Bus Simulation: How to improve run time on a busy bus service

Cy Chung
University of Victoria

CSC446

Instructor: Kui Wu

April 11th, 2023

Introduction

Public transportation can be extremely frustrating due to many outside variables that may cause delays and passengers unable to reach their destinations on time. The goal of this report is to simulate a rapid bus service in a busy metropolitan area. By the end of the project, the purpose is to find a solution on how to improve bus reliability by employing various time-saving techniques and measures used in busy transit systems. This project will utilize Python and Matplotlib to simulate and collect all statistics to prove this claim.

Simulation Model

The basic bus simulation model consists of three main characteristics, the bus, bus stops and the passenger. It also contains two channels where one of them is the

The bus stop contains a priority queue that holds all the passengers that arrive at the bus stop. It uses a FIFO priority. When a bus arrives at the bus stop, the passengers whose destination are at that stop exit the system before letting the people in the queue onto the bus. For the purposes of this report, Passengers entering and exiting a bus at an average rate of 0.5 seconds.

The bus contains a priority queue of passengers that entered from a bus stop. The queue is FIFO order based on when they entered the bus. The bus arrives at a set frequency or mean interarrival rate with a standard deviation to account for the bus being early or late.

Passengers arrive exponentially (in a poisson distribution) to a bus stop. Each passenger has a destination they want to go to unless they reach the end of the line. For the purposes of this study, passengers arrive approximately once a minute to a bus stop.

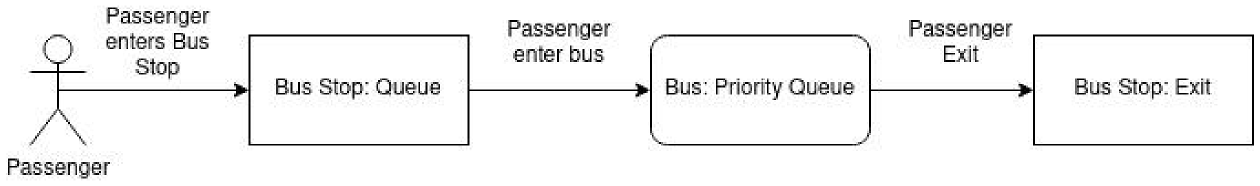


Figure 1: Bus Simulation Diagram

Data and Simulation Parameters

The simulation will be conducted with route 16 in Vancouver, BC. The 16 is an east to west trolley-route starting from Marpole passing through Downtown Vancouver before going down Hastings Street and terminating in East Vancouver at 29th Avenue station. In 2019, the 16 was the sixth busiest ranked bus route with a weekly average boarding of 24190 passengers and an average of 76 boardings per hour. The bus serves over 150 stops with approximately 78 stops served in each direction. The mean inter-arrival time during rush hour for each bus during peak hours is 7 minutes. The customer set amount of stops is set at a mean of 5 stops with a standard deviation of 1.5 stop. The interarrival time for each bus was tested with various combinations to represent days where the route may experience significant or no delays on the route. This can be seen in Table 1 where x is the arrival rate of each bus and y is the delay rate of each bus.

The main statistical goal to prove the original claim is by measuring the average trip time for a bus to make, the average amount of passengers on the bus, and the amount of passengers that are left behind at a stop due to overcrowding. The leftover passengers are calculated from the remaining passengers that are at a bus stop when the bus leaves that stop. The average number of passengers is calculated by getting the total number of passengers on the bus at each stop divided by the total number of stops. The average trip time will be the total trip time of all buses in an hour period by the number of buses run.

Table 1: Parameter values used for the Simulation
(Each parameter set was ran 5 times on seeds 10, 20, 30, 40 and 50)

Frequency	Standard Deviation = y
7	0.05

7	0.1
7	1
10	0.05
10	0.1
10	1
5	0.1

Methodology

The simulation was done in Python 3.9 alongside the matplotlib package to produce the histograms and tables found in the Results section. Initial code was built upon the many Python examples used in class as well as crediting Loïc Séguin-Charbonneau (loicseguin) for his version of simulating a bus system to help with finalizing the model design.

The simulation consists of five classes. There are three classes modeled after the characteristics mentioned in the simulation model, a statistics class that runs all the statistics and creates all the plots from the simulation result, an event class and a class that contains all the parameters for the simulation. The classes that are used are listed as follows (Code for these classes are available in the provided python file):

- Bus
- BusStop
- Passenger
- Event
- Simulation

The Bus class requires the following input parameters from the simulation class:

- Positional index
- A list of bus stops the bus has to visit
- The current time in the simulation

The Bus class stores information on the bus, the current amount of passengers on the bus and its corresponding stops and position. It also keeps track of the timing on when to let people in and off the bus.

The bus stop uses the following parameters:

- passengers: Passengers who are waiting at the bus stop
- position: Bus stop's position on the route
- average_passengers: Mean of passengers who usually wait at the stop
- next_arrival_time: Arrival time between passengers arriving at the bus stop

The bus stop is located somewhere along the bus' route. Passengers arrive in a Poisson distributed amount of time and are ordered in a FIFO queue. When the bus reaches the bus stop, passengers enter the bus in a FIFO priority with the remaining passengers left behind for the next bus.

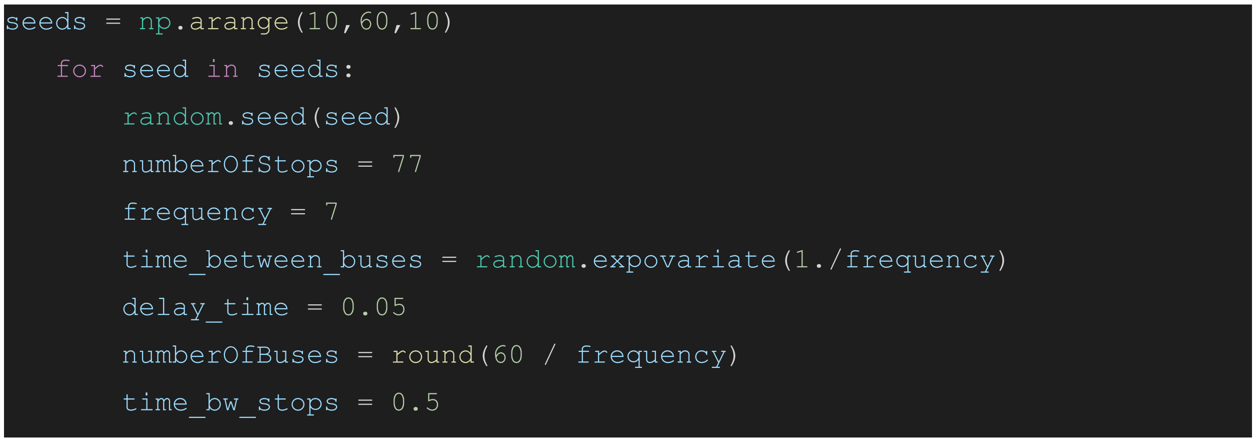
The Passenger class has the following parameters:

- depart_stop: Stop the passenger is departing from
- arrival_stop: Stop the passenger is arriving to
- start_time: When the passenger arrives to the stop
- end_time: When the passenger leaves the bus

The Passenger class stores information the passenger on where they want to get on and off the bus as well as their time entering and exiting the bus system.

The events class describes events that are placed into a timeline for the simulation to work through. An event occurs at a specific time and involves a specific object. Comparing two events amounts to figuring out which event occurs first.

The simulation class is in charge of running the entire bus system simulation. It contains and sets all the parameters described to run the system. The simulation runs five times on five different seeds; additional parameters can be changed before the simulation is run. This can be seen in the next section of methodology in Figure 2. For statistical collection, after the events were finished, the time to finish each run was printed to the console for each bus and averaging them.



```
seeds = np.arange(10,60,10)
   for seed in seeds:
       random.seed(seed)
       numberOfStops = 77
       frequency = 7
       time_between_buses = random.expovariate(1./frequency)
       delay_time = 0.05
       numberOfBuses = round(60 / frequency)
       time_bw_stops = 0.5
```

Figure 2: Parameters used in the simulation

Results and Analysis

The statistical results for the parameters described earlier are included in separate tables. As far as the correlation and final point estimation, the standard deviation for delay was not sufficient in making an impact on the average run time and the confidence for each run is very low as there is a high variance. (I believe this is due to the model being incorrectly implemented in a short time frame)

Table 2: Frequency = 7 Std. Dev= 0.05 | Average = 67.022

Seed 10	54.55
Seed 20	144.60
Seed 30	50.22

Seed 40	40.65
Seed 50	45.09

Table 3: Frequency = 7 Std. Dev = 0.1 | Average = 67.022

Seed 10	54.55
Seed 20	144.60
Seed 30	50.22
Seed 40	40.65
Seed 50	45.09

Table 4: Frequency = 7 Std. Dev = 1 | Average = 67.022

Seed 10	54.55
Seed 20	144.60
Seed 30	50.22
Seed 40	40.65
Seed 50	45.09

Table 5: Frequency = 10 Std. Dev = 0.05 | Average = 62.266

Seed 10	50.74
Seed 20	133.97
Seed 30	46.74
Seed 40	37.89
Seed 50	41.99

Table 6: Frequency = 10 Std. Dev = 0.1 | Average = 62.266

Seed 10	50.74
Seed 20	133.97
Seed 30	46.74
Seed 40	37.89
Seed 50	41.99

Table 7: Frequency = 10 Std. Dev = 1 | Average = 62.266

Seed 10	50.74
Seed 20	133.97
Seed 30	46.74

Seed 40	37.89
Seed 50	41.99

Bonus Run: Bus Capacity of 100 people | Frequency = 7, Std Dev = 1 | Average = 71.16

Seed 10	58.69
Seed 20	148.73
Seed 30	54.36
Seed 40	44.79
Seed 50	49.23

Conclusion

The simulation did not produce results as expected as the run time decreased as the frequency of buses decreased. A noticeable correlation happened in Seed 20 for all the results as the run time was higher than expected. During the simulation, I tested each line for this seed and noticed that the frequency of the bus was higher than the other seeds. However, it is considered a failed experiment because the delays were not prevalent enough to make a difference as well as the This is attributed to multiple errors in the simulation code that were unable to be fixed in time. A similar occurrence happened when running for a higher capacity of passengers.

The original method for running the simulation involved using Omnet++, however due to the small time-frame to learn the application, I was not able to create a proper simulation. Some improvements that can be made are to measure customer satisfaction. Customer satisfaction is important as it gauges how well received the changes are. Although some methods may improve the overall run time of the bus and improve reliability, if a customer is unhappy with the changes there will be no revenue generated from that bus since there will be less people riding them. I would also simulate a headway improvement instead of frequency and run time improvements. Buses can sometimes be delayed to where multiple buses stack causing big gaps in service.

In the scope of the project, a more detailed criteria would be useful as to see what my deliverables are. A simple run through of a simulation software in class or assignment would also help with this project as a real simulator would make the project less on the heavy side.

References

Banks, et al. "Discrete Event System Simulation", Fifth edition. Prentice Hall 2010

Loicseguin. (n.d.). *Loicseguin/roadsim: Simulate a single public transport circuit.* GitHub. Retrieved April 11, 2023, from https://github.com/loicseguin/roadsim

This class was interesting but not for me and I take full responsibility for my grade :)