Feedback

Hi Varughese,

Glad to see you utilised the provided schedules in your final report, well done! As I understand from the report, your recommended approach for dealing with emergency jobs does not account for the jobs deadline constraint. In this case, the observations look interesting for the future of the project. Thanks for sharing the report.

Kind regards,
Arezoo.

Report:

AIM:

Making modifications to preoptimized schedules of facilities maintenance engineers for two instances:

1) Inserting a new job before the beginning of a workday

2) Inserting a new job during the workday

Implementation:

A scheduling algorithm was designed based on local search heuristics and the algorithm was tested on the real-life test data that was provided. Ten percent of the jobs assigned in each of the schedules were randomly removed and reassigned using the proposed scheduling algorithm and the results before and after reinsertion was analysed. The primary feature that was being compared were the total driving times for each engineer in the schedules.

The first schedule to be tested was day one of the 5-Day schedule where 60 jobs out of 600 jobs were randomly removed and reinserted into the schedules of 110 engineers. But due to the request limits set by the open-source geocoder and routing engine that was used, the analysis was limited to 30 engineers doing 150 jobs.

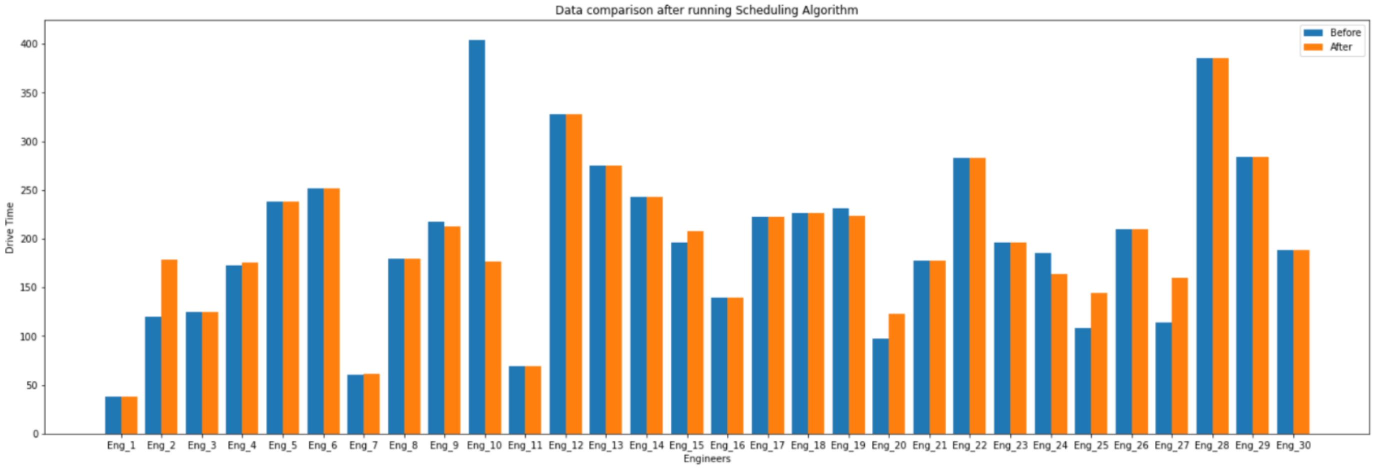

Figure 1: Drive Times of Engineers for Day 1 in 5-day Schedules (S1)

Similarly, the same test was done on the other three sample schedules.

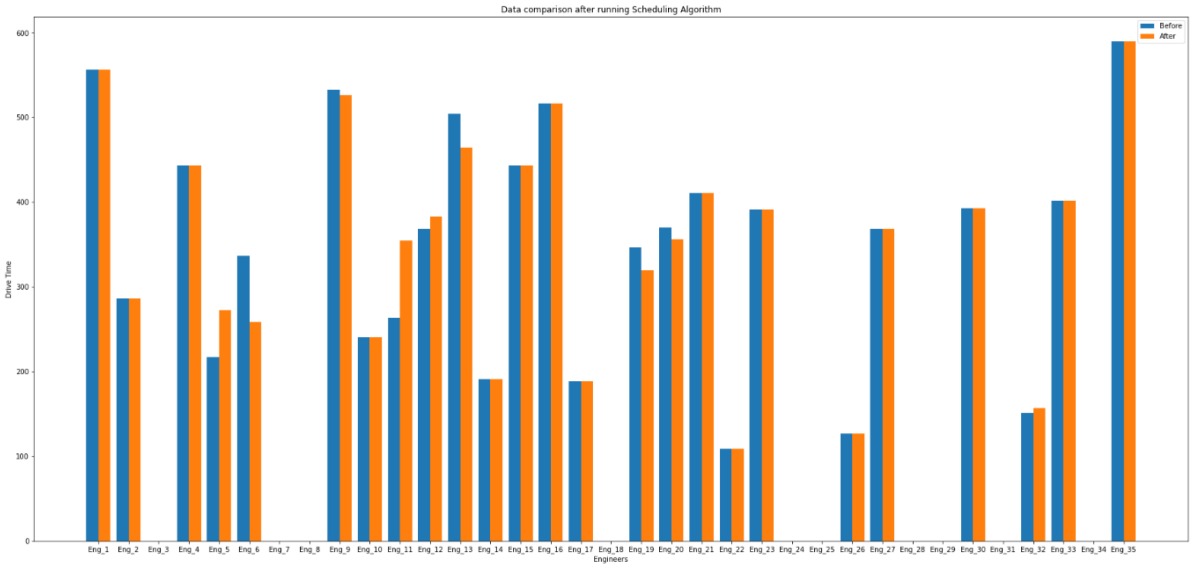

Figure 2: Drive Times of 35 Engineers assigned 97 Jobs (S2)

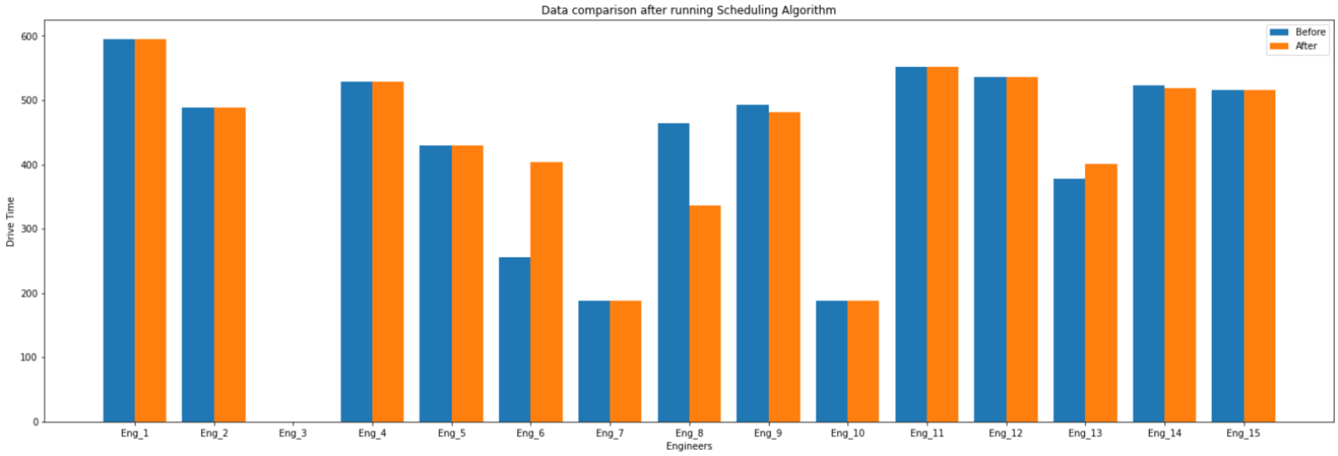

Figure 3: Drive Times of 15 Engineers assigned 59 Jobs (S3)

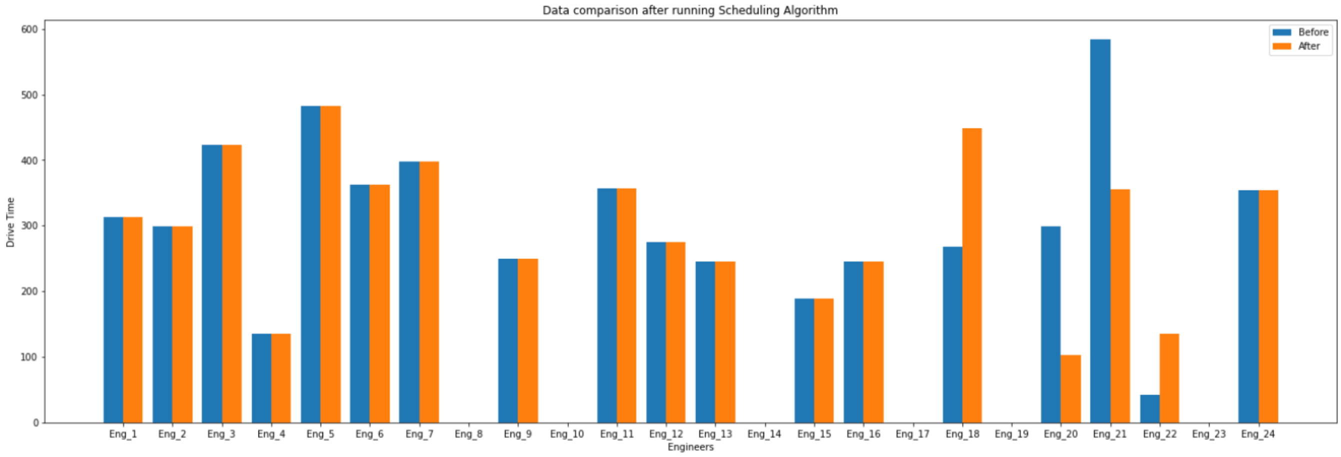

Figure 4: Drive Times of 24 Engineers assigned 38 Jobs (S4)

The general trend that can be seen from all four test cases where ten percent of the assigned jobs were randomly removed and reassigned, is that the driving times of many engineers are very similar. The schedules that were used in the test were preoptimized and the role of the proposed scheduling algorithm is to insert a new job into them. If the driving times of engineers were similar before and after the test were conducted, then that means that the scheduler was able to reassign the job into a position which was considered as optimal. The scheduler has also been able to significantly reduce the driving times of some engineers by assigning the jobs to more appropriate engineers. This can be seen in the drive times for engineer 10 in schedule 1 (Figure 1 (S1)) and engineer 21 in schedule 4 (Figure 4 (S4)).

The scheduler was able to evenly distribute the jobs such that the total drive times of the schedules were more or less equal to the original schedule. But the overall drive time for some engineers were increased by assigning them new jobs so that the there would be a significant reduction in the drive times of other engineers. A bar graph of the total drive times of all the engineers (S1 to S4) before and after the test was conducted is as follows:

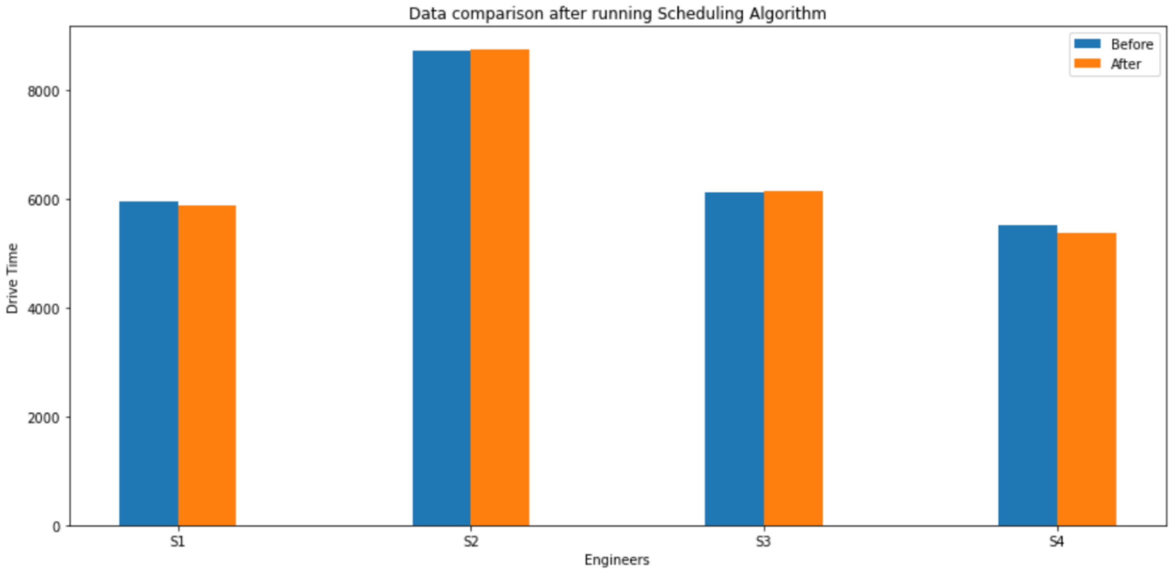

Figure 14: Total Drive Times the Schedules (S1 to S4)

Recommendation:

Based on the results from the evaluation, the best approach for inserting a new job into a preoptimized schedule is to insert the job into a qualified engineer’s schedule and run a Travelling Salesman Problem solver on the schedule to get the best combination of jobs. Then a greedy descent approach can be taken which searches for a better solution by performing the same procedure on another schedule. If a schedule is found with a lower total time, then it becomes the new solution. The search is performed for all the other assignments until a global optimum is reached.

This approach would result in the least number of changes made in the original schedule and a solution will always be obtained unless hard constraints such as time limits or skill requirements are placed. But such constraints can be useful for the scheduling algorithm as the search space would be reduced and a global optimum can be found faster.

The primary weakness of this approach can be seen in the analysis that was conducted on the schedule S1 with 110 engineer and 600 jobs. Since no hard constraints such as skill requirements or time limit was specified, the algorithm must consider a large search space and a global optimum cannot be reached without systematically searching every possible assignment, making it very time consuming.

The scheduler is meant to be used during two instances:

1) Inserting a new job before the beginning of the workday:

This can be done by following the implementation procedure mentioned above by providing the algorithm with the original schedule and the new job which needs to be inserted.

2) Inserting a new job during the workday:

This scenario can be tackled in a similar way as instance 1. The locations of the engineers are first tracked. This will usually be the location of the jobsite they are currently at. Fleet tracking systems like Verizon Connect, Samsara or Jobber can be used to get accurate estimates of engineer locations. Once such data has been recorded, the completed jobs are removed from the schedule and the ongoing jobs become the starting locations for the engineers. The new job is then inserted into an engineer’s schedule depending on the urgency of the new job in a similar fashion as instance 1.