CS166 First Project – Airport Security Queues

Minerva University

CS166 - Modeling and Analysis of Complex Systems

Stênio Alves de Assis

Prof. Tambasco

January 30th , 2024

CS166 First Project – Airport Security Queues

Model Description

The following assignment simulates the queue formation inside of an airport security screening. We will discuss this model's variables, parameters, assumptions, and rules. Although precise metrics are not found in the real world, we must define them here to run the model and compute its results. Therefore, by running the simulation, we better understand how queues are formed inside the airport screening check; in other words, this model does not represent reality, given that we need simplification, but it gives a first direction to understand reality. This simulation was implemented using object-oriented programming (OOP) written with Python and Numpy, Matplotlib, Scipy.stats, and Heapq libraries.

The queue type used is M/G/c, which has an Exponential distribution for the arrival time, General Distribution (Truncated Normal) for the service time, and c number of service stations. This simulation aims to find the optimal number of service stations for a better average waiting time, average queue length, and maximum queue length during a day.

General Simulation

The airport has multiple queues, each with its own service station. Each person arrives at the airport independently. The person joins the shortest queue. The inter-arrival time comes from an Exponential distribution. From a First-In-First-Out (FIFO) approach, the first person will be served first. The service station can only attend one person at a time, while the others must wait in the queue. The service time comes from a Truncated Normal Distribution. Only one senior security officer is randomly requested for an additional screening in a service station. When the security officer is busy, the other stations needing his help must wait until he finishes. The senior security officer's time also comes from a Truncated Normal Distribution. When the service is done, the person leaves the security checkpoint, and the queue moves.

Variables

Variables are the changing values of our simulation. In the airport model, the variables are:

- Number of people in the queue at a specific time
- Each person waiting time
- Time between consecutive arrivals
- Service time

Parameters

Parameters are the values set at the beginning of the simulation. These values do not change during the simulation. The parameters used in the airport model are:

- Arrival rate of the Exponential Distribution of $\lambda = 10$ travelers per minute;

- Mean service time of the Truncated Normal Distribution of $\tau_1 = 0.5$ minutes or 30 seconds
- Variance of service time of the Truncated Normal Distribution of $\sigma_1 =$ 1/6 minutes or 10 seconds
- Number of security stations/queues *c* from 1 to 15
- Number of senior security officer equal to 1
- Probability of additional screening for each passenger is 3%
- Mean time of additional screening $\tau_2 = 2$ minutes, or 120 seconds
- Variance time of additional screening $\sigma_2 = 2$ minutes, or 120 seconds

Assumptions

Assumptions are information held as accurate without needing proof so we have a baseline from where the model can perform. Those assumptions simplify the reality so that the modeling process becomes feasible.

- The airport works 24 hours
- The airport has an infinity maximum capacity
- No preference in the queue (FIFO)
- Inter-arrival comes from an Exponential distribution
- Service time comes from a Truncated Normal Distribution
- Additional screening time comes from a Truncated Normal Distribution
- Each passenger is an independent entity
- Each passenger waits patiently for their turn
- Each passenger always joins the shortest queue
- Each service station only attends one person at a time while the others wait.

Rules

Rules are the set of steps in the simulation process.

1. A new passenger joins the shortest queue.
2. If the queue corresponding to the service station is empty, the passenger goes to the service station; otherwise, he waits for his turn.
3. In the service station, generate a random uniform number between 0 and 1 for the passenger. If below or equal to 0.03, the passenger will need additional screening.
4. If the senior security officer is free, he will immediately attend to the passenger. Otherwise, the queue stops and enters the security officer's priority queue.
5. Once the service is done (with or without additional screening), the passenger leaves the service station, and the next passenger comes to the service station.
6. Repeat 1) and simulate until a specific time.

Theoretical Analysis

The queue type of this simulation is M/G/c, indicating that the interarrival time comes from an exponential distribution, a truncated normal distribution generates the service time, and there are c service stations in the simulations. There are also c queues, each with a station at the end. The performance metrics for this type of queue are still an open problem. Therefore, the metrics used here are approximations found by some authors and approximations from other types of queues. The approximation is not actually precise since no senior security officer is considered in the queue theory, making the actual metrics harder to be computed theoretically. For this assignment, the metrics are computed as recommended by the class notes of A. Gosavi[1]. He assumes a homogeneous case with multiple parallel servers and one single waiting queue. This assumption is valid in our simulation since the arrival distribution for each queue is identical, and each person is only joining the shortest queue, indicating a constant approximated queue length.

Average Queue Length

Queue length is the number of people in the queue at a specific time. The average queue length is the size of the queue when in equilibrium. We compute the average queue length L_q for a queue of type M/M/c as an approximation for the average queue length for M/G/c given the latter requires much higher mathematical and computational operations out of the scope of this assignment.

$$L_q = \frac{P_0\left(\frac{\lambda}{\mu}\right)^c \rho}{c!(1-\rho)^2} \tag{1}$$

where the utilization $\rho = \frac{\lambda}{c\mu}$, where λ is the arrival rate, c the number of queues, μ the service rate. Here $\mu = \frac{1}{\tau_1}$, where τ_1 is the mean service time. We assume $\rho < 1$, otherwise the queue would grow infinitely. P_0 is the probability that there are zero customers in the system. It is denoted as:

$$P_0 = 1/\left[\sum_{m=0}^{c-1} \frac{(c\rho)^m}{m!} + \frac{(c\rho)^c}{c!(1-\rho)}\right] \tag{2}$$

[1] Notes prepared by A. Gosavi, Department of Engineering Management and Systems Engineering, Missouri S & T available at https://web.mst.edu/~gosavia/queuing_formulas.pdf

In Figure 1, we have the plot for equation (1) for values of c ranging from 1 to 15 for the defined parameters of the model.

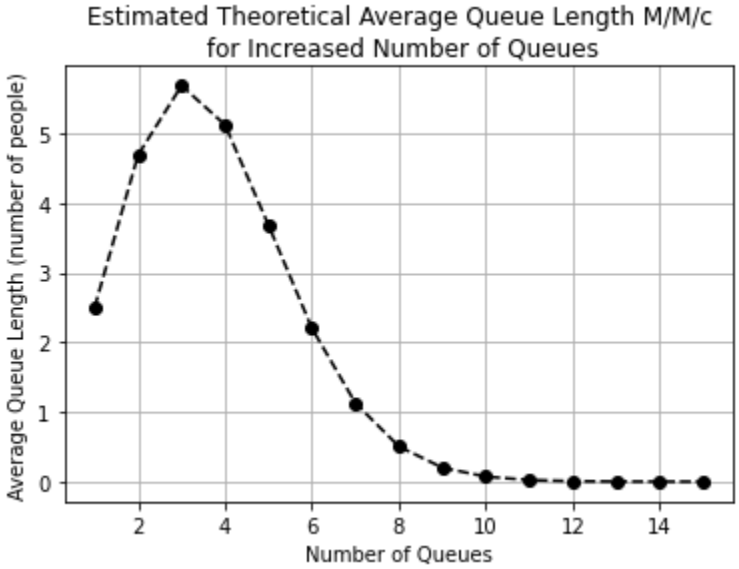

Figure 1: Estimated Theoretical Average Queue Length for M/M/c queue
for an increased number of queues. There is an increase in the queue length up to around 6,
before decreasing to almost zero after ten queues.

We can notice from the plot that the average queue length reaches its maximum, between 5 and 6 passengers, when we have three queues, before decreasing until almost reaching zero at ten queues. After ten queues, there is an equilibrium since increasing the number of queues does not change the metric.

Average Waiting Time

The waiting time is the time between when a customer joins the queue and the time when he gets screened. The average waiting time is computed in the equilibrium. Gosavi also derives a formula for the waiting time $W_q^{G/G/c}$ for a queue G/G/c that seems to describe well the average waiting time of a M/G/c queue.

$$W_q^{M/G/c} \approx W_q^{G/G/c} \approx W_q^{M/M/c} \frac{C_a^2 + C_s^2}{2} \tag{3}$$

Where $W_q^{M/M/c}$ is the average waiting time in a queue M/M/c defined as:

$$W_q^{M/M/c} = L_q/\lambda \tag{4}$$

And C_a^2 and C_s^2 are respectively the squared coefficient of variation of inter-arrival time and the square coefficient of variation of service time, defined as the ratio between the variance and the square of mean.

$$C_a^2 = \frac{(1/\lambda^2)}{(1/\lambda)^2} = 1 \tag{5}$$

$$C_s^2 = \frac{\sigma_1^2}{\tau_1^2} = \frac{1}{9} \tag{6}$$

In Figure 2, we have the average waiting time plotted for an increasing number of queues computed from the above formula.

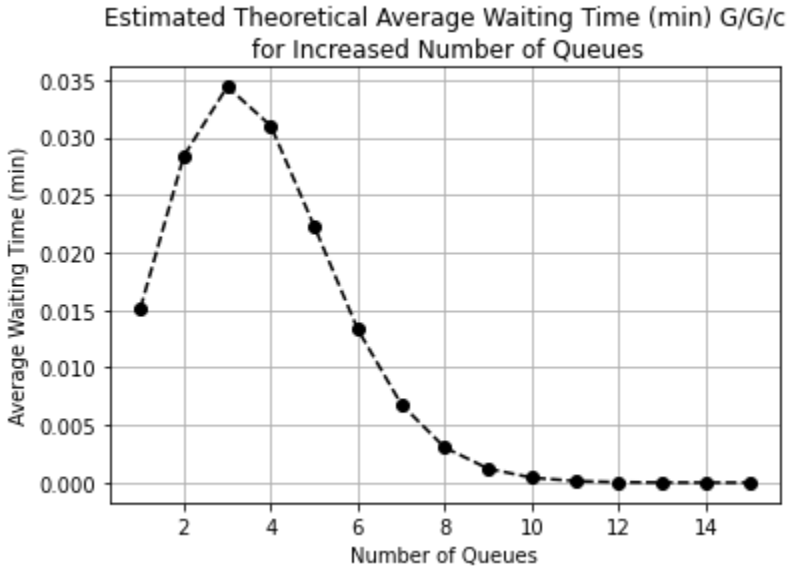

Figure 2: Estimated Theoretical Average Waiting Time (min) for G/G/c queue for an increased number of queues. There is an peak of around 0.035 minutes when there are three queues before decreasing to almost zero after 10 queues.

Average Maximum Queue Length

Unfortunately, up-to-date, there is no theoretical formula for description or approximation of the average maximum queue length of M/G/c queue. However, we will see in our simulation that the maximum average queue length has very similar behavior to the average queue length, explained by the constant growth of the queue length for the given parameters and the presence of a senior security officer who delays the queue. The behavior is also similar when taking the average for increasing the amount of queues. We will, therefore, use the average queue length as a theoretical approximation for the average maximum queue length. Therefore, Figure 2 also shows the expected average maximum queue length for a type G/G/c queue.

Empirical Analysis

Classes

The simulation was written in Python by defining four OOP classes. *Event* class creates each event that will be added to the *Schedule* class, such as adding passengers to the queue, screening passengers, and passengers going through additional screening. In the *Event* class, the function *run* runs the function associated with the *Event* object. *Schedule* class is responsible for implementing a schedule of such events in order of occurrence using a heap. The time for each event comes from probability distributions, as discussed in the Parameters and Assumptions section. The class *Queue* also implements each queue with its functions, such as adding people, starting and finishing service, and starting and finishing additional screening. The main attributes of this class are people in the queue, waiting time, and maximum queue length. Class *Airport* creates objects of the class *Queue*. It implements the overall features of an airport, such as adding passengers and managing the work of the senior security officer. Its main attributes are the *Queue* objects and the arrival distribution.

Data Generation

We are simulating 100 trials for each number of queues from 1 to 15. For each trial, we store the values of waiting time, average queue length, and maximum queue length for all the queues in the airport after 24 hours of simulation timing. Then, we compute the expected value and a 95% confidence interval for each metric. The simulation is set up in minutes. Therefore, the simulation will run until 1440 minutes or one day, after which the simulation stops, the values are computed and stored in lists. The results found are the following.

Average Queue Length

Figure 3 shows a significant decrease in the average queue length for increasing queues/service stations after 24 hours of simulation time. From 12000 customers, when there is only one queue, we decrease sharply to close to zero after implementing eight queues in the

airport. Adding more queues does not significantly decrease the average length since it has reached its minimum. The bars are hard to see, given the narrow confidence interval caused by the large sample size (100 trials).

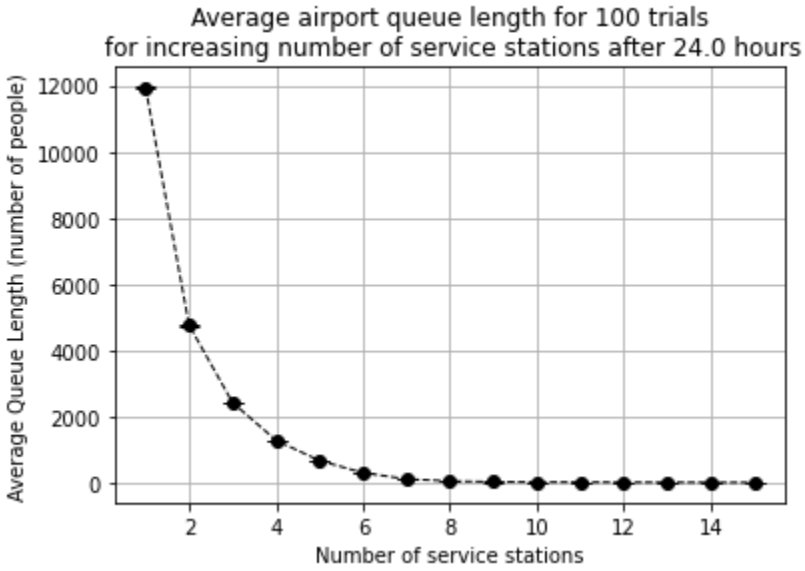

Figure 3: Average Airport Queue Length for 100 trials after 24 hours (simulation time) for an increasing number of service stations. We can notice the quick decrease of the queue length from 12000 when there is only one queue to close 0 when there are 8 queues.

By comparing this plot to the plot of Figure 1, we see how greatly they differ in the range of the y-axis and the peak when the number of queues is three, which is not present in the error plot. Figure 3 plot is an error plot with confidence intervals as bars. However, the exponential decrease behavior is shared by both plots after 4 number of service stations, which correctly models that after eight service stations, the average queue length decreases to below one.

Average Waiting Time

Figure 4 shows that the average waiting time decreases approximately linearly from 600 to 100 after six queues. After that, the model assumes an exponential decrease that stabilizes after ten queues. The small confidence interval also is a reflection of the increased sample size.

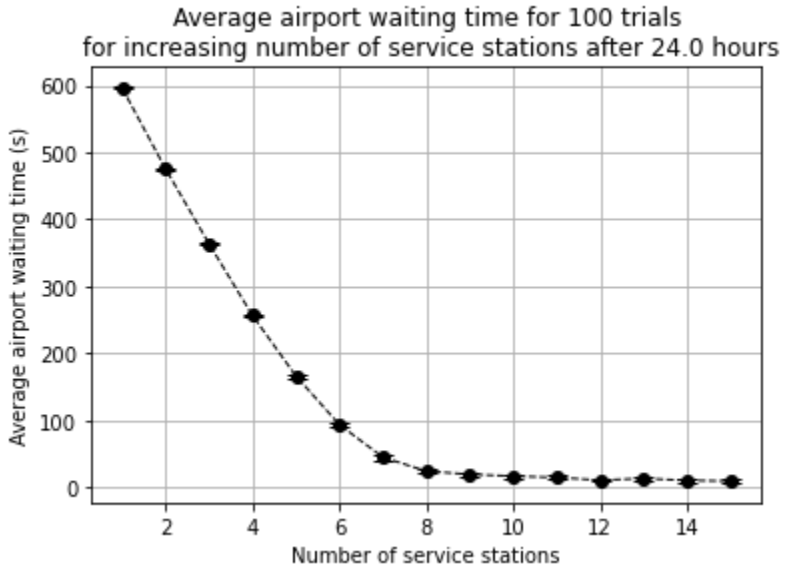

Figure 4: Average Airport Waiting Time for 100 trials after 24 hours (simulation time) for an increasing number of service stations. We can notice a linear decrease until 6 queues, After that the model seems to have an exponential decrease that stabilizes after 10 queues.

Comparing this plot with Figure 2, we can see that the plots also differ significantly in the range of y-values, although we see similar behavior after four queues. Both plots indicate that the equilibrium is reached after ten queues, indicating that more than ten queues do not significantly decrease the average waiting time. The tiny scale of Figure 2, which has values ranging from 0 to 0.035, is worth mentioning.

Average Maximum Queue Length

Here, the average maximum queue length shown in Figure 5 has behavior that is very similar to the average queue length shown in Figure 3. This behavior can be caused by the constant increase of the queue length for increased running time due to the presence of senior security officers given the parameters defined, indicating that the maximum queue length at a particular time is the queue length at the same time since the queue is constantly growing. This indicates that the queue has not reached equilibrium after 24 hours (simulation time) or the equilibrium does not exist for increasing running time. However, we see that maximum queue length also decreases drastically for increasing the number of queues, stabilizing after 9 or 10 queues. Given that there is no precise theoretical formula we can use, we can compare this empirical behavior to the theoretical values of the average queue length given the similar plot behavior.

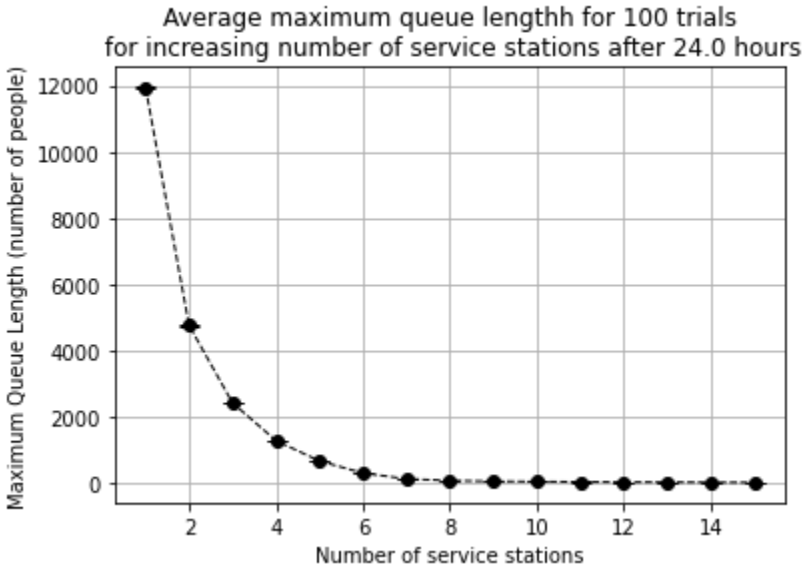

Figure 5: Average Maximum Queue Length for 100 trials after 24 hours (simulation time) for an increasing number of service stations. We can notice the quick decrease of the queue length from 12000 when there is only one queue to close 0 when there are 8 queues. This plot is very similar to Figure 3.

We observe a sharp decrease that is only counted after four queues in the theoretical approach, disregarding the discrepancy in the y-axis range. As we see, both plots stabilize after ten queues.

Conclusion

Given the set assumption and approximations for this simulation, we can infer that the minimum number of queues that drastically decreases the waiting time, queue length, and maximum queue length is 10, as discussed extensively in the previous section. Above this value, those metrics are not impacted, indicating that the system has reached its minimum equilibrium.

One significant constraint of the simulation is its extensive running time due to nested loops and high *running until* simulation parameter value. Further improvements would be a better use of data structures such as heaps instead of queues to store and retrieve values. Here, we use approximations for the theoretical calculations of the metrics, which do not account for the presence of the senior security officer. We are using approximations mainly due to the need for simpler and better formulas for calculating queue performances for queues M/G/c in the mathematical realm. Better predictions would benefit from the theoretical formulation of such metrics.

Word-count: 2544 words
No AI tool but Grammarly was used in this assignment.