Real Time Large Railway Network Re-Scheduling

Erik Nygren[*], Christian Eichenberger[†], Emma Frejinger[‡]

July 15, 2020

Contents

Abstract

In this paper, we describe our first steps to tackle the Re-Scheduling Problem for Real-Time Large Railway Networks by a combination of techniques from operations research with different heuristics from ML and domain-specific heuristics.

The Industry State of the Art manages to resolve re-scheduling conflicts fully automatically only at narrowly defined hubs. In our approach, we aim at combining the best of two worlds: the rigor of the OR formulation with condensed experience in the form of a hypothesized Oracle. The Oracle's prediction could either narrow down the solution space (hard constraints) or speed up the solution process by strong priorities (soft constraints and solver heuristics).

[*]erik.nygren@sbb.ch

[†]christian.markus.eichenberger@sbb.ch

[‡]emma.frejinger@cirrelt.ca

We believe that the very nature of railway system allows for very strong heuristics which could allow for the problem to become tractable for large networks in real-time scenarios.

The goal of this paper is four-fold:

G1 report the decompositional problem formulation in a formal way;

G2 show the validity of the approach for one OR solver;

G3 report our first steps in tackling the Oracle;

G4 provide an extensible playground implementation for further research.

We hope that this will draw the attention of both academic and industrial researchers to find other and better approaches and collaboration across Railway companies and from different research traditions.

1 Context and Goals

1.1 Real-world Context

Switzerland has a dense railway network with both freight and passenger trains running on the same infrastructure. More than 1.2 million people use trains on a daily basis [1]. In Railway Operations, the operational schedule has to be continually re-computed because of many smaller and larger delays that arise during operations. Not all of those can be absorbed by extra times in the schedule, and if the delay has an impact on other trains, decisions on re-ordering or re-routing trains have to be taken to derive a new feasible operational plan. The industry state of the art is that delay propagation is efficiently re-computed by online IT systems. Conflicts, however, have to be most often resolved by humans by explicitly deciding on re-ordering or re-routing based on their experience. Because of the massive combinatorial complexity of these microscopic models, Operations Research models are currently only applied in very restricted, highly condensed geographic areas for re-ordering decisions but do not consider routing alternatives.

This situation is depicted in a schematic way in Figure 1: There is a common view of the current situation within system boundaries; the whole system is decomposed into disjoint geographic cells of responsibility. Most of them are handled by humans: human dispatchers have a view of the full system and can communicate through structured (e.g. IT system of incident messages) or informal ways (e.g. phone call with station managers or locomotive drivers). There are only a few condensation areas [2] (in bottleneck areas such as merging areas in front of large stations or tunnels) that are operated through automatic systems. Between these areas, trains need to be able to compensate: if trains are reordered, these decisions must be taken into account in the neighboring areas.

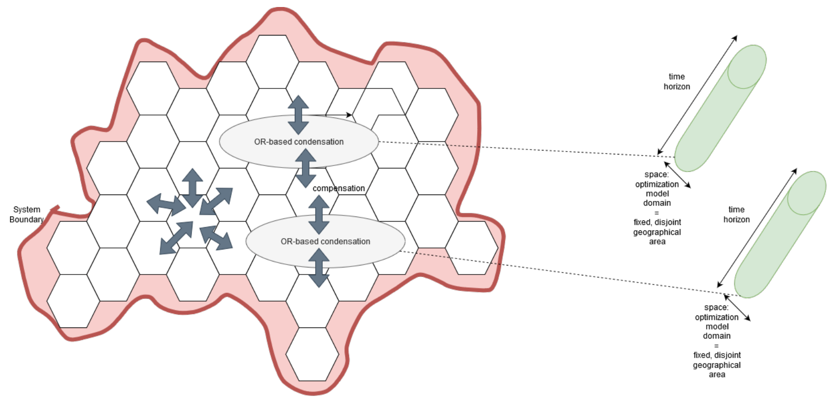

Figure 1:

1.2 Research Approach: Decomposition in Space and Time

Physical railway infrastructure is expensive in building and maintenance [3]. Therefore, the existing infrastructure capacity should be exploited as best as possible. As the number of trains operating increases and condensation areas increase in number and size, it will become increasingly difficult to compensate for decisions taken within condensation zones or to define restrictions that keep the effects on the neighboring compensation zones as predictable as possible.

We will argue that

H2 it is possible to predict the affected time-space, either from the problem structure or from historic data (see below, Section 1.3.2);

H1 such a prediction allows for a speed-up of the OR model (see below, Section 1.3.1).

To tackle this problem, our approach is to combine Operations Research and Domain-specific Learning to get the best of both worlds: an "Oracle" is able to predict the "impact" of a delay, with or without a knowledge base learnt by training; we hope the Oracle could predict which trains and which departures are or could be affected by the delay based on past decisions. This piece of information from the Oracle then helps the solver to constrain the search space or at least drive its search more efficiently (driving the branching process).

This approach is shown in Figure 2: the Oracle predicts restrictions in time and space, which are passed to the solver.

1.3 Research Approach: a Synthetic Playground

We now give a short introduction to our playground implementation (G4) and its limitation with respect to real-world features.

Synthetic Infrastructure and Simplified Resource Model [4] : Generator, no shared occupation

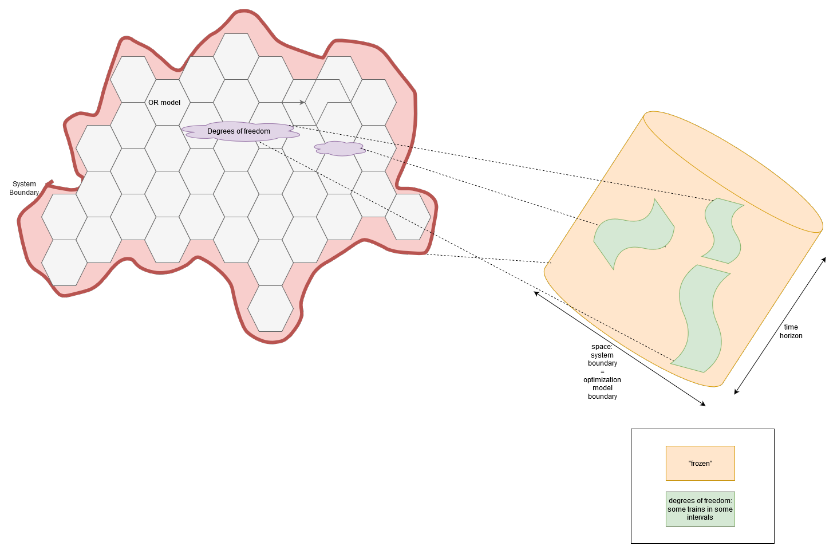

Figure 2:

Synthetic Timetable Every train has one single source and target, no intermediate stations Every train has a constant speed (may be different from train to train) no cycles in the schedule schedule is generated to minimize sum of travel times satisfying an arbitrary upper bound might be unrealistic no time reserves (no catching up) no distinction between published timetable and operational schedule (the published timetable imposes no connections or vehicle tours (turnrounds) no stops simplified train dynamics

Synthetic route alternatives 10 shortest paths (in reality, in particular in the case of disturbances affecting a whole are, we might need a different scheme knowing the parts that cannot be taken)

Simple Disturbance Model : one train stopped for d discrete time steps at time t; in reality, the delay might not be known or only probabilities can be assumed. In reality, update information comes in batches and we would need to consider multiple delays in the same update interval

1.3.1 Hypothesis 1

show speed-up in at least one implementation

1.3.2 Hypothesis 2

show some ideas, show why space and time

2 Show Pipeline H1

Infrastructure generation Assumption and difference to real world Limitations Schedule and Malfunction Generation Stochastic Model Schedule cost function and link to real world Generic Solver Model ("ScheduleProblemDescription", mathematical ASP model) Re-scheduling Full and Delta ("ScheduleProblemDescription" in these cases with pseudo-code)

3 Ideas H2

Heuristic ideas ML Ideas

4 Discussion: show links to

ML delay propagation / recourse simulation OR / heuristics decomposition approaches

5 Results H1

6 Early Results H2

References

[1] Rcs – traffic management for europe's densest rail network. controlling and monitoring with swiss precision. `https://company.sbb.ch/content/dam/sbb/de/pdf/sbb-konzern/sbb-als-geschaeftspartner/RCS/SBB_RCS_Broschuere_EN.pdf.sbbdownload.pdf`. Accessed: 2020-07-15.

[2] Gabrio C. Caimi. *Algorithmic decision suport for train scheduling in a large and highly utilised railway network*. PhD thesis, ETH Zurich, Aachen, 2009.

[3] smartrail 4.0: Ein modernisierungsprogramm der schweizer bahnbranche. `https://www.smartrail40.ch/index.asp?inc=programm/uebersicht.asp`. Accessed: 2020-07-15.

[4] Flatland challenge. multi agent reinforcement learning on trains. `https://www.aicrowd.com/challenges/flatland-challenge`. Accessed: 2020-07-15.