An iterative algorithm for the coordinated train rerouting and rescheduling problem

Xiajie Yi[1], Grégory Marlière[3], Paola Pellegrini[2,1], Joaquin Rodriguez[3,1], Raffaele Pesenti[4]

[1] Institut de Recherche Technologique Railenium, F-59300, Famars, France
xiajie.yi@railenium.eu

[2] Univ. Gustave Eiffel, IFSTTAR, COSYS, LEOST, F-59666 Villeneuve d'Ascq, France
paola.pellegrini@ifsttar.fr

[3] Univ. Gustave Eiffel, IFSTTAR, COSYS, ESTAS, F-59666 Villeneuve d'Ascq, France
joaquin.rodriguez@ifsttar.fr

[4] Dept. of Management, Università Ca' Foscari Venezia, Cannaregio 873, 30121 Venice, Italy
pesenti@unive.it

Mots-clés : *coordination, dispatching, iterative algorithm, linear programming*

1 Introduction

This paper deals with a collaborative train rerouting and rescheduling problem faced by traffic controllers at regional railway control centers. Typically, the railway network is divided into non-overlapping control areas. Each control center coordinates several control areas. The problem we consider arises when a perturbation (i.e., an unexpected, degraded operation) occurs and the timetable cannot be operated as planned. To minimize the impact of such a perturbation, e.g., to minimize delay propagation, measures as train rescheduling and rerouting can be applied. This problem is known in the literature as real-time Railway Traffic Management Problem (rtRTMP) [1]. Several approaches have been proposed to deal with it [2]. Nevertheless, only few papers focus on the coordination of traffic management across different control areas.

In the approach we propose, for each control center, the real-time railway traffic management is hierarchically organized into two decision levels. At the lower level, dispatchers manage train schedules and routes in their own control areas. At the higher level, a coordinator ensures the compatibility of dispatchers' decisions over two or more areas. To optimize the overall system performance, the coordinator may impose constraints to dispatchers. Hereinafter, we will refer to the problems tackled at the lower and higher levels as *dispatching* and *coordination problem*, respectively.

2 Problem description

In this paper, two control areas are called *adjacent* if a train can move from one to the other without crossing any other control area. The space connecting adjacent control areas is named *coordinator space*. It is composed of *border sections* : points between bordering control areas and lines that join separated ones. Remark that several border sections can connect two adjacent control areas.

The coordinator problem consists in choosing precedences between trains using the same border sections, and possibly the specific border sections used by each train if alternative options exist. The sequence of control areas traversed by a train is considered unmodifiable (this hypothesis may be relaxed in the future).

The dispatching problem consists in finding a traffic management strategy which is compliant with the choices of the coordinator. The dispatcher is in charge of choosing precedences between trains sharing track portions (track-circuits, block sections, track sections, ... according to the model adopted for the control area dealt with), train timings and train routes within their control areas, provided their consistency with the entrance and exit precedences and locations chosen by the coordinator.

In this paper, we assume that all trains start and end their journey in a control area among the ones considered.

3 Iterative algorithm

We propose an iterative algorithm to optimize the collaborative real-time railway management problem.

At each iteration, the coordinator solves the problem of choosing precedences and locations of trains passing from a control area to an adjacent one with the objective of minimizing delay estimates. We do so by solving an integer linear programming formulation based on time-indexed binary variables. Precedences and location choices are passed to dispatchers, who solve the real-time traffic management problem in each control area to comply with them, by choosing internal precedences and routes. If a feasible solution does not exist in a control area, the dispatcher sends feedback to the coordinator in the form of cuts to be added to the formulation. In this case, a new iteration starts. If feasible solutions exist for all control areas, the set of all routes and precedences there defined constitute a detailed and complete traffic management strategy for the whole network.

However, the internally computed timings may not be coherent when complete train paths are considered. To coherently assess these timings and hence delays, we solve an overall LP in which precedences and routes are inputs. Here, track capacity constraints translate in having non-overlapping train utilization of common portions. We then pass the optimal solution and the shadow prices of the overall LP to the coordinator so as to either modify delay estimates or define cuts that exclude the current solution in the next iteration. In this way, the coordinator tries to progressively improve solution quality.

In addition to the timing and objective function assessment, the solution of the overall LP supplies the shadow prices of capacity constraints. We use these shadow prices to define cuts which will be added to the coordinator problem in the next iteration, to try to progressively improve solution quality. The whole iterative algorithm is stopped by the elapse of the available computational time.

Références

[1] Paola Pellegrini, Grégory Marlière, Raffaele Pesenti, and Joaquin Rodriguez. RECIEF-MILP : An effective MILP-based heuristic for the real-time railway traffic management problem. *IEEE Transactions on Intelligent Transportation Systems*, 16(5) :2609–2619, 2015.

[2] Wei Fang, Shengxiang Yang, and Xin Yao. A survey on problem models and solution approaches to rescheduling in railway networks. *Transportation Systems*, 16(6) :2997–3016, 2015.