Get in Line!

Solving capitalism's greatest problem: long queues

Rhali Attar

CS166: Modeling and Analyzing Complex Systems
Professor Ribeiro
College of Computational Sciences
September 30 2022

Introduction and Motivation

Everyone hates grocery store queues. From a customer's perspective, after spending many long and dreadful minutes picking out delicious food, the last thing they want to see is endless lines keeping them from going home and enjoying their purchase. On the other side of the counter, the minimum wage worker grows increasingly exhausted and the thought of scanning more items becomes as insurmountable as climbing Mount Everest. However, as it would also be unprofitable to hire an unreasonable number of cashiers to deal with all these people quickly, a balance needs to be found to please the customers, the workers and the store owners. In this report, I will present my results for the optimal number of cashiers using a self-made grocery store simulation supported with a theoretical analysis. I will also explore whether including express queues for customers with less than 10 items create a more efficient system. Let's solve this queuing problem together!

Theoretical Analysis

Before exploring the results of the simulations, their validity must be assessed through a thorough theoretical analysis. Indeed, this discussion will not only allow us to make sure that the code works as expected but it will also clarify the assumptions being made.

The kind of model we are exploring in this report is an $M/G/c$ queue. The M stands for Markov as we are assuming that customers will be joining the system with a Poisson process, or in other words, the time between consecutive customers' arrival into the queues is exponentially distributed. For this analysis, we will use a rate parameter $\lambda = 1$ which means that, once a customer joins the system, the next one will arrive 1 minute later on average. Moreover, the G stands for General, as we are assuming that the service rate (the time taken to serve each customer) follows a Gaussian distribution. Here, we will set its mean to $\mu = 3$, and its standard deviation to $\sigma = 1$ which means that it takes 3 minutes to serve each customer on average. So, this model will be stochastic which is appropriate because there is no way to ever determine exactly when customers will arrive and how long it will take to serve them. Lastly, the c represents the number of cashiers or queues that this system has, which will be our independent variable in this study. We will be exploring the efficiency of the system for c in the discrete range $[1, 10]$.

Now the next question that must be addressed is what metric will we use to measure efficiency. In queuing theory, the most common metrics that are used for this are the average customer waiting time (the difference between the time at which they joined a queue and the time when the server finished serving them) and the maximum queue length. Finding these values will be beneficial to analyzing the efficiency of the system because we will know that, the lower their values, the less amount of time customers need to wait. We can predict that as the number of queues increases, these metrics should decrease. Moreover, I will also calculate the number of empty queues on average in my simulation as that will inform us on when the system is so efficient that it becomes inefficient. Indeed, if many queues are empty on average, it is clear that some cashiers never need to serve customers which is a waste of a queue. We can predict that as the number of queues increases, so will this metric. Therefore, the optimal number of queues will be at the intersection of the three metrics as it will be the point where the customers do not have to wait very long but without wasting queues.

On another important note, the average customer waiting time can be calculated theoretically as long as the arrival rate is smaller than the service rate. While we cannot calculate it for $M/G/c$ queues, we can calculate it for $M/D/1$ (exponential arrival rate and deterministic service rate) and $M/M/1$ (exponential arrival and service rates) queues. The reason why this matters is that it will allow us to check that our simulation works and is reliable enough to be extended to the $M/G/c$ case. Let's carry out these calculations:

M/D/1 queue:

For this type of queue, we get the following formula for finding the average customer waiting time when the arrival rate = λ and the service rate = μ and (μ :

$$\text{average waiting time} = \frac{1}{\mu} + \frac{\lambda}{2\mu^2(1 - \frac{\lambda}{\mu})}$$

Let's see to what extent the simulation's results match the theoretical results:

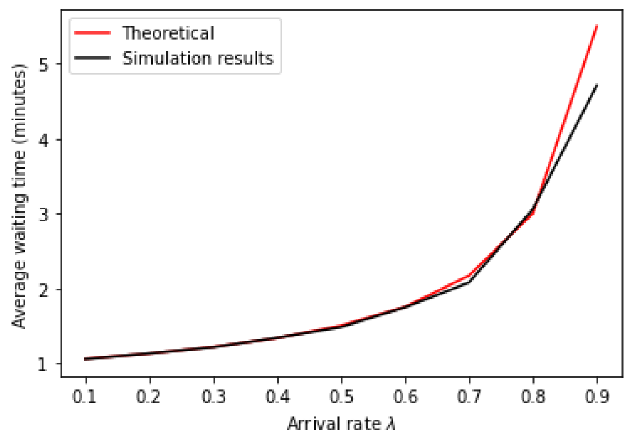

Figure 1: Line graph demonstrating the theoretical and empirical waiting time for an $M/D/1$ queue when varying the arrival rate and keeping the service rate constant at 1

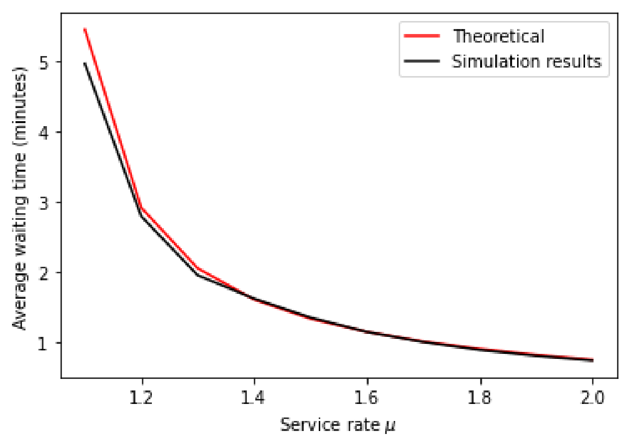

Figure 2: Line graph demonstrating the theoretical and empirical waiting time for an $M/D/1$ queue when varying the service rate and keeping the arrival rate constant at 1

So, it is clear that the simulation can accurately model $M/D/1$ queues.

M/M/1 queue:

Here, we get the following formula for finding the average customer waiting time when the arrival rate = λ and the service rate = μ:

$$\text{average waiting time} = \frac{1}{\mu} + \frac{\lambda}{\mu(\mu - \lambda)}$$

Let's see to what extent the simulation's results match the theoretical results:

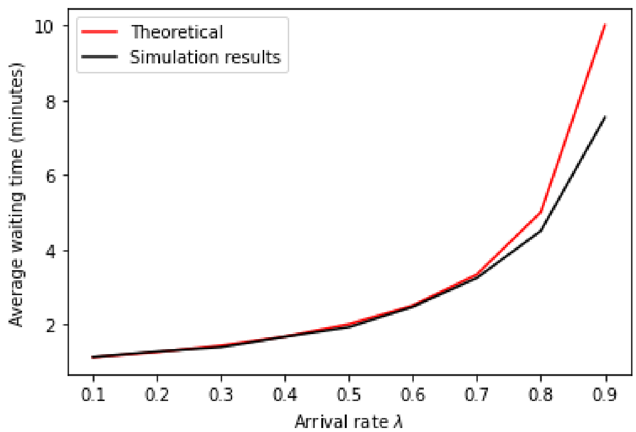

Figure 3: Line graph demonstrating the theoretical and empirical waiting time for an $M/M/1$ queue when varying the arrival rate and keeping the service rate constant at 1

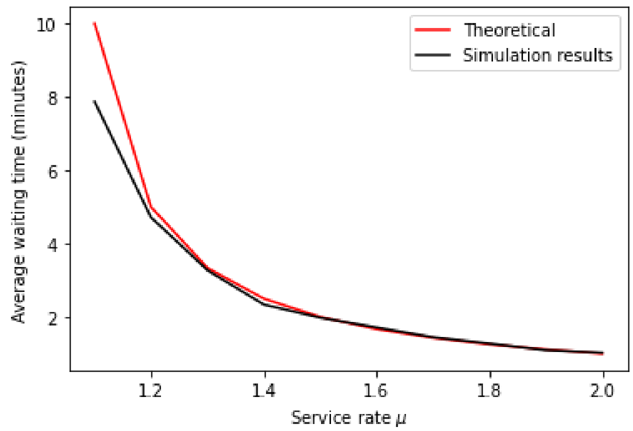

Figure 4: Line graph demonstrating the theoretical and empirical waiting time for an $M/M/1$ queue when varying the service rate and keeping the arrival rate constant at 1

So, it is clear that the simulation can accurately model $M/M/1$ queues.

Results and Discussion

Now that we know that our simulation works, we can starting talking about our results. After running the simulation for 100 9 hour days for various numbers of cashiers, we get the following data:

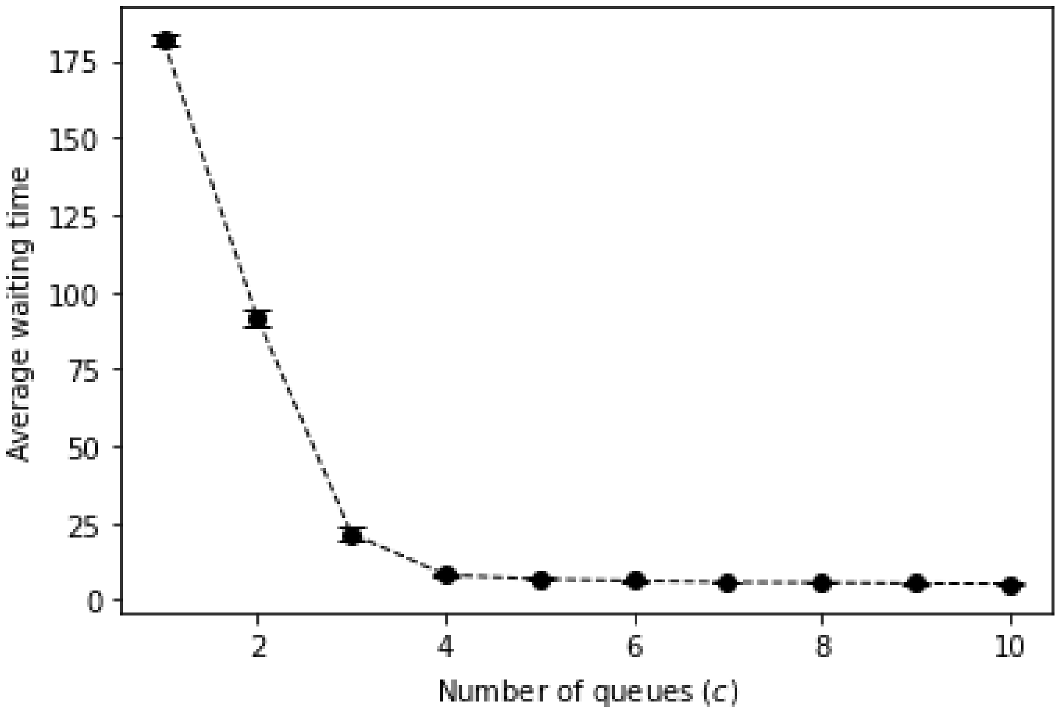

Figure 5: Line graph demonstrating how increasing the number of cashiers decreases the average customer waiting time

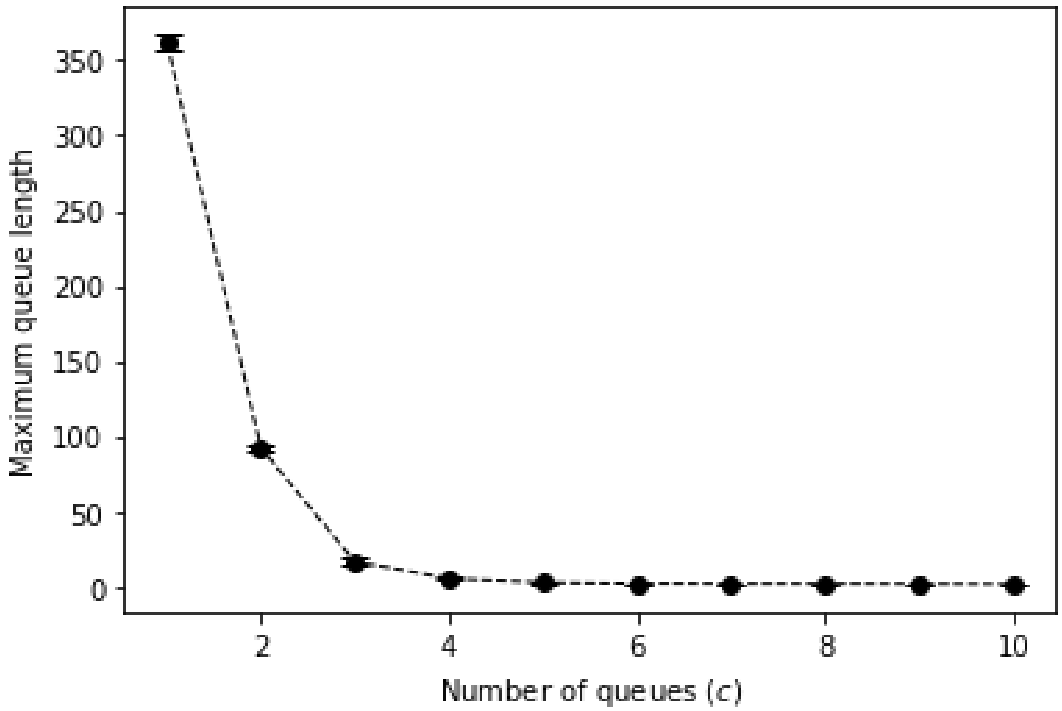

Figure 6: Line graph demonstrating how increasing the number of cashiers decreases the maximum queue length

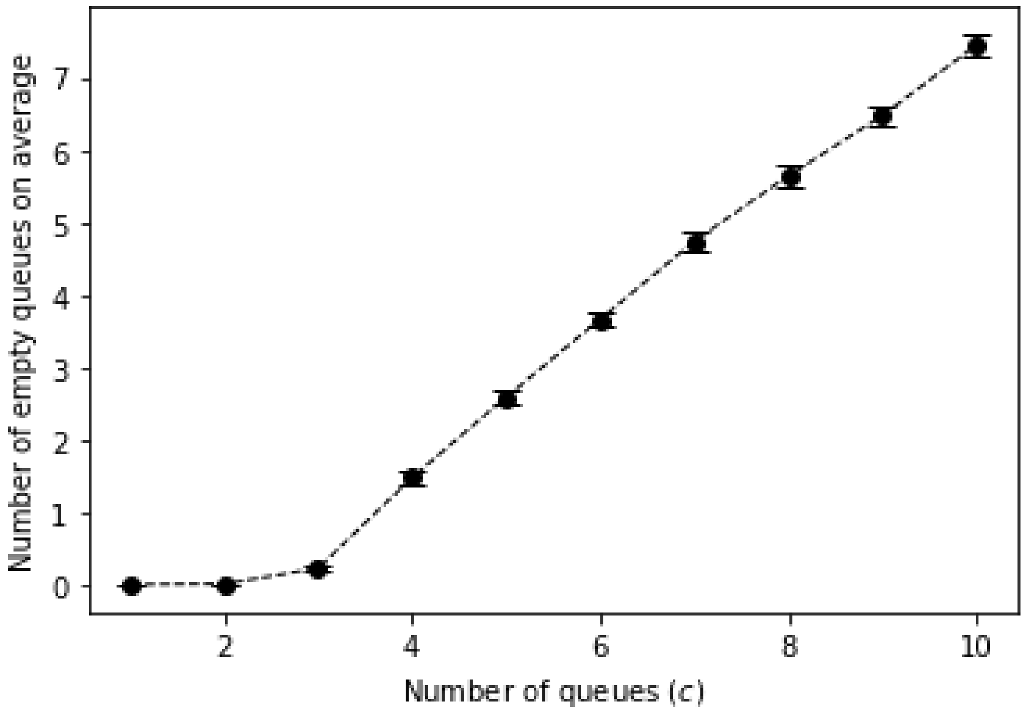

Figure 7: Line graph demonstrating how increasing the number of cashiers increases the number of empty queues on average

As we can see in these results, the 95% confidence intervals for these values are very narrow. Indeed, while the model is stochastic, these metrics should tend towards the same value if each simulation is ran for long enough. Since the simulations were run for 540 minutes, this explains why the outputs were very similar each time. Therefore, while at a first glance this seems worrying, it is actually further evidence that our model works as intended!

From this model, we can conclude that the optimal number of queues is 4. This is because the average waiting time is around 8 minutes which is not too far off from the minimum value of around 5 minutes when there are 10 queues. Moreover, the average queue length is around 4.5 people which, once again, is not too far off from the 1.5 people when there are 10 queues. The system seems more efficient when there are 10 queues as we can see from the previous metrics, however, that is not the case. When there are 10 queues, over 7 of them are empty on average, which is a huge waste, but when there are 4 queues, only around 1.5 of them are empty on average. This number almost doubles when we increase the number of queues to 5 while the average waiting time only decreases by around 1 and the average queue length by 2. Therefore, having 4 cashiers helps significantly decrease the average waiting time to a reasonable value without wasting resources.

Optional section: Extension model

What about the idea of adding express lines for people who purchase less than 10 items though? Well, by slightly modifying the simulation, we can also find the optimal proportion of regular to express lines. Since there are two independent variables (number of express lines and number of regular lines) at play here, it is quite difficult to visualize the way the average waiting time would vary as we change them. However, with some coding magic, we can find which combination of express and regular lines are optimal. I have found that this optimal combination to minimize the weighted average waiting time (two-thirds of customers use the regular cashiers since we assume that the number of items a customer will pick up is uniformly distributed from 1 to 30) is having 9 regular cashiers and 1 express cashier. This will make the average waiting time to be around 3.2 minutes for people using regular cashiers and around 2.9 minutes for people using express cashiers.

An important question to also ask would be when a customer should join the express queue if they have less than 10 items. Indeed, the express queues tend to be longer than regular ones as they have four cashiers, but it takes less time to serve each customer. Well, if we implement the optimal result found above, we can see that the express cashier is actually only about 1.1 times faster. Therefore, if there is a regular line that is shorted than the express line, the customer should use a regular line as it will be faster. However, it is important to note we did not really optimize for people using express cashiers as we give them less importance since there are less of them, instead we optimized to have an average waiting time per customer to be as low as possible.

Word count: 1,350

LO tags

PythonImplementation

In this assignment, I implemented two simulations on Python to model a grocery store and I checked that these simulations worked through theoretical analyses. These simulations followed the requirements from the assignment description and used data structures such as heaps to increase efficiency.

TheoreticalAnalysis

I analyzed this model theoretically as much as possible to check that my simulation was reliable. However, the theoretical analysis was quite lacking since it is not possible to analyze an $M/G/c$ queue theoretically.

EmpiricalAnalysis

I analyzed the results of the simulation to draw an appropriate conclusion as to how many queues is optimal for both the compulsory and optional sections. Additionally, I considered three variables (and explained why they were each important) to help me come to the best possible conclusion.

CodeReadability

I created very clear code by using object oriented programming. Additionally, I made sure to use appropriate and descriptive variable names and thoroughly comment my code. I also added titles in my jupyter notebook to make each section of code clearer.

Professionalism

I used the conventions of the Computer Science field by typing up my assignment in LaTeX so that it could be appropriately formatted. I also made sure to create very clear and well captioned figures when necessary.

Modeling

I created a very thorough model of that can be applied to simulate any type of queue. This was used to analyze queuing in a grocery store and the assumptions underlying this analogy was explained. I also made sure to analyze this model both empirically and theoretically.