CS 4830

Systems Simulation

Professor: Mateen Rizki

Final Project Report

By Matthew Slusser

Introduction

Motivation

For my final project I decided to look at the elevators of the Fritz and Delores Russ Engineering Center. The lobby of Russ has two notoriously slow elevators, which by default appear to operate such that when someone presses the call button on any floor, if one of the elevators is free it travels to the requested floor. After dropping off its rider, the elevator remains idle or parked on that floor until the call button and pressed again and the cycle repeats. For this project, what I wanted to know was if a different idle scheduling strategy was used, could it reduce the average wait time for riders. My goal for the project was to consider three alternative idle scheduling strategies for the elevators and see if any of them were able to produce an average wait time less than the default configuration. After creating the sequence of events for the default configuration I would make three variations to it. One where the elevators idle on the first floor, one where the elevators move sequentially, and one where the elevators idle on the first and fourth floors. After creating the four configurations I'm looking at, I will collect and report some statistics for comparison.

System

System Configurations

The default configuration for the elevator operation is the base class that I used when making this simulation. The default class elevator configurations does not even implement the rider simulation, instead I elected for that to be implemented by extending this default configuration. The default class for this does handle some useful functions that all configurations make use of; namely the selection of the servicing elevator, starting and ending floor selection, and calculation of floor differences for use in travel time calculation. The default elevator works as described above, moving between floors when called and idling where it was last left.

The configuration I call idle one, works the same as the default configuration until the doors close after the rider has gotten off. If there is no queue for the elevator at that time, it will return itself to the first floor. If there is a queue it stays where it is to be later called and moved to its next location.

The sequential configuration had a similar change to its idle configuration, but with an additional change made to elevator selection. In this configuration elevator one always moves up while elevator two always moves down. Additionally, it will send elevator one back down to the first floor if it gets to the third or fourth floor but only if it has no queue and will send elevator two up to the fourth floor if it goes down to the second or first floor, also only if the elevator has no queue.

In my final configuration, one-four, once again it is similar to the default configuration expect the elevators will return to the first and fourth floor if they have no queue.

Assumptions

In creating my simulation, I had to make many assumptions. The assumptions I made that I think had the biggest impact relate to the distribution of the rider's starting and ending floors. For the selection of these two variable, I assumed that; 5% of people start in basement or on the maintenance level, 40% start on the first floor, 15% start on the second floor, 27% start on the third, lastly 13% start on the fourth floor. For ending floor selection, I assumed; 5% of people go to the basement or maintenance level, 45% end on the first floor, 15% end on the second floor, 25% end on the third floor, finally 10% end on the fourth floor. As you will later see in my results, I think these distributions lead to a bias towards the default scheduling configuration.

A less impactful assumption that I made was that no one bawks from the elevator, meaning after the call button was pressed someone is going to ride the elevator. I also assume that each arrival might not be a single person but can be considered a "batch" arrival where more people might show up and get in the elevator someone else called. I did not differentiate riders in such a way that a new rider would get in someone else's elevator, this means only one person riding the elevator is treated the same as a "full" elevator.

Other marginally impactful assumptions I made are that the travel time between floors will remain relatively consistent but is not exact, it is estimated to be a uniformly distributed random variate within a range of 9.5 to 10.5. This range does come from real collected data that was quite uniform but with strict bounds, so this seemed to be the best case for such a bound variable. A similar story is seen for the time it took for the elevator doors to open and close, these had a bit more variation but for the data I was able to collect was also very bounded and uniformly distributed between 3.8 and 5.3 seconds. The last notable assumption I made was to use an exponentially distributed interarrival time with a mean value of 35 seconds. I reasoned that because I was concerned with wait time and not throughput in the system, so long as my interarrival rate did not overly saturate the elevators queue the impact of estimating this value instead of real-world data would be minimal. My final assumption is that I am not concerned with any case where the elevator is broken or otherwise inoperable.

Results

Below is a table of the results of 1000 replicate of my simulation running for 7200 seconds or 2 hours:

Results w/ a mean Interarrival Rate of 35 (s)	**Average Number of Riders / Hour**	**Mean of Mean Wait Times (seconds)**	**Mean of All Wait Times (seconds)**	**Variance of All Wait Times**
Default	102.5	35.81	36.12	50.73
Idle-One	102.7	48.78	49.27	139.11
Sequential	103.0	81.27	82.19	590.97
One-Four	102.8	40.94	41.28	56.31

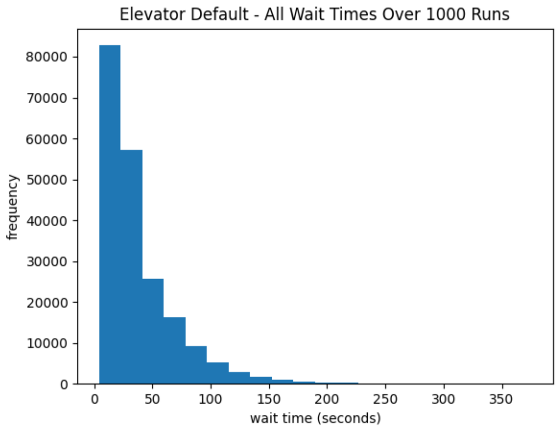
After running all my main tests I decided to also show how the system might handle a different interarrival rate, these were my results from testing an interarrival rate of 25 seconds and 45 seconds.

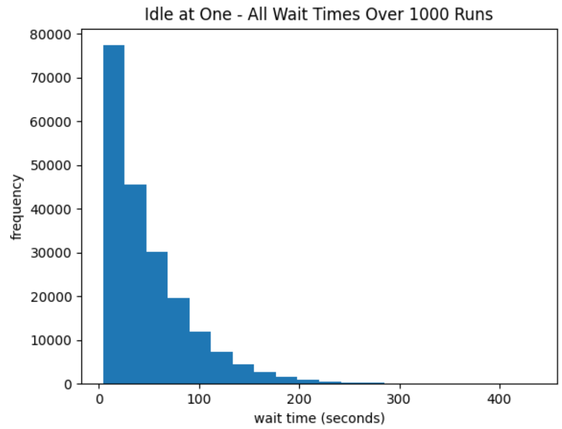
Results w/ a mean Interarrival Rate of 25 (s)	Average Number of Riders / Hour	Mean of Mean Wait Times (seconds)	Mean of All Wait Times (seconds)	Variance of All Wait Times
Default	144.4	106.92	108.69	2,721.83
Idle-One	144.0	220.48	223.00	15,614.38
Sequential	144.0	318.89	321.03	15,284.13
One-Four	144.0	112.19	113.98	2864.60

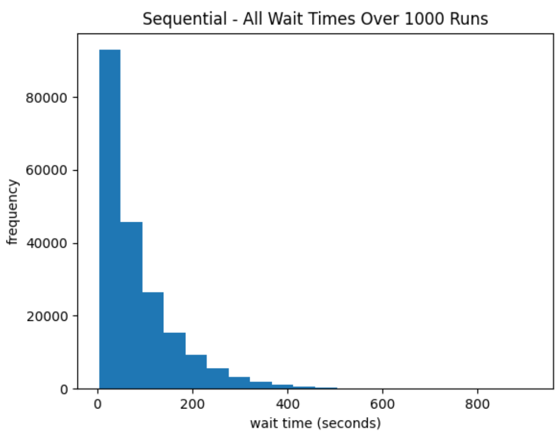
Results w/ a mean Interarrival Rate of 45 (s)	Average Number of Riders / Hour	Mean of Mean Wait Times (seconds)	Mean of All Wait Times (seconds)	Variance of All Wait Times
Default	79.9	25.83	25.98	13.82
Idle-One	79.8	31.92	32.15	25.39
Sequential	80.0	48.69	49.14	107.73
One-Four	79.8	29.80	29.98	17.11

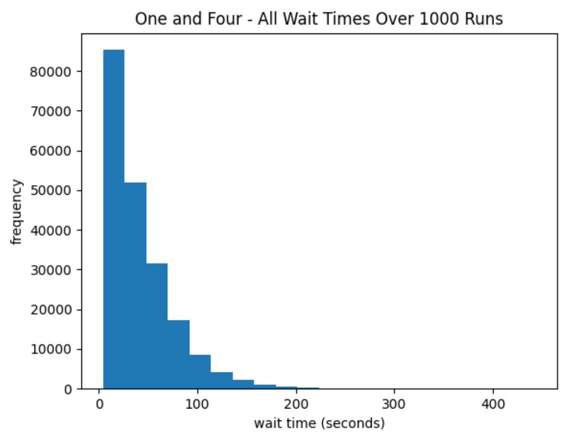
Graphs

Mean Wait Time of All Riders

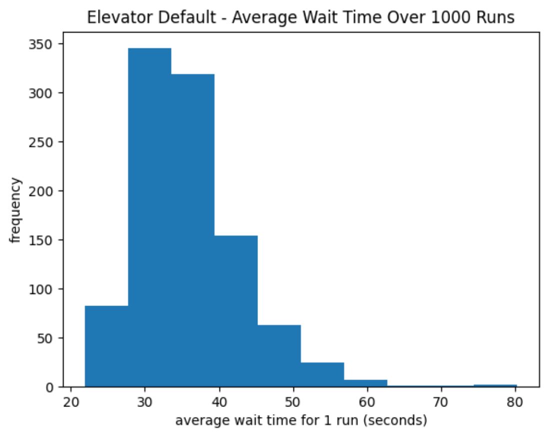


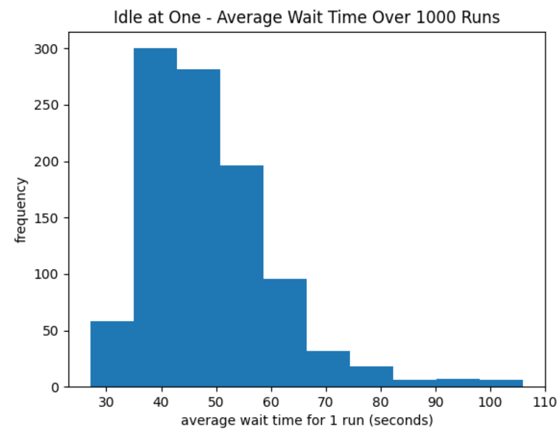


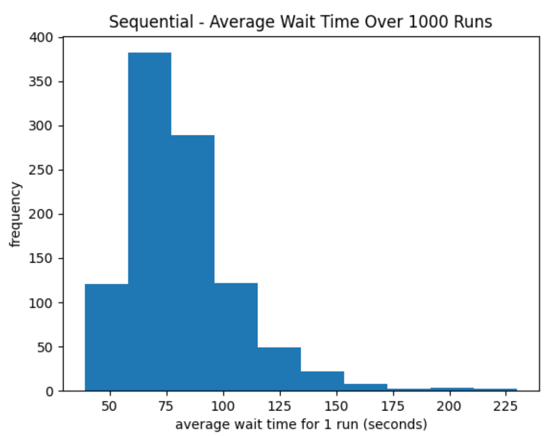


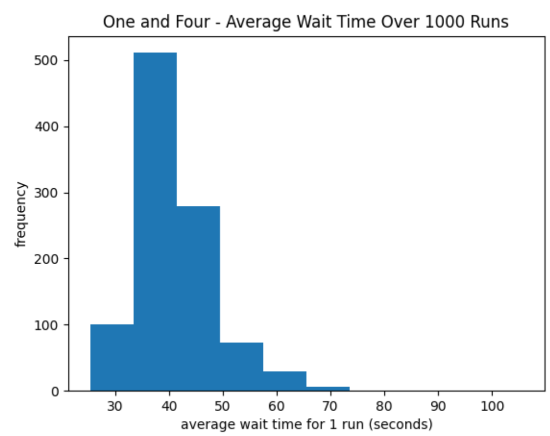


Mean of Replicate Mean Wait Times


Elevator Default - Average Wait Time Over 1000 Runs
frequency
average wait time for 1 run (seconds)


Idle at One - Average Wait Time Over 1000 Runs
frequency
average wait time for 1 run (seconds)


Sequential - Average Wait Time Over 1000 Runs
frequency
average wait time for 1 run (seconds)


One and Four - Average Wait Time Over 1000 Runs
frequency
average wait time for 1 run (seconds)

Conclusions

Analysis

I was quite surprised to find that none of the alternate scheduling strategies were able to perform better than the default. After considering why this might be and looking at the other configurations results, I've come to the conclusion that there is likely some impact as a result of a bias I introduce with my starting and ending floor selection, but the more likely culprit is something else. It wasn't until I stepped away from the project for a day before I realized it, there is a simple elegance to the strategy of parking the elevator where it was last used that is hard to beat. Meaning if people most often go to the first or third floor, the elevator is going to be left there more often than a floor that people go to less and all the other configurations take extra time and steps to move the elevator somewhere that people aren't. Additionally, I think that is why the one-four scheduling strategy was the closest to matching the average wait times.

Conclusion

In closing, I was not able to simulate an alternate scheduling configuration for the elevators of the Russ Engineering Center that reduced the average wait time. Although this result was surprising at first, it now seems to be the intuitive answer. There are other factors that I think could be interesting to consider in a later study, namely the collection of real data for the floor selection distributions, and even an increased resolution of the single rider experience instead of my "batch" rider approach.