London High-Speed Line Simulation and Optimization

Kenechukwu Otito Ajufo
School of Computing
National College of Ireland
Dublin, Ireland
x19190174

***Abstract*—This project involves Discrete Event Simulation and Optimization. The project has two objectives. The main objective of this project is to create a simulation for the London Old Oak Commons to Birmingham Interchange section of the London Underground high-speed line accommodating the variability of travelling times. After simulating the high-speed line, the next objective is to optimize the number of trains operating per hour. Here the goal is to maximise the number of trains operating per hour under the constraint that the average delay time is not be higher than half the scheduled time between consecutive trains. This is achieved by performing grid search optimization.**

***Index Terms*—discrete event simulation and optimization, grid search optimization**

I. Introduction

The project is aimed at modelling, simulating, and optimizing a high-speed line in the London Underground. Simpy [1] is used to create the train simulation. The processes used to achieve the goals of the project are discussed in the following sections.

A. Libraries

Libraries are groups of classes and interfaces which perform specific tasks. In this project the following Python libraries are used. Numpy and Pandas are used for data preparation and manipulation, matplotlib and seaborn are used to create visualizations and finally as previously stated Simpy is used to create the train simulation.

B. Dataset

The dataset is created by using the details in the context of the project descriptor. The dataset comes in a CSV file format which is then imported into the Google Colaboratory using Pandas.

II. Simulation - Part One

A. Simulation Entities

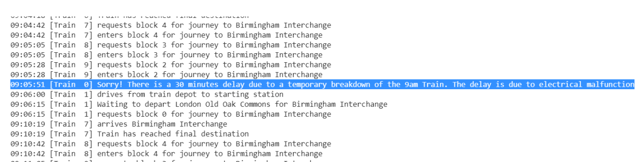
To create the simulation, the following entities were made: Train, Station and Signalling Block.

The first entity is the Train class, the train class has a constructor method where the instance's variables are declared. This includes the maximum speed and acceleration, brake speed and optimal deceleration. The constructor performs the task of assigning values to any instance variables that the object will need when the class is called. Furthermore, the Stations and Signalling blocks are defined in their respective code blocks. In the Signalling Blocks class, Simpy Resource is used to ensure the block has a capacity of one train at a time

The Train's process is defined in the process method. Here it is assumed the train takes 15 seconds to get to the first station. Although the maximum speed and acceleration, brake speed and optimal deceleration are declared in the constructor. I was unable to incorporate it into the train's process, so it is assumed that the trains travel at the same speed during particular stretches of the journey.

B. Simulation

The simulation model is verified by inducing a temporary breakdown due to electrical malfunction of the 9am train from London to Birmingham. This breakdown takes 30 minutes to fix. The effect of the breakdown is shown in the figure 1 below



```
09:04:42 [Train  7] requests block 4 for journey to Birmingham Interchange
09:04:42 [Train  7] enters block 4 for journey to Birmingham Interchange
09:05:05 [Train  8] requests block 3 for journey to Birmingham Interchange
09:05:05 [Train  8] enters block 3 for journey to Birmingham Interchange
09:05:28 [Train  9] requests block 2 for journey to Birmingham Interchange
09:05:28 [Train  9] enters block 2 for journey to Birmingham Interchange
09:05:51 [Train  0] Sorry! There is a 30 minutes delay due to a temporary breakdown of the 9am Train. The delay is due to electrical malfunction
09:06:00 [Train  1] drives from train depot to starting station
09:06:15 [Train  1] Waiting to depart London Old Oak Commons for Birmingham Interchange
09:06:15 [Train  1] requests block 0 for journey to Birmingham Interchange
09:10:19 [Train  7] arrives Birmingham Interchange
09:10:19 [Train  7] Train has reached final destination
09:10:42 [Train  8] requests block 4 for journey to Birmingham Interchange
09:10:42 [Train  8] enters block 4 for journey to Birmingham Interchange
```

Fig. 1. 9 am Train Delay

In order to reflect operational conditions and introduce variability of travelling times a log-normal distribution is used, the log-normal distribution is done using Numpy [2] this is based on Yuxiang Yan [3] which reports a log-normal distribution for the delay times in minutes. The same parameters in the paper are used here. The distribution plot of the delay time is shown in figure 2

However, I noticed that after implementing the log-normal distribution the code section that executes the 9 am delay due to the breakdown is never reached. As a result the log normal distribution for the operational delay is commented out so that the 9 am delay is implemented.

The number of trains and signalling blocks are varied to get an idea what the optimal values for both trains and signalling blocks are required. The visualizations are shown in Figures 3 and 4.

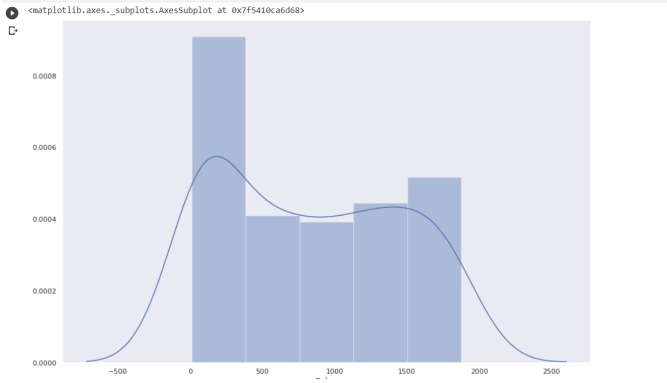


Fig. 2. The distribution of actual delay times

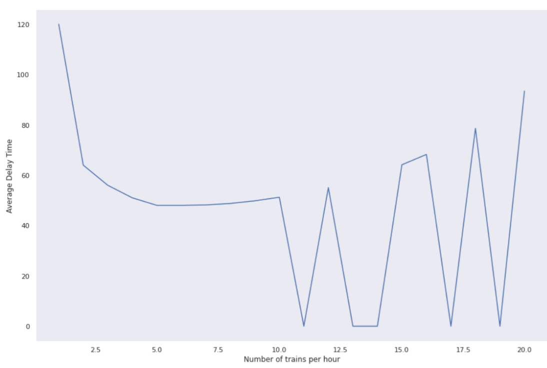


Fig. 3. Line Plot Showing Average Delay Time and Number of Trains

The results of varying the number of trains with the signalling blocks at a constant 14 shown in the Line Plot above, it is seen that at 11, 13, 17 trains per hour the average delay time is minimal.

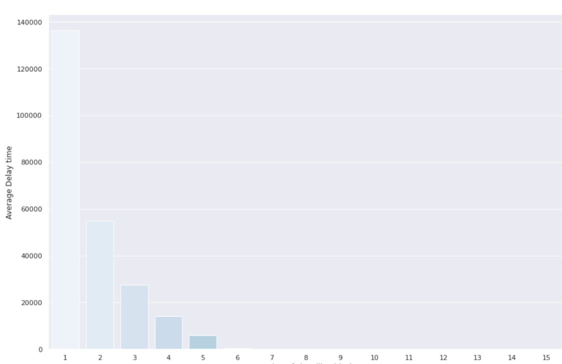


Fig. 4. Bar Plot Showing Average Delay Time and Number of Signalling Blocks

The results of varying the signalling blocks (k) shown above, It can be seen that at 13 trains per hour, the less signalling blocks, the more the average delay time. Five signalling blocks has the least average delay.

III. Optimization - Part Two

Grid Search Optimization is used to maximise the number of trains operating per hour under the constraint that the average delay time should not be higher than half the scheduled time between consecutive trains. To get the optimal number of trains the technique is run for a search space of 1-30 Trains per hour and 1-20 Signalling Blocks. The technique is exhaustive and computationally expensive, it is only used because the search space is relatively small (600 iterations). Because the technique iterates, the code block returns the optimal number of trains and signalling blocks that fit the constraints, these figures are saved and converted to a CSV file. A short list of the optimal number of trains and signalling blocks for the constraint are shown in Table I below

TABLE I
Optimal Number of Trains and Signalling Blocks that satisfy the constraint

Number of Trains	Signalling Blocks
11	6
11	7
11	8
28	16
28	17
28	18

IV. Conclusion

This project attempts to model, simulate and optimize the London Old Oak Commons to Birmingham Interchange section of the London Underground high-speed line. This is achieved using Google Colaboratory, Simpy and other Python Libraries discussed previously. Google Colaboratory is used because it enables me get faster computational power compared to my 8 GB RAM in my personal computer.

References

[1] "Simpy in 10 minutes — simpy 3.0 documentation," https://simpy.readthedocs.io/en/3.0/simpy_intro/index.html, (Accessed on 08/06/2020).

[2] "numpy.random.lognormal — numpy v1.19 manual," https://numpy.org/doc/stable/reference/random/generated/numpy.random.lognormal.html, (Accessed on 08/06/2020).

[3] Y. Yang, P. Huang, Q. Peng, L. Jie, and C. Wen, "Statistical delay distribution analysis on high-speed railway trains," *Journal of Modern Transportation*, vol. 27, no. 3, pp. 188–197, 2019.