Train Schedule Optimization with focus on Robustness

Focal Points: Train Timetables, Optimization, Simulation

Student: Kevin Reust

Supervisor: Dr. Donnacha Daly, Lucerne University of Applied Sciences and Arts

Expert: Sandro Cilurzo, Sedimentum AG

Client: Lucerne University of Applied Sciences and Arts - Department of Information Technology, Rotkreuz

Keywords: Genetic Algorithms, Simulation-based Optimization, Train Timetables, Rail Transport Planning, Train Networks

1. Importance of robust train schedules in modern railway networks

Scheduling of railway networks is an old problem, which is extremely well studied and understood. Switzerland is clearly a great example, having a railway network which functions very well in delivering punctual, densely connected and frequent train services across the nation. Nevertheless, there are two pressing issues which confound scheduling planners and need good solutions for future scheduling:

a) The number of passengers is increasing at a fast rate, which requires more trains running more frequently on the same network.
b) The impact of delays in a more highly utilized network is more significant, with cascading effects causing rolling delays in over congested parts of the network.

The goal of the project is to explore different methods for optimizing train schedules, such that the resulting schedules can absorb delays and stress in the network. The optimized schedules are then used to examine the trade-off between efficiency and robustness.

2. Comparison between robust and efficient schedules

In the context of the bachelor thesis the following results were developed:

- **Simplified Train Network Simulation**: A simplified train network simulation with stations, trains and tracks was implemented using Python 3.7. In the simulation, behaviours such as passenger movement, traffic and resource management, as well as train operation were modelled.
- **Performance Indicators:** To rate the performance of a train network with a given schedule, a set of performance indicators were designed for the simulation. The performance indicators reflect the punctuality and effectiveness of passenger transport on the network, as well as the ability of the network to absorb shocks.
- **Genetic Optimizer:** An optimizer for train schedules was implemented from scratch in Python using genetic algorithms. It utilizes the performance indicators of a simulated network as its objective function to create robust or efficient schedules.

The described products were then used to perform a series of experiments with star networks of different sizes, whereby the schedules were optimized once for robustness and again for efficiency. The aim was to investigate the performance difference of robust and efficient train schedules under normal network conditions and under continuously increasing network load.

It was expected that efficient schedules would work better under normal conditions but would have a rapid decline in performance under stress. On the other hand, robust schedules would initially perform slightly worse than efficient schedule but would maintain their performance level for much longer and therefore outperform the efficient schedules on congested networks.

The results from the experiments showed, that for increasing amount of trains or delays in the network, the described behaviour could indeed be observed in the examined star networks. In contrast, the increase in passenger numbers led to a similar decline in performance for both robust and efficient train schedules. Under normal conditions, the performance of efficient schedules in each experiment was higher than that of robust schedules.

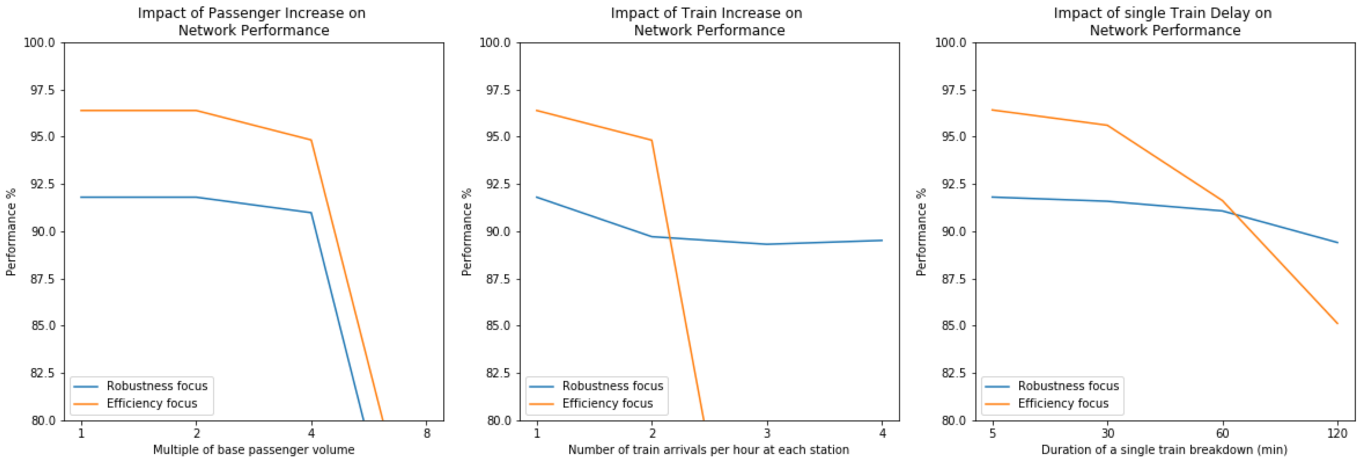

Figure 1 - Comparing the performance of robust and efficient schedules under different types and levels of stress. The star network under study consisted of 10 stations, 10 track sections and 2 train lines. The performance is an indicator for the punctuality and the effectiveness of the network to transport passengers.

3. Concept for creating robust train schedules

The problem of creating robust timetables was divided into the following four subproblems:

- **Model:** The model defines the basic components (e.g. Stations, Tracks) as well as the relationship between the components (e.g. Track tAB connects station A with station B), such that an arbitrary train network can be created.
- **Simulation:** The simulation handles all interactions between the components defined in the model (e.g. train movement, passenger movement)
- **Results:** The results are used to analyse and rate the performance of a train network with a given schedule. They contain human readable information in form of plots/visualizations, as well as performance data used by the optimizer.
- **Optimizer:** The optimizer utilizes multiple simulations as its objective function to generate robust schedules.

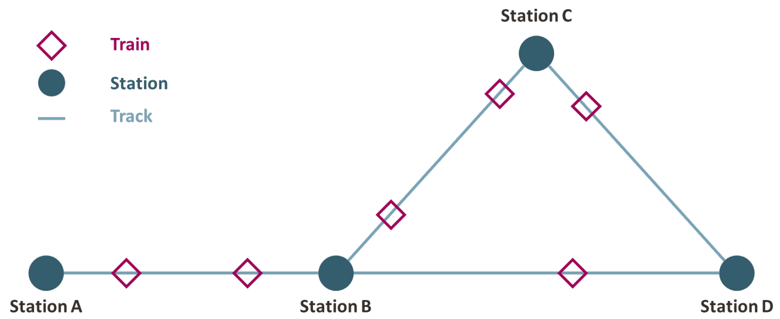

Figure 2 - Simplified model of a train network with stations, trains and tracks.

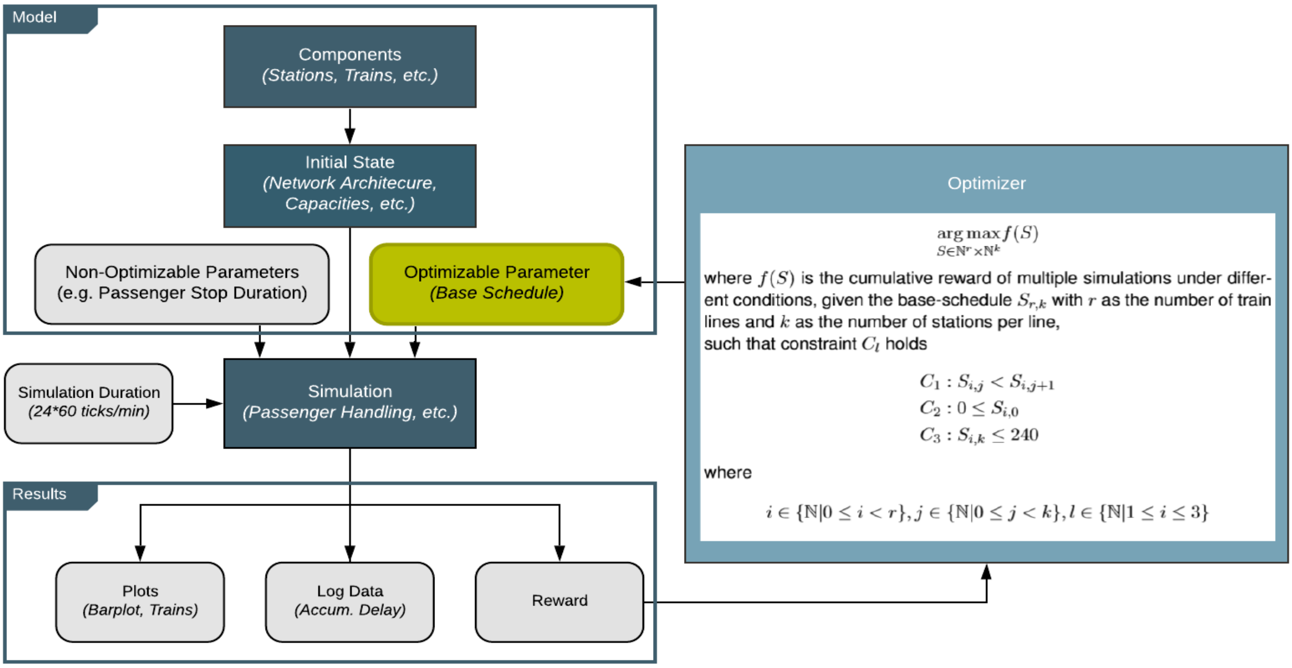

Figure 3 – Concept for create optimized train schedules.

4. Challenges in the modelling of real-world railway networks

One of the main difficulty in the project was to create a simplified simulation of a complex real-world system, so that the computing time for the simulation is as short as possible, while at the same time modelling the most important behaviour of train networks. This required to model complex tasks such as traffic management, passenger movement, network design, etc. The complexity additionally led to many variables (e.g. passenger demand per hour per station, minimum dwell and turnaround times, train speeds, etc.) which have an influence on the performance of the network and therefore had to be fixed so that the experiments were comparable. Therefore, a lot of time went into the process of designing a network component, analysing the performance of the resulting simulation and then change the component to counteract unintended behaviour.

5. Suggestions for improving simulator and experiments

The project results have shown that simulation-based optimization with genetic algorithms can work well in creating robust schedules in small star networks. As the performance indicators used in this work were based around the operational performance of the network (delay, transported passengers, etc.), it would be interesting to introduce additional performance indicators, such as passenger satisfaction (e.g. travel duration, number of train changes between locations), in a future project.

The simulation software could also be used as the basis to implement different optimization methods, for instance reinforcement learning.