MTA Simulation

Analysis and Results

Melissa Lynch Davis Allen Shadman Quazi

May 23, 2018

1 Introduction

1.1 Problem definition

Accumulation of trash on NYC subway tracks poses a real problem for the MTA, leading to track fires and, consequently, numerous subway delays and interruptions of service in recent years.

While efforts are made to perform routine trash removal from its stations, a 2015 Comptroller's audit found that the MTA was failing to meet its maintenance goals. The audit recommended the development of "a systematic approach for ensuring track beds are cleaned frequently and no track beds remain uncleaned for extended periods."

1.2 Goals

We had several goals for our simulation that we were able to meet. The primary objective was to develop a model for trash accumulation in the NYC subway system, as well as the associated maintenance, cleaning and service concerns. We considered two scenarios for our simulation. We created a simulation that models the current trash removal scheme, using scheduled cleanings; we also modeled an alternative scheme that relies on deploying cleanup crews when a certain amount of trash has accumulated. We then compared these schemes with the aim of eliminating track fires and reducing delays (while keeping estimated budget within reasonable bounds).

It is worth noting that although the model for the scheduled cleanings is intended to represent the MTA's current cleaning system, the audit states that these scheduled cleanings are often not completed. That is, our "baseline" model performs considerably better than the real system. Nevertheless, we still intended to show that a demand-based cleanup scheme would be more optimal.

Our threshold model assumes we have some system of detecting the amount of trash on the tracks in real-time. We imagine a scheme involving installation of camera and computer vision technology, however the full details of such a system are beyond the scope of this project. For our purposes, we assume access to this data.

2 Formal Definition of the Model

We keep track of two different scenarios in our simulation: one where there are periodic cleanings (which we call the baseline simulation), and another where the cleaning crew is dispatched once a certain trash amount has accumulated (called the alt simulation).

The arrivals of trash are the same between the two scenarios. We synchronized the simulation in this way because we want circumstances that are as identically comparable as possible.

2.1 Events

We have four events: event g, which is an arrival of trash on the tracks; event c, which is the occurrence of a scheduled periodic cleaning; event b, in which a fire breaks out at the station in the baseline simulation; and event a, which is a fire breakout at the station in the alternative (demand-based) simulation. So, we define any event e to be:

$$e \in \{g, c, b, a\}$$

There are also four residual clocks for each of the events. We denote the residual clock of event e as:

$$Y_n(e): \text{ Residual clock for event } e$$

The inter-event times are denoted as τ_n. We have τ_{n+1} as the time until the next event. To determine this, we take the minimum of the residual clocks such that $\tau_{n+1} = \min\{Y_n(e)\}$ as the smallest residual clock; that is, the time until the next event.

We also consider a threshold cleaning as a possible event; however, the only time the threshold is exceeded, and therefore a cleaning is triggered, is from a new arrival of trash. In other words, a threshold cleaning is a "special case" of event g, and we do not need to keep an additional residual clock. Because of this, it is not explicitly mentioned in the event list.

2.2 Arrival of trash

We refer to the amount of trash on the tracks at time t as $X(t)$. To distinguish between the baseline and alternate simulations, we use a superscript notation as follows:

$$X^b(t): \text{ Aggregate trash (baseline)}$$

$$X^a(t): \text{ Aggregate trash (alt)}$$

The inter-arrival times of trash are denoted $\{A_m\}$. We also have a ridership rate, R, for each station; R is equivalent to the average number of riders per minute. The inter-arrival times of trash are distributed exponentially with parameter $\lambda(R)$. The rate of the arrivals of trash λ is a function of the ridership rate R. So, we have the following expression:

$$A_m \sim \exp(\lambda(R))$$

The absolute event times are $\{T_n\}$. For the baseline simulation, the amount of trash on the tracks at the time of the n^{th} event is $\widetilde{X}_n^b$. In the demand-based simulation we use $\widetilde{X}_n^a$.

2.3 Arrival of fire

We use a similar notation for the arrivals of fire at a station. Z_n is the time until the next fire breaks out. To distinguish between the baseline and alternate simulations, we use a superscript notation as follows:

Z_n^b : Time until next fire (baseline)

Z_n^a : Time until next fire (alt)

The times until outbreaks of fire are distributed exponentially with parameter $\rho(X_n)$. The rate of fire outbreaks ρ is a function of X_n, the aggregate trash at the time of the n^{th} event. For both the baseline and alternative simulations we have:

$$Z_n^b \sim \exp(\rho(\widetilde{X}_n^b))$$

$$Z_n^a \sim \exp(\rho(\widetilde{X}_n^a))$$

The rate of fire arrivals $\rho(\widetilde{X}_n) = 10^{-8} \cdot \widetilde{X}_n$. We use the following conditional to define Z_n:

$$Z_n = \begin{cases} -\frac{1}{\rho(\widetilde{X}_n)} \cdot \ln(1 - U_n), & \text{if } x > 0 \\ \infty, & \text{otherwise} \end{cases}$$

In considering how to model the outbreak of fire on the tracks (due to trash), we begin by thinking of the problem in a tick-based simulation model paradigm. We consider an arbitrary and fixed short interval of time and propose that the probability that a fire starts in this interval depends exclusively on the amount of trash that has accumulated on the tracks. For a given amount of trash, fire may or may not break out in a given short interval (e.g. trash may blow onto the third rail, or it may not). Beyond a loose conceptualization, we do not further consider the details of how the fire occurs. We assume the probability of fire outbreak in any fixed-length short interval to be the same for a constant amount of trash. For varying trash, this probability changes; more trash means a fire is more likely, so we can conceive the occurrence in such an interval as distributed $\sim$ Bernoulli($\mu(x)$) where the parameter $\mu(x)$ depends on the amount of trash on the tracks, x. We could conceive the number of ticks before the occurrence of fire as a geometric random variable. In the limit as the size of the interval h tends to zero, we can consider a continuous-time model in which the time until fire breaks out is an exponential random variable $\sim \exp(\rho(x))$, where the rate, $\rho(x)$ depends only on the amount of trash x. It is in this fashion that we incorporate the outbreak of fire into our discrete event simulation model. It should be noted that the time until fire breaks out, Z_n, must be recomputed with a new rate each time the level of trash changes, but this is justified by the memoryless property of the exponential distribution.

2.4 Scheduled cleanings

The time until the next scheduled cleaning is C_n. This is a predetermined period depending on the traffic of the station. For example, a heavily trafficked station will be cleaned every three weeks, but a low volume station will only be cleaned every six months.

2.5 Maintenance cost and productivity loss

The total cost of cleaning and maintenance at the time of the n^{th} event is M_n. The total productivity loss due to cleanings at the time of the n^{th} event is P_n. It will cost \$10,000 to dispatch a crew and clean a station. If a cleaning crew needs to be sent due to a fire, it will cost a higher \$30,000. This is including the cost to repair damages due to the fire. Furthermore, the time to clean the station (and therefore, the time the station is closed) is 90 minutes when it is a "normal" cleaning. When there is a crew dispatched due to fire, the time to clean and repair the station is 270 minutes.

Delays happen as a result of station cleanings and fire damage repair. We say G is the total number of commuters affected by a given delay. G is a Poisson distribution with the ridership rate times the length of the delay as a parameter:

$$G \sim \text{Poisson}(R \cdot \text{length of delay})$$

Given that the hourly wage in New York City is \$34, we define the total productivity loss as follows:

$$\sum_{i=1}^{G} 34 \cdot \text{length of delay} \cdot U_i(0,1)$$

2.6 Modeling the cost of productivity loss

In our model, we do not simulate events for each passenger arrival. Instead, we generate only the trash arrivals a rate lower than the ridership rate, R. We could do the former and say a passenger has some probability of dropping trash, and from that, generate trash arrivals. However, this would be inefficient and require the generation of many random variables. In regards to productivity loss, however, what we are interested in is the rate of arrival of people (R), not rate of arrival of trash ($\lambda(R)$). So, only in these instances (a cleaning or fire), when we need to work with people instead of trash, do we generate a value for the number of commuters affected. This value is generated as a Poisson($R \cdot$ length of delay), and we assume the arrivals of these commuters is uniformly distributed over the interval of the delay.

3 Implementation of the Model

3.1 Main loop

To implement our simulation, we used a loop structure typical of a discrete event simulation. We have the event list, and we choose the next event to be the one with the smallest residual clock. The procedure is as follows:

```
1: procedure SIMULATE(endtime)
2:     initialize_simulation()
3:     while (time < endtime) do
4:         next_event ← arg min_{e∈{g,c,b,a}} ({Y_n(e)})
5:         if next_event is g then
6:             ...
7:         else if next_event is c then
8:             ...
9:         else if next_event is b then
10:            ...
11:        else                                          ▷ next_event is a
12:            ...
13:        end if
14:    end while
15: end procedure
```

3.2 Handling event cases

If the next event to occur is a trash arrival, we first set the time elapsed to $Y_n(g)$. We also set $Y_n(c)$ equal to $Y_n(c)$ minus the time elapsed. $Y_n(g)$ is recalculated by generating an exponential random variable based on the trash arrival rate. Next, we increase the simulation time by time elapsed. $X^b(t)$ and $X^a(t)$ are also incremented. We perform a check testing whether or not the trash threshold has been exceeded; if so, then we perform a cleaning in the alternate simulation. Finally, we recalculate Z_n^b and Z_n^a.

If a scheduled cleaning is the next event, we begin by setting the time elapsed to the residual clock of event c, $Y_n(c)$. We update our residual clocks and set $Y_n(g)$ equal to $Y_n(g)$ minus the time elapsed; $Y_n(a)$ is set to $Y_n(a)$ minus the time elapsed. $Y_n(c)$ gets set to the cleaning period. Finally we increase the simulation time by time elapsed and clean the station in the baseline simulation.

When we have a fire at a station in the baseline simulation we need to first store the values of the residual clocks for events a and b in temporary variables. Then we set time elapsed equal to $Y_n(b)$, and subtract the time elapsed from $Y_n(c)$ and $Y_n(g)$; we increase time by time elapsed. Next, we increment the number of fires in the baseline as well as clean and repair the station in the baseline. Now, we check if our temp variables for the residual clocks are equal; if so, this means we have a fire in both stations simultaneously. We increase the number of fires and perform a cleaning and repair in the alt simulation. If the variables are not equal, then we simply decrease $Y_n(a)$ by time elapsed.

Our last case is that a fire breakout in the alternative simulation is the next event. We set the time elapsed equal to $Y_n(a)$ and subtract the time elapsed from the residual clocks for events $g, c,$ and b; we increase the time by time elapsed. Finally, we simply increase the number of fires and perform a cleaning and repair in the alt simulation.

4 Simulation Results and Output Analysis

For a given station, each replication of our simulation simulates one year of operation at that station, simulating in parallel both the baseline periodic scheme and the alternative demand-based scheme. The replication produces simulated values for:

- The annual number of fires
- The annual maintenance cost
- The annual cost of lost productivity due to delays from maintenance and fire repair

for *each* of the baseline and alternative approaches. These six measures constitute our performance metrics. For a given station, we run a large number of replications, and for a given performance metric, the sequence of values produced by these replications constitute a sequence of i.i.d. random variables, with some mean Θ and some variance σ^2. For a given performance metric, we take the sample mean as our estimator of Θ

$$\overline{X} = \sum_{i=1}^{n} \frac{X_i}{n}$$

$$\mathbb{E}[X_i] = \Theta$$

$$\text{Var}[X_i] = \sigma^2$$

.

We employ the Central Limit Theorem to assert that when our number of samples, n, is large,

$$\frac{\sqrt{n}(\overline{X} - \Theta)}{\sigma}$$

is appropriately distributed as a standard normal random variable (it tends toward normal as $n \to \infty$) and so

$$\mathbb{P}[|\overline{X} - \Theta| > \frac{c\sigma}{\sqrt{n}}] \approx 2[1 - \phi(c)]$$

where $\phi(c)$ is the cumulative distribution function of the standard normal. We will present all results as intervals with 95% confidence:

$$95\% \text{ confidence interval: } \overline{X} \pm \frac{1.96S}{\sqrt{n}}$$

where S is our sample standard deviation:

$$S = \sqrt{s^2}$$

$$S^2 = \frac{\sum_{i=1}^{n}(X_i - \overline{X})^2}{n-1}$$

4.1 Choosing parameters for simulation

The parameters for the station were based on the MTA's annual ridership report. Assuming that the heavily trafficked stations received more cleanings annually, we used the categories in the Comptroller's report listed above to find a midpoint for each category. For example, stations that receive 0-3 cleanings make up 30% of the stations, so we used a station in the 15^{th} percentile of train traffic. The number of cleanings our "typical" stations got was simply the average of the number of cleanings listed for each category in the Comptroller's report. For each of the "typical" stations, we run 1500 replications of a year-long simulation, capturing Total Annual Maintenance Cost and Total Annual Productivity Loss. We choose a value for the threshold in the alt simulation that yields a Total Annual Maintenance Cost that is similar to that of the baseline simulation.

4.2 The simulations and stopping criteria

The relative performance of the baseline scheme and the alternative scheme depends on the threshold we choose for the alternative scheme:

1. Equal maintenance: we choose a threshold value that yields similar estimated annual maintenance costs between the baseline and alt schemes and observe whether there is any significant difference in *productivity loss cost*.

2. Equal cost of productivity loss: we choose a threshold value that yields similar costs due to productivity loss between the baseline and alt schemes and observe whether there is any significant difference in *maintenance cost*.

When simulating a station in either of the above scenarios, we calculate sample mean, sample standard deviation, and our confidence intervals for all metrics after each replication. We always perform at least 30 replications. Then, after each subsequent replication, we consider the confidence intervals for each of the baseline and alternative simulations, for the metrics we are interested in. In the equal maintenance scenario, we do not stop running further replications if there is any overlap between the confidence intervals of the productivity loss metric for the baseline and alt simulations. Further, we verify after each replication that the confidence intervals for maintenance cost *do* overlap between the schemes, so that we may continue to be confident that the annual maintenance costs are similar.

We continue running further replications, verifying after each that the maintenance cost intervals overlap, until we have run 1500 replications, at which point we terminate and calculate our final estimator from the sample mean of 1500 and our confidence interval.

We take a similar approach in the equal-cost-of-productivity-loss scenario; but this time verifying that the productivity loss confidence intervals overlap while verifying that the maintenance costs have diverged.

Full results, including annual number of fires in all scenarios and schemes can be found at the end of the report.

4.3 Variance reduction

We used common random numbers as a method of variance reduction for our simulation. The inter-arrival times of trash, A_m, is shared across both schemes (baseline and alt), rather than generated independently for each. For Z_n^a and Z_n^b we generate the exponential random variables $Z_n^{a/b}$ as a function of a uniform random variable, U_n. This U_n is shared across the schemes as the common random number, rather than generating the two exponentials $Z_n^{a/b}$ entirely independently. When considering the difference in in maintenance cost, for example, between the two schemes, we expect that employing these common random numbers would yield reduced variance in the estimate of the difference in maintenance, yielding a better estimator.

5 Further research

Here we present areas of our report that we could extend upon in further research. Instead of constantly generating the exponential $Z_n^{a/b}$ every time X_n changes (which requires a lot of generation of random variables and is thus inefficient), we could take a different approach. If instead of considering *time* until next fire, suppose we consider the *amount of trash* at which the next fire breaks out. We might observe that this amount of trash obeys some distribution and if we could discern what it is, we might be able to generate this amount just once after each fire. This would yield the amount of trash at which the next fire breaks out. This would mean that we generate just one random variable between successive fires, instead of thousands, thus improving efficiency.

6 References

1. Ross, "Introduction to to Probability"
2. Ross, "Simulation"
3. Taylor and Karlin, "Introduction to Stochastic Modeling"
4. "Audit Report on the New York City Transit Authority's Track Cleaning and Painting of The Subway", NYC Comptroller's Office, May 14th 2015
5. "The Economic Cost of Subway Delays", NYC Comptroller's Office, Oct 1st 2017

# Stations	Annual Ridership	# of Track Beds	Threshold	Cleaning Period	Baseline Maintenance	Alt Maintenance	Baseline Prod Loss	Alt Prod Loss
83	1000000	1	500	350400	76840.0 +/- 1571.47	76806.6 +/- 1237.13	91853.3 +/- 2068.27	70540.8 +/- 2150.94
146	2800000	1	600	87600	143860.0 +/- 2137.80	142260.0 +/- 1599.71	381055.1 +/- 7855.16	327414.6 +/- 7599.52
26	7770000	1	1300	50060	230220.0 +/- 2803.91	231853.3 +/- 2207.82	1664538.1 +/- 28572.96	1589360.4 +/- 29329.89
14	13300000	1	1725	36250	300400.0 +/- 3177.40	299306.7 +/- 2447.76	3606764.6 +/- 55335.20	3491853.8 +/- 56028.69
7	25000000	1	2150	26280	429540.0 +/- 3689.50	430053.3 +/- 2751.88	9700143.5 +/- 120851.36	8892975.7 +/- 118173.80
83	1000000	1	500	350400	76840	76806.6	91853.3	70540.8
146	2800000	1	600	87600	143860	142260	381055.1	327414.6
26	7770000	1	1300	50060	230220	231853.3	1664538.1	1589360.4
14	13300000	1	1725	36250	300400	299306.7	3606764.6	3491853.8
7	25000000	1	2150	26280	429540	430053.3	9700143.5	8892975.7
83	1000000	1	500	350400	6377720	6374947.8	7623823.9	5854886.4
146	2800000	1	600	87600	21003560	20769960	55634044.6	47802531.6
26	7770000	1	1300	50060	5985720	6028185.8	43277990.6	41323370.4
14	13300000	1	1725	36250	4205600	4190293.8	50494704.4	48885953.2
7	25000000	1	2150	26280	3006780	3010373.1	67901004.5	62250829.9
				TOTALS:	40579380	40373760.5	224931568	206117571.5