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

sma+

Robustness vs. Train Simulation

Abstract

We are working on a macroscopic train simulator using our Viriato Algorithm Platform that allows us to apply customised conflict resolution strategies for robustness studies in Viriato. The goal of SMA's ongoing development is to validate that robustness studies can be carried out using an interaction of a dispatcher with a macroscopic traffic simulation of a railway network. We are proceeding with this by using the Viriato Algorithm Platform as a source for infrastructure and timetable data. Following this step we are going to enhance the simulator with features - e.g. to sample randomly stops on demand - to increase its analysis capabilities. From the study of the realisations of a timetable under perturbation scenarios, recommendations on how to plan robust timetables can be derived for train planners. We will also develop the framework to carry out Monte-Carlo types of analyses to give the user even more tools for robustness studies of timetables. The dispatching strategies are exchangeable in our model. Therefore, customers can also implement their own dispatching strategies using the Viriato Algorithm Platform to model the dispatching behaviour in their own networks and to study their timetables in cases of deviation from planned. Moreover, the results of robustness studies carried out by different dispatching strategies can be compared, which is interesting in its own right as potential changes in dispatching recommendations can be investigated. In this post, we are going to first explain the existing Viriato robustness module and then show its relation to a macroscopic train simulation. In a later post we will present the current architectural draft of this prototype.

Timetable Robustness and the Viriato Robustness Module

Robustness is the key property of a timetable to resist deviations from plan caused by delays due to train operations. In a Viriato robustness analysis we investigate what effects these deviations have on the planned timetable and how long it takes to return to normal train operation after a disturbance has occurred on the network. To do this, we choose a timetable scenario, a delay scenario, define constraints between trains, known as a 'Link scenario' in addition to the general modelling parameters.

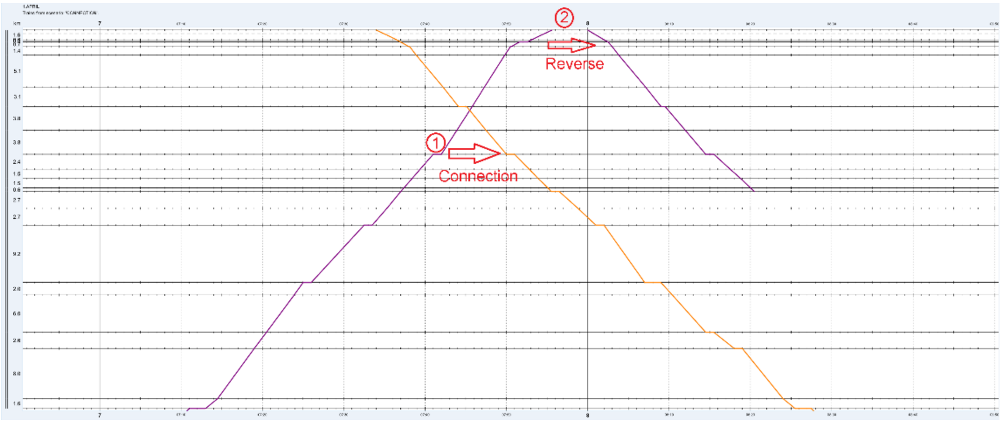
In Viriato there currently exist two types of constraints. Connections are a commercial requirement to offer passengers a high quality of service. When a delay to a train occurs, its connections may be broken in order to not delay other trains or alternatively not to further delay late trains. A connection can also be held in order to allow the passengers on the delayed train to reach their connecting

SMA Offices

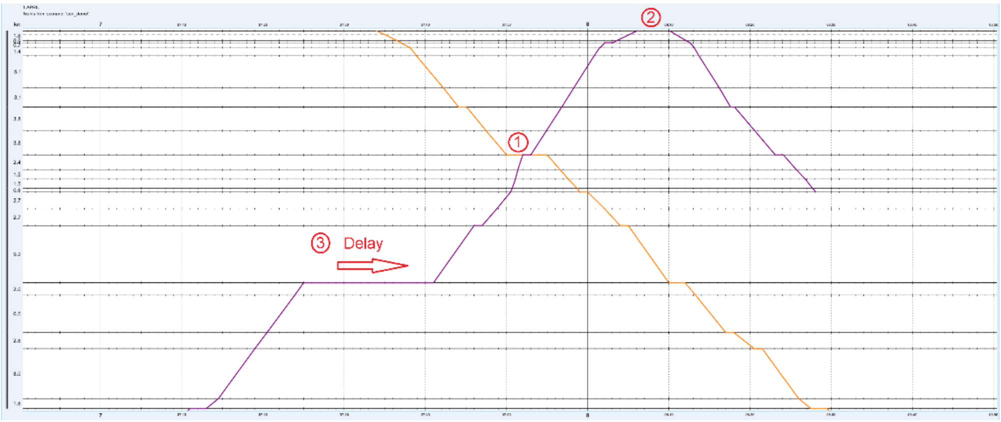
Zürich
Lausanne
Frankfurt
Paris

train. This second type of constraint cannot be broken, and is used to represent any dependency between two trains due to train reversing movements, joins or splits, etc.

Consider a simplified study in the figure below in which we set a delay of 15 minutes to the purple train on the left, for example to simulate a temporary vehicle failure. In Node 1, let there be a planned connection between the purple and the orange train. In Node 2 let there be a train reverse.



Running the Viriato robustness analysis gives the following timetable.



We see that the trains was delayed in Node 3 through the use of increased stop time there. If we assume that the properties of the connection defined in Viriato at Node 1 is that it should not be broken for at least 10 minutes, then in our case the connection is held, allowing the passengers to board the orange train which causes a four minute delay to it. The train reverse at Node 2, as specified by a dependency in Viriato, means that the purple train on the right must depart at least four minutes later than planned, as the minimum turnaround duration after a reverse is a property of the dependency specified in Viriato. For example, a minimum dwell time of four minutes set between the arrival of the incoming train and the departure of the outgoing train could be to give the driver sufficient time to walk to the other end of the train before they can depart again.

Macroscopic Train Simulation with the Viriato Algorithm Platform

Taking the perspective of an observer who is watching the traffic situation and sees the trains as they run, in the previous example the observer sees at 7:26 that the purple train cannot depart as planned, and that it has a delay and will stop longer. If the observer is a dispatcher, they know that the train will need at least 11 minutes to travel from Node 3 to Node 1. Therefore, if the passengers need 3 minutes to change in Node 1, the dispatcher knows at 7:37 that they must decide at the latest by 7:51 if they want to hold or to break the connection between the purple and orange trains.

Viriato has a macroscopic infrastructure model, meaning the infrastructure consists of nodes with their node tracks, sections with their section tracks, but no signals or switches and a conflict model aligned to this macroscopic level. Therefore, after a departure the exact position of a train on the section track is unknown and the observer sees the train only when it departs from a node or when it arrives at the next node. Thus the situation can be seen as a discrete event-based simulation, with which a dispatcher can interact:

The dispatcher may want to realise the departure of the orange train from Node 1 at 7:51 (and therefore to break the connection) or to reschedule the departure event (and therefore hold the connection). Similarly, a dispatcher can decide if they want to realise or reschedule a train arrival at a station. From this perspective, realising or rescheduling a train event to a certain time-point are the only choices a dispatcher may make today in Viriato. It is easy to see that in general there are more degrees of freedom regarding decisions a dispatcher could make. For example, a dispatcher can determine that a train should switch to another node track when it arrives in a station, in contrast to what was planned before a simulation run. All of these choices that can be made - realise or reschedule (and when), remain on a node track or switching the planned train - depend on the dispatching strategy implemented by the dispatcher. The results of a simulation run depend on the implemented dispatching strategy. The responsibility of the Viriato train simulator therefore is to ensure that the dispatcher sticks to the model constraints.

The goal of SMA's ongoing development of a Viriato train simulator prototype is to validate the macroscopic train simulation model described above by finding an appropriate software design, i.e. the interaction of a dispatcher with a macroscopic traffic simulation of a railway network. Following this we are going to continue to develop the simulator with features to increase its analysis capabilities. In future, customers will be able to implement their own dispatching strategies using the Viriato Algorithm Platform to model the dispatching behaviour in their own networks and to study their timetables in cases of deviation.