Real-Time Railway Traffic Management Problem: a Dynamic Decomposition Approach

Matteo Petris[1], Federico Naldini[2], Paola Pellegrini[2], Raffaele Pesenti[3]
[1] Univ. Lille, Inria, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France
`matteo.petris@inria.fr`
[2] COSYS-LEOST, Univ Gustave Eiffel, IFSTTAR, Univ Lille, F-59650 Villeneuve d'Ascq, France
`{federico.naldini,paola.pellegrini}@univ-eiffel.fr`
[3] Department of Management, Università Ca' Foscari Venezia, Cannaregio 873, 30121 Venice, Italy
`pesenti@unive.it`

Keywords : *railway traffic management, real-time scheduling, problem decomposition.*

1 Introduction

Railway services are operated following a predefined timetable. However, their execution is often perturbed by unexpected events that make this timetable infeasible. Delay caused by these events is named primary delay, and it implies that trains occupy tracks at times that are different from the planned one. Depending on traffic and track layout, these late occupations may bring to conflicts, in which at least one train must slow down or even stop to preserve safe separation. This slowing down generates secondary delay, which may quickly propagate in the network. Dispatchers can take actions to limit delay propagation, as train rerouting and rescheduling. Several optimization approaches have been proposed in the literature to tackle this problem and support dispatchers [1]. This problem is named real-time Railway Traffic Management Problem (rtRTMP). Few papers try to coordinate traffic management decisions made on several microscopic parts of infrastructures, or on a somehow obtained decomposition of the overall problem (see [3, 5]). In this paper, we propose a neighborhood-based traffic management algorithm, following to some extent the problem conception of [5].

2 Modeling principles

We denote by BS and TC the set of block sections and track-circuits composing the infrastructure, respectively. We consider a set T of n trains traveling in the network. They may use different routes to reach their final destination: the set of the available routes for train t is denoted by $\mathcal{R}_t$. A train makes a decision upon alternative routes once it has reserved the block section where the switch that gives rise to the alternatives is located. We call *route decision block section* (RDBS) a block section where such decision is to be made.

We describe railway traffic as a discrete event system. Events occur at the latest times at which route or precedence decisions must be made. They are the situations in which a train must exit a RDBS. Hereinafter, with a little abuse of notation, when we write *time* k we understand the time instant of the occurrence of event k. The *state* of the network at time $k = 0, 1, 2, \dots$ is identified by:

- the vector $p(k) = [p_t(k) : t \in T]$ of the positions of all trains, that is the last block section reserved by t;
- the set $Y(k) = \{y_{t,t',tc}(k) : t, t' \in T,\ tc \in TC\}$ of the precedences previously defined and, hence, in force between times $k-1$ and k, on common track-circuits. Specifically, a value

$y_{t,t',tc} \in Y(k)$ is set equal to 1 (respectively to 0) if train t is planned to use tc before (respectively after) train t'. If no precedence has been fixed yet, $y_{t,t',tc}$ is undefined.

At a certain time $\bar{k}$, the system is in its *final* state if all the trains reserved or crossed their final destination. Finally, given the state of the system at time k, let $S_t(k)$ be the set of block sections that must be considered in the traffic management decisions involving train t. These are the block sections *claimed* by t at k. They are the ones that t may use between $p_t(k)$ and further RDBSs.

3 Solution algorithm

Our algorithm makes decisions asynchronously to solve the rtRTMP and it is based on a dynamic decomposition of the problem. Specifically, at time k, train t for which decisions are to be made is identified. Its neighborhood is defined and the rtRTMP is locally solved by iteratively calling the optimization algorithm RECIFE-MILP [4]. The neighborhood constitutes a *sub-instance* of the overall problem. A sub-instance is defined by a pair $(Q_t(k), S(k))$, where $Q_t(k) \subseteq T$ is a subset of trains and $S(k) \subseteq BS$ is the subset of block sections claimed by the trains in $Q_t(k)$, i.e. those along which they may travel. At the first iteration, the smallest possible sub-instance is defined, including trains that may interfere with t in the very near future. RECIFE-MILP is applied to this sub-instance. If a solution is found, the algorithm stops and the solution is the *short-term strategy* for managing traffic until the next event occurs. Otherwise, the sub-instance is extended, that is $Q_t(k)$ and/or $S(k)$ are enlarged, and the next iteration starts. If no extension is possible, then the infeasibility due to the current traffic state is returned to the dispatcher, who will operate emergency operations to restore feasibility. We observe that in the worst-case scenario the sub-instance may be extended to consider the entire network and all the trains travelling on it.

This algorithm allows to decompose the overall rtRTMP problem into small sub-problems, and it enjoys an important property if the network of interest can be modeled as a series-parallel graph [2]:

Theorem 1 *The algorithm always allows to reach the final state of the railway traffic system on rail networks that can be modeled as series-parallel graph, if at all possible.*

In the paper, we will formally prove the validity of this theorem and we will provide a proof-of-concept of the applicability of the algorithm.

References

[1] V. Cacchiani, D. Huisman, M. Kidd, L. Kroon, P. Toth, L. Veelenturf, and J. Wagenaar. An overview of recovery models and algorithms for real-time railway rescheduling. *Transportation Research Part B: Methodological*, 63:15–37, 2014.

[2] Richard J Duffin. Topology of series-parallel networks. *Journal of Mathematical Analysis and Applications*, 10(2):303–318, 1965.

[3] X. Luan, B. De Schutter, L. Meng, and F. Corman. Decomposition and distributed optimization of real-time traffic management for large-scale railway networks. *Transportation Research Part B: Methodological*, 141:72 – 97, 2020.

[4] P. Pellegrini, G. Marlière, R. Pesenti, and J. Rodriguez. RECIFE-MILP: an effective MILP-based heuristic for the real-time railway traffic management problem. *Intelligent Transportation Systems, IEEE Transactions on*, 16(5):2609–2619, 2015.

[5] S. Van Thielen, F. Corman, and P. Vansteenwegen. Considering a dynamic impact zone for real-time railway traffic management. *Transportation Research Part B: Methodological*, 111:39 – 59, 2018.