86	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7	8	8	8	8	8
91	6	6	6	6	6	7	7	7	7	7	7	7	7	8	8	8	8	8	8	8
96	6	6	7	7	7	7	7	7	7	7	8	8	8	8	8	8	8	8	9	9
101	7	7	7	7	7	7	7	7	8	8	8	8	8	8	8	8	9	9	9	9
106	7	7	7	7	7	7	8	8	8	8	8	8	8	8	9	9	9	9	9	9
111	7	7	7	8	8	8	8	8	8	8	8	9	9	9	9	9	9	9	9	10
116	7	8	8	8	8	8	8	8	8	9	9	9	9	9	9	9	9	10	10	10
121	8	8	8	8	8	8	9	9	9	9	9	9	9	9	10	10	10	10	10	10

Figure-7: Optimal number of visits to the kiosk for Menu, M: ={1:$10, 2:$20, 3:$30, 4:$40}

2. Menu, $M = \{1: \$10, 2: \$20, 3: \$25, 4: \$30\}$

Model:

$$Min\ x_1 + x_2 + x_3 + x_4$$

Subject to

$$10x_1 + 20x_2 + 25x_3 + 30x_4 \geq 2.40N$$
$$10x_1 + 20x_2 + 25x_3 + 30x_4 - 2.40N \geq R$$
$$x_i = non-negative\ Integer$$

Solution:

N/R	0.25	5.25	10.25	15.25	20.25	25.25	30.25	35.25	40.25	45.25	50.25	55.25	60.25	65.25	70.25	75.25	80.25	85.25	90.25	95.25
1	1	1	1	1	1	1	2	2	2	2	2	2	3	3	3	3	3	3	4	4
6	1	1	1	1	2	2	2	2	2	2	3	3	3	3	3	3	4	4	4	4
11	1	2	2	2	2	2	2	3	3	3	3	3	3	4	4	4	4	4	4	5
16	2	2	2	2	2	3	3	3	3	3	3	4	4	4	4	4	4	5	5	5
21	2	2	3	3	3	3	3	3	4	4	4	4	4	4	5	5	5	5	5	5
26	3	3	3	3	3	3	4	4	4	4	4	4	5	5	5	5	5	5	6	6
31	3	3	3	3	4	4	4	4	4	4	5	5	5	5	5	5	6	6	6	6
36	3	4	4	4	4	4	4	5	5	5	5	5	5	6	6	6	6	6	6	7
41	4	4	4	4	4	5	5	5	5	5	5	6	6	6	6	6	6	7	7	7
46	4	4	5	5	5	5	5	5	6	6	6	6	6	6	7	7	7	7	7	7
51	5	5	5	5	5	5	6	6	6	6	6	6	7	7	7	7	7	7	8	8
56	5	5	5	5	6	6	6	6	6	6	7	7	7	7	7	7	8	8	8	8
61	5	6	6	6	6	6	6	7	7	7	7	7	7	8	8	8	8	8	8	9
66	6	6	6	6	6	7	7	7	7	7	7	8	8	8	8	8	8	9	9	9
71	6	6	7	7	7	7	7	7	8	8	8	8	8	8	9	9	9	9	9	9
76	7	7	7	7	7	7	8	8	8	8	8	8	9	9	9	9	9	9	10	10
81	7	7	7	7	8	8	8	8	8	8	9	9	9	9	9	9	10	10	10	10
86	7	8	8	8	8	8	8	9	9	9	9	9	9	10	10	10	10	10	10	11
91	8	8	8	8	8	9	9	9	9	9	9	10	10	10	10	10	10	11	11	11
96	8	8	9	9	9	9	9	9	10	10	10	10	10	10	11	11	11	11	11	11
101	9	9	9	9	9	9	10	10	10	10	10	10	11	11	11	11	11	11	12	12
106	9	9	9	9	10	10	10	10	10	10	11	11	11	11	11	11	12	12	12	12
111	9	10	10	10	10	10	10	11	11	11	11	11	11	12	12	12	12	12	12	13
116	10	10	10	10	10	11	11	11	11	11	11	12	12	12	12	12	12	13	13	13
121	10	10	11	11	11	11	11	11	12	12	12	12	12	12	13	13	13	13	13	13

Figure-8: Optimal number of visits to the kiosk for Menu, M: ={1:$10, 2:$20, 3:$25, 4:$30}

3. Menu, $M = \{1: \$5, 2: \$10, 3: \$20, 4: \$25\}$

Model:

$$Min\ x_1 + x_2 + x_3 + x_4$$

Subject to

$$5x_1 + 10x_2 + 20x_3 + 25x_4 \geq 2.40N$$
$$5x_1 + 10x_2 + 20x_3 + 25x_4 - 2.40N \geq R$$

x_i = non-negative Integer

Solution:

N/R	0.25	5.25	10.25	15.25	20.25	25.25	30.25	35.25	40.25	45.25	50.25	55.25	60.25	65.25	70.25	75.25	80.25	85.25	90.25	95.25
1	1	1	1	1	1	2	2	2	2	2	3	3	3	3	3	4	4	4	4	4
6	1	1	1	2	2	2	2	2	3	3	3	3	3	4	4	4	4	4	5	5
11	2	2	2	2	2	3	3	3	3	3	4	4	4	4	4	5	5	5	5	5
16	2	2	2	3	3	3	3	3	4	4	4	4	4	5	5	5	5	5	6	6
21	3	3	3	3	3	4	4	4	4	4	5	5	5	5	5	6	6	6	6	6
26	3	3	3	4	4	4	4	4	5	5	5	5	5	6	6	6	6	6	7	7
31	3	4	4	4	4	4	5	5	5	5	5	6	6	6	6	6	7	7	7	7
36	4	4	4	5	5	5	5	5	6	6	6	6	6	7	7	7	7	7	8	8
41	4	5	5	5	5	5	6	6	6	6	6	7	7	7	7	7	8	8	8	8
46	5	5	5	6	6	6	6	6	7	7	7	7	7	8	8	8	8	8	9	9
51	5	6	6	6	6	6	7	7	7	7	7	8	8	8	8	8	9	9	9	9
56	6	6	6	6	7	7	7	7	7	8	8	8	8	8	9	9	9	9	9	10
61	6	7	7	7	7	7	8	8	8	8	8	9	9	9	9	9	10	10	10	10
66	7	7	7	7	8	8	8	8	8	9	9	9	9	9	10	10	10	10	10	11
71	7	8	8	8	8	8	9	9	9	9	9	10	10	10	10	10	11	11	11	11
76	8	8	8	8	9	9	9	9	9	10	10	10	10	10	11	11	11	11	11	12
81	8	8	9	9	9	9	9	10	10	10	10	10	11	11	11	11	11	12	12	12
86	9	9	9	9	10	10	10	10	10	11	11	11	11	11	12	12	12	12	12	13
91	9	9	10	10	10	10	10	11	11	11	11	11	12	12	12	12	12	13	13	13
96	10	10	10	10	11	11	11	11	11	12	12	12	12	12	13	13	13	13	13	14
101	10	10	11	11	11	11	11	12	12	12	12	12	13	13	13	13	13	14	14	14
106	11	11	11	11	11	12	12	12	12	12	13	13	13	13	13	14	14	14	14	14
111	11	11	12	12	12	12	12	13	13	13	13	13	14	14	14	14	14	15	15	15
116	12	12	12	12	12	13	13	13	13	13	14	14	14	14	14	15	15	15	15	15
121	12	12	13	13	13	13	13	14	14	14	14	14	15	15	15	15	15	16	16	16

Figure-9: Optimal number of visits to the kiosk for Menu, M: ={1:$5, 2:$10, 3:$20, 4:$25}

2.6. Observations

As N and R increases, the optimal number of visits to the kiosk also increases. The optimal solution for an individual with a high value of N and a low value of R is to refill using the highest refill value.

If you want to take more trips (high N) and have more residual money (high R), then you have to refill your card with more money. If this is the case, then it is optimal to refill using the higher values of the refill amount on the menu.

Table-1: Value of decision variables for highest number of visits

Menu, M	Highest Number of Visits	x_1	x_2	x_3	x_4
{1:$10, 2:$20, 3:$40, 4:$50}	8	0	0	0	8
{1:$10, 2:$20, 3:$30, 4:$40}	10	0	0	0	10
{1:$10, 2:$20, 3:$25, 4:$30}	13	0	0	0	13
{1:$5, 2:$10, 3:$20, 4:$25}	16	0	0	0	16

In general, the optimal solution tends to favor to refill using higher values to minimize the number of visits to the kiosk. As menu M changes, the optimal number of visits change. The result from the sensitivity analysis is logical since if you want to refill using smaller amounts and take the same number of trips N, then you have to refill the card more.

However, in general subway users do not refill using higher refill amounts. Thus, we can put an upper bound for decision variables x_3 and x_4 to be 5. Now the optimal number of visits to the kiosk would change.

1. Menu, M={1:$10, 2:$20, 3:$40, 4:$50}

Model:

$$Min\ x_1 + x_2 + x_3 + x_4$$

Subject to

$$10x_1 + 20x_2 + 40x_3 + 50x_4 \geq 2.40N$$

$$10x_1 + 20x_2 + 40x_3 + 50x_4 - 2.40N \geq R$$

x_i = non-negative Integer

$x_3, x_4 <= 5$

Solution:

N/R	0.25	5.25	10.25	15.25	20.25	25.25	30.25	35.25	40.25	45.25	50.25	55.25	60.25	65.25	70.25	75.25	80.25	85.25	90.25	95.25
1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2
6	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	3	3
11	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3
16	1	1	1	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3
21	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3
26	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	4	4
31	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	4	4	4	4
36	2	2	2	3	3	3	3	3	3	3	3	3	3	4	4	4	4	4	4	4
41	2	3	3	3	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4	4
46	3	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4	4	4	5	5
51	3	3	3	3	3	3	4	4	4	4	4	4	4	4	4	4	5	5	5	5
56	3	3	3	3	4	4	4	4	4	4	4	4	4	4	5	5	5	5	5	5
61	3	4	4	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5	5
66	4	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5	5	5	6
71	4	4	4	4	4	4	5	5	5	5	5	5	5	5	5	5	6	6	6	6
76	4	4	4	4	5	5	5	5	5	5	5	5	5	5	6	6	6	6	6	6
81	4	4	5	5	5	5	5	5	5	5	5	5	6	6	6	6	6	6	6	6
86	5	5	5	5	5	5	5	5	5	6	6	6	6	6	6	6	6	7	7	7
91	5	5	5	5	5	5	5	6	6	6	6	6	6	6	6	7	7	7	7	7
96	5	5	5	5	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7
101	5	5	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7	8	8
106	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7	8	8	8	8
111	6	6	6	6	6	7	7	7	7	7	7	7	7	8	8	8	8	8	8	8
116	6	6	6	7	7	7	7	7	7	7	7	8	8	8	8	8	8	8	8	9
121	7	7	7	7	7	7	7	7	8	8	8	8	8	8	8	8	9	9	9	9

Figure-10: Optimal number of visits to the kiosk for Menu, M: ={1:$10, 2:$20, 3:$40, 4:$50}

2. Menu, $M = \{1{:}\ \$5, 2{:}\ \$10, 3{:}\ \$20, 4{:}\ \$25\}$

Model:

$$Min\ x_1 + x_2 + x_3 + x_4$$

Subject to

$$5x_1 + 10x_2 + 20x_3 + 25x_4 \geq 2.40N$$

$$5x_1 + 10x_2 + 20x_3 + 25x_4 - 2.40N \geq R$$

x_i = non-negative Integer

$x_3,\ x_4 <= 5$

Solution:

N/R	0.25	5.25	10.25	15.25	20.25	25.25	30.25	35.25	40.25	45.25	50.25	55.25	60.25	65.25	70.25	75.25	80.25	85.25	90.25	95.25
1	1	1	1	1	1	2	2	2	2	2	3	3	3	3	3	4	4	4	4	4
6	1	1	1	2	2	2	2	2	3	3	3	3	3	4	4	4	4	4	5	5
11	2	2	2	2	2	3	3	3	3	3	4	4	4	4	4	5	5	5	5	5
16	2	2	2	3	3	3	3	3	4	4	4	4	4	5	5	5	5	5	6	6
21	3	3	3	3	3	4	4	4	4	4	5	5	5	5	5	6	6	6	6	7
26	3	3	3	4	4	4	4	4	5	5	5	5	5	6	6	6	6	7	7	7
31	3	4	4	4	4	4	5	5	5	5	5	6	6	6	6	7	7	7	7	8
36	4	4	4	5	5	5	5	5	6	6	6	6	7	7	7	7	8	8	8	8
41	4	5	5	5	5	5	6	6	6	6	7	7	7	7	8	8	8	8	9	9
46	5	5	5	6	6	6	6	7	7	7	7	8	8	8	8	9	9	9	9	10
51	5	6	6	6	6	7	7	7	7	8	8	8	8	9	9	9	9	10	10	10
56	6	6	6	7	7	7	7	8	8	8	8	9	9	9	9	10	10	10	10	11
61	7	7	7	7	8	8	8	8	9	9	9	9	10	10	10	10	11	11	12	12
66	7	7	8	8	8	8	9	9	9	9	10	10	10	10	11	11	12	12	13	13
71	8	8	8	9	9	9	9	10	10	10	10	11	11	12	12	13	13	14	14	15
76	8	9	9	9	9	10	10	10	10	11	11	12	12	13	13	14	14	15	15	16
81	9	9	9	10	10	10	10	11	11	12	12	13	13	14	14	15	15	16	16	17
86	10	10	10	10	11	11	12	12	13	13	14	14	15	15	16	16	17	17	18	18
91	10	10	11	11	12	12	13	13	14	14	15	15	16	16	17	17	18	18	19	19
96	11	12	12	13	13	14	14	15	15	16	16	17	17	18	18	19	19	20	20	21
101	12	13	13	14	14	15	15	16	16	17	17	18	18	19	19	20	20	21	21	22
106	13	14	14	15	15	16	16	17	17	18	18	19	19	20	20	21	21	22	22	23
111	15	15	16	16	17	17	18	18	19	19	20	20	21	21	22	22	23	23	24	24
116	16	16	17	17	18	18	19	19	20	20	21	21	22	22	23	23	24	24	25	25
121	17	18	18	19	19	20	20	21	21	22	22	23	23	24	24	25	25	26	26	27

Figure-11: Optimal number of visits to the kiosk for Menu, M: ={1:$5, 2:$10, 3:$20, 4:$25}

Table-2: Value of decision variables for highest number of visits

Menu, M	Highest Number of Visits	x_1	x_2	x_3	x_4
{1:$10, 2:$20, 3:$40, 4:$50}	9	0	0	4	5
{1:$10, 2:$20, 3:$30, 4:$40}	12	0	2	5	5
{1:$10, 2:$20, 3:$25, 4:$30}	16	0	6	5	5
{1:$5, 2:$10, 3:$20, 4:$25}	27	0	17	5	5

This is a more realistic solution since subway users generally recharge with smaller refill amounts.

This model gives the optimal solution for 1 user. Considering the number of daily users of the subway system in a metropolitan city like Boston, the congestion on the kiosks is very high.

2.7. Further Work

An application can be designed for smartphones that can help users decide the number of visits to the kiosks required for their personalized menu M and a fixed number of trips N per month. This would help reduce the congestion on the kiosks which is the main objective of this study.

2.8. Appendix

MATLAB Code for x_i = non-negative Integer:

```
charlieCard = optimproblem;

% coefficient of the decision variables in the objective function
coefficients_refill = [1;1;1;1];

% Menu, M
value = [5; 10; 20; 25];

refill_table = table (coefficients_refill, value);


% decision variable: x
refill = optimvar ('refill', size (refill_table, 1),'Type', 'integer', 'LowerBound', 0);

% min the number of visits
refill_amount = coefficients_refill' * refill;

% to see the number_visits
%showexpr (refill_amount)
```

```
% objective function
charlieCard.Objective = refill_amount;

N = 120; % number of trips
R = 10; % residual amount left

% to calculate number of rows for final table = a
a = size (1:5:N+1);

% to calculate number of columns for final table = b
b = size(0.25:5:100);

% creating output matrix
final_table = zeros (a (2), b(2));

x = 1;
for n = 1:5:N+1

    y = 1;
    r = 0.25;
    while r < 100
        % Constaints
        % 1.
        totalmoney = value' * refill;

        % constraint expression is named 'cons_money'
        cons_money = totalmoney >= 2.40 * n;
        %showconstr (cons_money)

        % include the constaint on money in the problem
        charlieCard.Constraints.cons_money = cons_money;

        % 2.
        money_left = value' * refill;
        % constraint expression is named 'cons_money'
        cons_residual = money_left >= 2.40 * n + r;
        %showconstr (cons_residual)

        % include the constaint on money in the problem
        charlieCard.Constraints.cons_residual = cons_residual;

        % see the optimization problem
        %showproblem (charlieCard)

        % solving the problem
        % sol = optimal variable values
        % fval = optimal objective function value
        [sol, fval] = solve (charlieCard);

        % optimal values
        % Number_of_refills = sol.refill
        final_table (x, y) = fval;

        y = y + 1;
```

```
        r = r + 5;
    end
    x = x + 1;
end

disp final_table;

filename = 'Charlie Card Solution.xlsx';
writematrix(final_table3,filename);
```

MATLAB Code for x_i = non-negative Integer, and x_3, x_4 <= 5:

```
charlieCard = optimproblem;

coefficients_refill = [1;1;1;1];
value = [10; 20; 25; 30];

refill_table = table (coefficients_refill, value);


% decision variable: x
refill = optimvar ('refill', size (refill_table, 1),'Type', 'integer', 'LowerBound', 0);

% min the number of visits
refill_amount = coefficients_refill' * refill;

% to see the number_visits
%showexpr (refill_amount)

% objective function
charlieCard.Objective = refill_amount;

N = 120; % number of trips
R = 10; % residual amount left

% to calculate number of rows for final table = a
a = size (1:5:N+1);

% to calculate number of columns for final table = b
b = size(0.25:5:100);

% creating output matrix
final_table = zeros (a (2), b(2));

x = 1;
for n = 1:5:N+1

    y = 1;
    r = 0.25;
    while r < 100
        % Constaints
        % 1.
        totalmoney = value' * refill;
```

```
        % constraint expression is named 'cons_money'
        cons_money = totalmoney >= 2.40 * n;
        %showconstr (cons_money)

        % include the constaint on money in the problem
        charlieCard.Constraints.cons_money = cons_money;

        % 2.
        money_left = value' * refill;
        % constraint expression is named 'cons_money'
        cons_residual = money_left >= 2.40 * n + r;
        %showconstr (cons_residual)

        % include the constaint on money in the problem
        charlieCard.Constraints.cons_residual = cons_residual;

        % 3. putting constraint on upper bound for x3 and x4
        coeff_x3 = [0; 0; 1; 0];
        coeff_x4 = [0; 0; 0; 1];

        refill_value_x3 = coeff_x3' * refill;
        refill_value_x4 = coeff_x4' * refill;

        % x3 and x4 <= 5
        cons_x3 = refill_value_x3 <= 5;
        cons_x4 = refill_value_x4 <= 5;

        charlieCard.Constraints.cons_x3 = cons_x3;
        charlieCard.Constraints.cons_x4 = cons_x4;

        % see the optimization problem
        %showproblem (charlieCard)

        % solving the problem
        % sol = optimal variable values
        % fval = optimal objective function value
        [sol, fval] = solve (charlieCard);

        % optimal values
        % Number_of_refills = sol.refill
        final_table (x, y) = fval;

        y = y + 1;
        r = r + 5;
    end
    x = x + 1;
end

% final_table

% filename = 'Charlie Card Solution.xlsx';
% writematrix(final_table,filename)
```