optimising railways

SMA and partners Ltd

Gubelstrasse 28
8050 Zurich
Switzerland
Phone +41 44 317 50 60
info@sma-partner.com
www.sma-partner.com

Robustness Analysis through the Algorithm Platform

Abstract

We would like to present our robustness analysis tool, which we have applied in the context of a case study that we have carried out for a customer. Parts of it (e.g. the dispatching strategy) are connected to Viriato via the Algorithm Platform. We will show selected features of our tool, highlight the advantages of our approach using an external algorithm over a monolithic and closed implementation, and explain in which aspects a macroscopic robustness analysis can outperform a microscopic one. We will describe how the tool was helpful to us in the case study.

The Case Study

SMA has carried out a study to assess the effect of different actions on a given timetable over a wide area of the Western Swiss railway network on the basis of a macroscopic train simulation using a dispatching algorithm connected to Viriato through the Algorithm Platform. The goal was to evaluate the effectiveness of slight timetable variations on the required fleet size, robustness and travel times. We planned the variations on the basis of the reference timetable and used them as input scenarios to the analysis: These scenarios were: offering fewer connections, reducing the service offer, regularising the timetable and redesigning the timetable through the modification of reserve times.

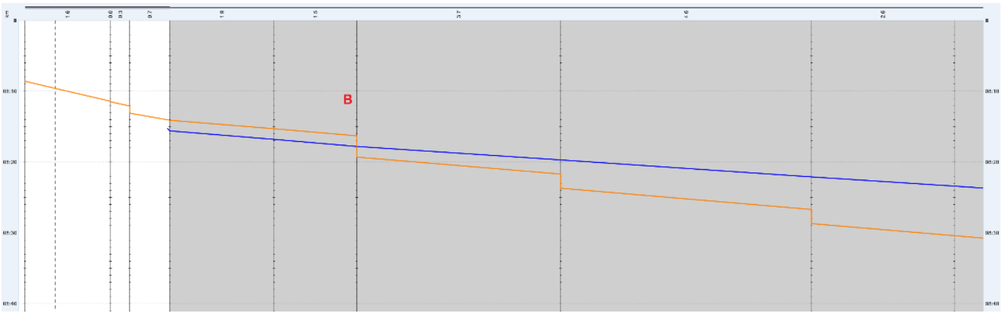
We defined the infrastructure model only once, and used roster and connection links as an input for each input scenario which we then compared to the reference timetable in order to analyse the network-wide effects. To assess the robustness of each scenario, initial delays were inserted into the test scenarios and a simulation run. These initial disruptions in the test scenarios were developed from complex situations observed in operational practice (e.g. Friday evening, Sunday, disruptions on a congested section, etc.) influencing mainly the duration of train stops and general speed restrictions. The output of a simulation run is a set of KPIs and a visualisation of the results in a graphic timetable. The analysis of the KPIs and the as-run timetable allowed the assessment of the effects of each set of actions on the stability of operations. As a result of our case study, we gained the insight that regularising the timetable and operating faster trains with more reserve time had the largest impacts, and this led to much more robust timetables. We also found the unexpected result that defining fewer connections within a given timetable did not make a significant improvement to robustness if not combined with a structural change to the timetable itself.

SMA Offices

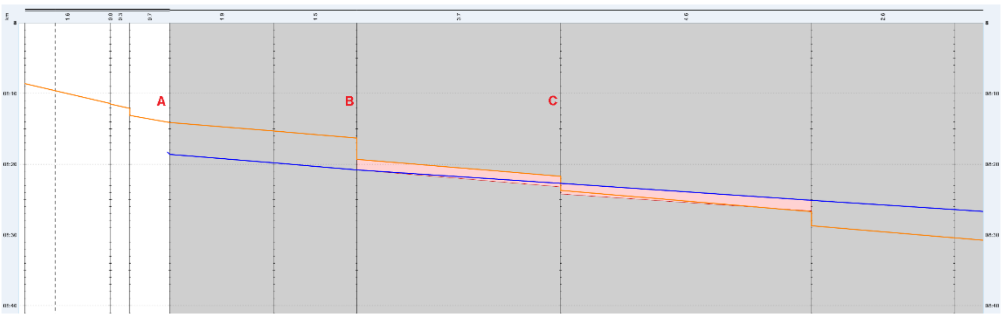
Zürich
Lausanne
Frankfurt
Paris

Exchangeability of the Conflict Resolution Strategy

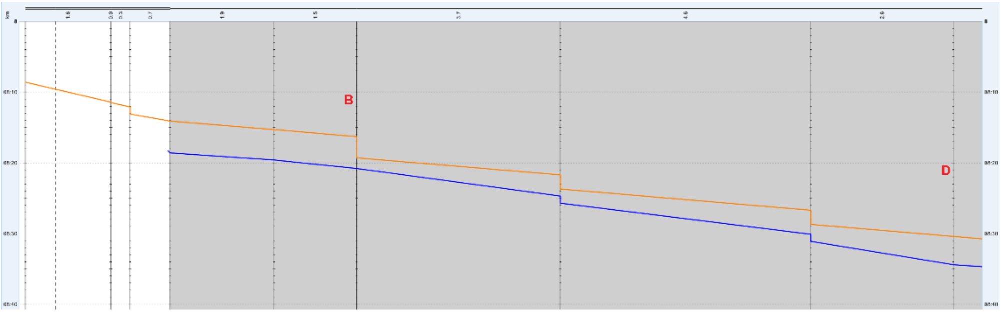
The dispatching strategy decides how resource conflicts, e.g. two trains competing for a section track, are resolved and therefore has a crucial impact on the results of a robustness analysis. For further information about the relationship between the simulation and a robustness analysis, we recommend our previous post "Robustness vs. Train Simulation". Our tool allows a customised dispatching strategy to be plugged-into the system, allowing more accurate, and country-specific modelling, of a complex system's behaviour than any fixed or pre-defined strategy from an off-the-shelf implementation, which can be demonstrated through the following example. Consider the situation below in which there is an overtaking activity planned in Node B. Note that the grey area is a single track section.



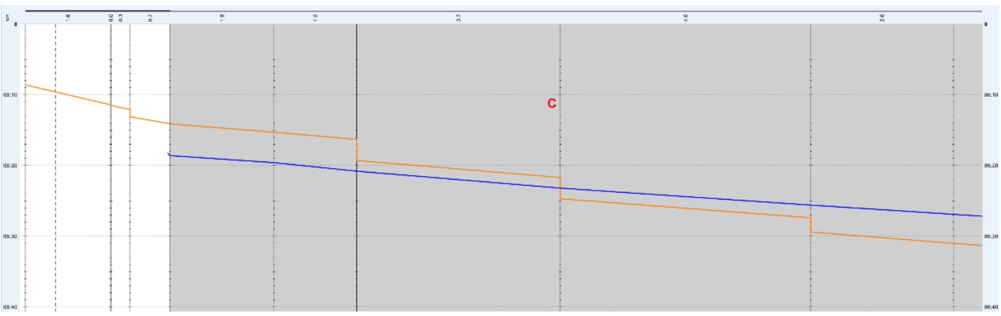
If the blue train arrives with an arrival delay of three minutes at the Junction A, there will be section track conflicts created on the section tracks after Nodes B and C. In addition there will be a node track conflict in C because trains there were planned on the same node track.



The choice of the dispatching strategy has a crucial impact on the delay of the trains in operation. If we used a simple First-Come-First-Served based strategy (FCFS) in our simulation, which does not allow node track changes, then in case of a resource conflict, as in the figure above, FCFS by definition lets the train that arrives first at the node track pass through Node C before any other. The screenshot below shows the as-run result of this situation, and it is apparent that the initial three minutes delay of the blue train at A leads to a delay of about eleven minutes in Node D.



Choosing an FCFS strategy is a natural initial choice because it is simple to implement and to conceptually understand what happens on the network when a disturbance occurs. However, it is only realistic when there is little deviation from the timetable or when there is little room for influencing the traffic situation through dispatching actions. Our robustness tool, which uses the Algorithm Platform, has an interface to which algorithm developers can attach their own dispatching strategies. For example, allowing a deviation from the planned node track may make a crossing or overtaking possible that was previously planned to occur at a different node, which may lead to a reduced total delay. In the example below, the dispatcher decided to deviate from the planned node track assignment for the blue train in node C. In order to make this manoeuvre possible, they decided to allow the orange train to lose one minute in Node C. Carrying out the overtaking in Node C now leads to an additional small delay of the blue train between B and C when it has to follow the slower orange train. Nonetheless, the blue train will have an overall delay in Node D of only 3.5 minutes, a total saving of more than seven minutes in comparison to the original FCFS strategy.



Alternatively, the dispatcher could have decided to give additional extra stopping time for the orange train in node B so that the overtaking still would have taken place there which results in no further delay for the blue train between B and C. We can see that the preferences of the dispatcher or country-specific operational rules determine which decision is likely to be chosen in a given situation.

In the future, we will continue to work on more features related to changing planned train paths. For example, we will include running-time penalties when a node track is changed during the simulation to account for lower speed limits over the switches or for other country-specific safety and operating rules. We also intend to investigate further how the choice in which node an overtaking or a crossings is implemented, in order to propose dispatching strategies for reducing delays.

Network-wide Analysis: Dealing with Uncertainty and Scalability of a Macroscopic Infrastructure Model

Viriato allows the construction of an infrastructure database with low input data requirements, minimising the time needed for defining the model. Having less data in the model compared to other modelling methods may seem like a drawback at first, but it can actually be a strength. We call this philosophy "step appropriate precision", and it enables the user with only a few clicks to model infrastructure that has not yet been built. The impact of building a new track on the capacity in a network can then be accurately estimated, even though it may not yet be known where the individual elements of the infrastructure (e.g. switches, signals, axle counters, etc.) will be located. Making assumptions about the location of those infrastructure elements means we would be pretending to work at higher a level of accuracy than is really available. This reduces the costs for modelling the infrastructure and carrying out the analysis. In addition, in our experience errors made by analysts during the construction of a microscopic model are harder to find than in macroscopic models. Our tool also allows conflicts outside the area of interest to be ignored, again saving time for input data preparation.

This coarse-grained macroscopic model also provides the benefit of a shorter simulation running time. As there is significantly less data in comparison to a microscopic infrastructure model, simulation of a larger network area and more iterations of the simulation can be carried out without requiring dedicated simulation hardware. Through another case study we have observed that the macroscopic simulation is approximately 50 to 60 times faster than a microscopic simulation, depending on the dispatching strategy when run for the same model area on identical hardware. For example, with this model we have been able to simulate several hundred iterations of a significant part of the Swiss railway network in an acceptable amount of time. Trading a speed-up of the computation time against the loss in precision of microscopic tools is generally acceptable for network-wide studies.

Therefore, a macroscopic simulation comes at only a fraction of the cost of a full microscopic simulation. This allows analysis which are rarely undertaken with microscopic simulation, of the the impact on larger network areas both technically and economically. Because of this cost-efficiency, it is also possible to study multiple variants of possible future upgrades to an existing infrastructure layout and other factors that influence the analysis, such as different dispatching strategies, connection scenarios, etc.. In our case study these advantages enabled us to perform an analysis of multiple scenarios.

Transparency of a Deterministic Simulation Approach and Extension to a Monte Carlo Simulation

Regarding the input data, a simulation run needs an infrastructure, a timetable and a set of primary delays. For a fixed set of delays and base timetable, a simulation run is carried out with the results reflecting the behaviour of the trains over Viriato's infrastructure according to the conflict model and the selected dispatching strategy. The result of each simulation can be imported into Viriato and stored in a timetable scenario, providing full transparency to the user of what occurred and the consequences that the input delays had on the timetable. The simulator and dispatching algorithm record the reasons why specific actions were performed and the causes of delays.

Simulation Time	Node	Delayed Train	Delayed Event Type	Conflicting Train	Delay Reason	Delay Duration	Updated Forecast
01.04.2010 07:23:54	A	IC LE-CS 01 06:35.0 01.04.2010	Departure		Primary delay occured	0:06:00	01.04.2010 07:31:00
01.04.2010 07:37:00	R	IC LE-CS 01 06:35.0 01.04.2010	Arrival	MR LE-CS 01 07:36.0 01.04.2010	Deceleration for / to avoid unplanned stop	0:01:00	01.04.2010 07:38:00
01.04.2010 07:38:00	R	MR LE-CS TEST 07:40.0 01.04.2010	Departure	MR CS-LE 02 07:27.7 01.04.2010	Waiting for train arrival (roster link)	0:00:48	01.04.2010 07:38:48
01.04.2010 07:38:00	R	IC LE-CS 01 06:35.0 01.04.2010	Departure	MR LE-CS 01 07:36.0 01.04.2010	Headway conflict	0:01:00	01.04.2010 07:39:00
01.04.2010 07:38:48	R	MR LE-CS TEST 07:40.0 01.04.2010	Arrival	MR CS-LE 02 07:27.7 01.04.2010	Waiting for train arrival (roster link)	0:02:24	01.04.2010 07:41:12
01.04.2010 07:39:00	A	IC LE-CS 01 06:35.0 01.04.2010	Departure		Acceleration after unplanned stop	0:02:00	01.04.2010 07:41:06
01.04.2010 07:40:00	B	MR LE-CS TEST 07:40.0 01.04.2010	Departure		Primary delay occured	0:07:00	01.04.2010 07:52:00
01.04.2010 07:41:12	R	MR LE-CS TEST2 07:40.0 01.04.2010	Departure	MR CS-LE 02 07:27.7 01.04.2010	Waiting for train arrival (roster link)	0:11:30	01.04.2010 07:52:42
01.04.2010 07:41:30	B	IC LE-CS 01 06:35.0 01.04.2010	Departure	MR LE-CS 01 07:36.0 01.04.2010	Headway conflict	0:00:06	01.04.2010 07:42:00
01.04.2010 07:41:30	B	IC LE-CS 01 06:35.0 01.04.2010	Departure		Primary delay occured	0:04:00	01.04.2010 07:47:00
01.04.2010 07:42:00	C	IC LE-CS 01 06:35.0 01.04.2010	Arrival	MR LE-CS 01 07:36.0 01.04.2010	Separation time conflict	0:01:24	01.04.2010 07:44:00
01.04.2010 07:42:00	C	IC LE-CS 01 06:35.0 01.04.2010	Departure		Primary delay occured	0:20:00	01.04.2010 08:04:00
01.04.2010 07:51:00	B	MR CS-LE 02 07:34.0 01.04.2010	Departure	IC LE-CS 01 06:35.0 01.04.2010	Waiting for passengers to get onto the train (connection link)	0:02:00	01.04.2010 07:53:00
01.04.2010 07:52:42	D	MR LE-CS TEST 07:40.0 01.04.2010	Departure	MR CS-LE 02 07:27.7 01.04.2010	Waiting for train arrival (roster link)	0:01:36	01.04.2010 07:54:18
01.04.2010 08:02:06	A	MR LE-CS TEST 07:40.0 01.04.2010	Arrival	IC LE-CS 01 06:35.0 01.04.2010	Headway conflict	0:03:18	01.04.2010 08:07:00
01.04.2010 08:20:18	C	MR LE-CS TEST 07:40.0 01.04.2010	Departure	IC LE-CS 01 06:35.0 01.04.2010	Separation time conflict	0:00:06	01.04.2010 08:21:06

Besides a fixed set of manually defined input delays, our tool also allows the flexible definition of departure distributions by node, train, train type or line. Through these distributions, we can model delay impacts from infrastructure, train or operational issues such as stops-on-demand, delayed departures in crowded stations or only for certain lines. At the beginning of a simulation run the occurrence of initial delays will be taken from the given statistical distributions. As in the single-run simulation case, the drawn delays form an input set for one deterministic run. We have extended this concept to perform a Monte Carlo analysis of the system: We carry out multiple runs, each time drawing the new primary delays according to the statistical distributions at the beginning of each run. In practice, the occurrence of delays is often modelled with an exponential distribution giving a constant failure rate. Because of the scalability of our model, we can run up to thousands of simulations per day depending on the modelled network size. The data for each run, including the specific input delays used are stored, allowing single simulation runs to be recreated and inspected manually. Various KPIs are collated for the Monte Carlo analysis, allowing us to make quantitative and qualitative statements on the robustness of a timetable. The most common KPIs used include median, average and maximum delays in a node, the number of connections held or broken and more. In this case study the available KPIs significantly aided us in our analysis.

Run Number	Number of Trains with Primary Delay	Number of Delayed Trains	Number of Primary Delays	Total Delay	Total Primary Delay	Median Primary Delay	Total Secondary Delay	Total Delay at Node Arrivals	Time to Recover	Total Remaining Delay	Median Total Delay of any Delayed Train	Maximum Total Delay of any Train
[illegible]	2	3	2	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1:02:12		[illegible]	[illegible]
2	0	2	0	5:54:24	0:00:00	0:00:00	5:54:24	2:38:12	0:32:12		5:20:36	5:20:36
3	1	2	1	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0:12:12		[illegible]	[illegible]
4	1	3	1	6:35:06	0:01:54	0:01:54	6:33:12	5:16:42	0:32:12		0:58:36	5:20:36
5	1	2	1	[illegible]	0:08:24	0:08:24	9:41:12	4:35:48	0:40:36		9:15:48	9:15:48

Median Arrival Delay of any Delayed Train at Destination Node	Maximum Arrival Delay of any Train at Destination Node	Median Delay of any Train in any Delayed Node	Maximum Delay of any Train in any Node	Number of broken connection links	Number of delayed trains due to connection links	Number of Connection Links held without extra waiting for Incoming Train
0:01:54	0:16:42	0:00:54	0:16:42	0	0	1
0:09:42	0:09:42	0:09:42	0:12:12	0	0	1
0:09:42	0:09:42	0:00:41	0:12:12	0	0	1
0:02:18	0:09:42	0:03:06	0:12:12	0	0	1
0:18:06	0:18:06	0:18:06	0:20:30	0	0	1