Real-time train scheduling: from theory to practice

Andrea D’Ariano¹, Paolo D’Ariano¹, Marcella Samà¹, Dario Pacciarelli¹

RT-DIA-207-2013 Ottobre 2013

(1) Università degli Studi Roma Tre
Dipartimento di Ingegneria
Sezione Informatica e Automazione
Via della Vasca Navale, 79
00146 Roma, Italy

This work was partially supported by Alstom Ferroviaria SpA.
ABSTRACT

This work reports on the preliminary results of an ongoing project devoted to evaluate the practical applicability of published techniques for practical real-time train scheduling. The final goal of the project is the development of an advanced decision support system for supporting dispatchers work and for guiding them toward near-optimal real-time train re-timing, re-ordering and re-routing decisions. Specifically, this paper addresses the coupling of the ICONIS RM6 (Integrated CONtrol and Information System) product, developed by Alstom Ferroviaria S.P.A. to monitor and control railway traffic in stations and railway lines, and the optimization system AGLIBRARY (Alternative Graph LIBRARY), developed by Roma Tre University to optimize the real-time performance of railway traffic. The final aim of this project is the development of an intelligent decision support system for reducing dispatchers workload and for guiding them toward near-optimal train re-timing, re-ordering and re-routing decisions. The problem is formulated by using microscopic information on train travel times and on the status of the network, at the level of block sections and signals. The outcome of AGLIBRARY is a detailed and conflict-free train schedule, being able to avoid deadlocks and to minimize train delays. The conflict resolution procedure adopted to design a global conflict-free schedule alternates a scheduling phase with fixed routes to a search for better alternative routes. The first phase is solved by a branch and bound algorithm, truncated after a time limit of computation, while train rerouting is solved by a tabu search algorithm. The test bed, provided by Alstom Ferroviaria S.P.A., is based on a UK railway network nearby London. Computational experiments, based on instances with multiple train delays and network disruptions demonstrate that near-optimal solutions can be found by AGLIBRARY within very short computation times, compatible with real-time operations.

Keywords: Railway Operations Management, Disruption and Delay Management, Mixed-Integer Linear Programming, Branch and Bound, Tabu Search.
1 Introduction

In the scheduling literature is well known the so-called scheduling gap, i.e., the difference between the level of sophistication of the theoretical results and algorithms available in the literature and that of the methods that are employed in practice. While the theory typically address simplified problems, achieving optimal or near-optimal performance, the practice must face all the complexity of real-time operations, often with little attention to the performance level. This difference is especially evident for real-time scheduling, and train scheduling is not an exception. As a result, the poorly performing scheduling methods that are used in practice has a direct impact on the quality of service offered to the passengers, and the negative effects of disruptions on the regularity of railway traffic may last for hours after the end of the disruption [Kecmanetal13]. However, there are recently many signals that the scheduling gap could be drastically reduced in the next few years. On the theoretical side, recent approaches to train scheduling tend to incorporate an increasing level of detail and realism in the models while keeping the computation time of the algorithms at an acceptable level. On the practical side, the railway industry is interested in assessing the suitability of these methods to the practical needs of real-time railway traffic management.

This paper reports on the preliminary results of an ongoing project focusing on the evaluation of the practical applicability of advanced scheduling techniques for real-time train scheduling. The final goal of the project is the development of an advanced decision support system for supporting dispatchers’ work and for guiding them toward near-optimal real time train re-timing, re-ordering and re-routing decisions. The Traffic Management System (TMS) evaluated in this work is composed by two sub-systems: the ICONIS RM6 (Integrated CONtrol and Information System) product, developed by Alstom Ferroviaria S.P.A. to monitor and control railway traffic in stations and railway lines, and the optimization system AGLIBRARY (Alternative Graph LIBRARY), developed by Roma Tre University. The scheduling algorithms employed by AGLIBRARY are the optimization core of the overall system and focus on the optimization of the real-time performance of railway traffic.

The structure of the next sections is the following: Section 2, Train dispatching approaches, describes the related literature and the alternative graph model of the train dispatching problem without and with rerouting. This model is also translated into a Mixed Linear Integer Problem (MILP) formulation. The general architecture of AGLIBRARY system is then provided in terms of train scheduling and routing algorithms. In Section 3, Computational experiments, the preliminary results for the East Coast Main Line (i.e. a UK railway network nearby London) are reported to compare AGLIBRARY with a commercial solver (CPLEX) in terms of solution quality and computation time.
2 Train dispatching approaches

2.1 State-of-the-art

Despite year 2013 celebrates the fortieth anniversary of the first research paper on train scheduling [Szpigel73], the study of the real-time aspect of the problem received rather limited attention in the literature. Moreover, most of existing approaches solve very simplified problems that ignore the constraints of railway signalling, and that are only applicable for specific traffic situations or network configurations (e.g., a single line or a single junction), see, e.g., the surveys of [Cordeau98, Ahujaetal05, Törnquist06, HansenPachl08, D'Ariano08, Lusbyetal11]. Among the reasons for this gap between early theoretical works and practical needs are the inherent complexity of the real-time process and the strict time limits for taking decisions, which leave small margins to a computerized Decision Support System (DSS).

Effective DSSs must be able to provide the dispatcher with a conflict-free disposition schedule, which assigns a travel path and a start time to each train movement inside the considered time horizon and, additionally, minimizes the delays (and possibly the main broken connections) that could occur in the network. The main pre-requisite of a good DSS is the ability to deal with actual traffic conditions and safety rules for practical networks. In other words, the solution provided by a DSS must be feasible in practice, since the human dispatcher may have not enough time to check and eventually adjust the schedule suggested by the DSS. A recognized approach to represent the feasibility of a railway schedule is provided by the blocking time theory (acknowledged as standard capacity estimation method by UIC in 2004), which represents a safe corridor for the train movements in the railway network with the so-called blocking time stairways.

With the blocking time theory approach, the schedule of a train is individually feasible if a blocking time stairway is provided for it, starting from its current position and leaving each station (or each other relevant point in the network) not before the departure time prescribed by the timetable. A set of individually feasible blocking time stairways (one for each train) is globally feasible if no two blocking time stairways overlap. The timetable prescribes the set of trains that are expected to travel in the network within a certain time window, the stops for each train and a pair of (arrival, departure) times for each train and each stop. At other relevant points (e.g., at the exit from the network) can be defined minimum and/or maximum pass through times.

Many models and algorithms for train dispatching have already been proposed in the literature, but only a few of them with successful application in practice. So far, the most successful attempt in the literature to incorporate the blocking time theory in an optimization model is based on the alternative graph model introduced by [MascisPacciarelli02]. Effective applications to real-time train scheduling are reported by [D'Arianoetal2007a, MazzarelloOttaviani07, ManninoMascis09]. Different promising approaches based on MILP formulations are reported by [TörnquistPersson07, Rodriguez07, Caimietal12].

The alternative graph model allows to directly model the individual and global train
schedule feasibility concepts expressed by the blocking time theory, thus enabling the detailed recognition of timetable conflicts in a general railway network with mixed traffic for a given look-ahead horizon, even in presence of heavy disturbances and network disruptions. The feasibility of the rescheduling solutions is ensured by the explicit representation of train blocking times in the model. Several later studies have confirmed the ability of the model to take into account different practical needs, such as train delays, travel times, passenger transfer connections and energy consumption [Cormanetal09, Cormanetal12], train re-routing [D'Arianoetal08, Cormanetal10], traffic coordination in different dispatching areas [Cormanetal12]. Clearly, an alternative graph formulation can be easily translated into a mixed integer program, and then solved with a commercial software. However, specialized solution algorithms can be developed, which allow to find a feasible, or even optimal, solution in a shorter computation time. A set of specialized algorithms based on the alternative graph formulation is included in AGLIBRARY. This dispatching system includes solution algorithms ranging from fast heuristic procedures that can be chosen by the user to sophisticated branch and bound algorithms based on [D'Arianoetal07a]. AGLIBRARY has been validated for various case studies provided by the Dutch infrastructure manager ProRail (the railway networks Leiden-Schiphol-Amsterdam;Utrecht-Hertogenbosch;Utrecht-Hertogenbosch-Nijmegen-Arnhem) and by the Italian infrastructure manager RFI (the line Campobasso-Nettuno), even if in principle the software can tackle any national or international standard based on the stairway concept or on the ERTMS moving block concept.

2.2 Alternative graph formulation

The AG formulation of the train scheduling problem with fixed routes (i.e., in which the route is prescribed and cannot be changed) is a triple $G = (N, F, A)$ where $N$ is a set of nodes, $F$ is a set of fixed arcs and $A$ a set of pairs of alternative arcs. The problem is thus based on two types of constraints:

1. Fixed constraints model the individual feasibility of a train schedule, i.e., the blocking time stairway. Each variable $t_i$ is associated to the entrance of a train in a resource (block section, platform of station route). The schedule is individually feasible if the entrance in a resource is at least $p$ time unit after the entrance in the former resource, where $p$ is the traversing time of the previous resource. Moreover, if at the current time the train occupies a certain resource, it cannot enter the next resource in its route before the time needed to traverse the remaining part of the current resource. Finally, since the timetable prescribes a departure (or a pass through) time for the train at each relevant point in its route, the train cannot enter the next resource of the route before the minimum (after the maximum) prescribed time of the relevant point. Each fixed constraint is associated with an arc $(u, v) \in F$ and has weight $f_{uv}$.

2. Alternative constraints model the global feasibility of a set of blocking time stairways (one for each circulating train). Given a resource traversed by two trains, the second train cannot enter the resource before the entrance of the previous train plus its blocking time, i.e., the time interval in which the resource is reserved for the previous train.
If a precedence constraint has not been fixed between the two trains on that resource (either by the timetable, or by the dispatcher, or by the physical topology of the network), then two orderings are possible and one of them has to be chosen in a solution. This fact is represented naturally in the alternative graph by defining a pair of constraints, one of which must be chosen in a solution. Alternative arcs \(((k, j), (h, i)) \in A\) model aircraft sequencing decisions, each one with its associated weight \(a_{kj}\) and \(a_{hi}\).

Each constraint of the alternative graph is in the form of a precedence between two time events, therefore can be modelled as a directed arc of a graph in which nodes are associated to events and arcs to constraints, as in [D'Ariano et al. 07a, 07b]. Alternative constraints are grouped in alternative pairs. One arc from each pair has to be chosen in a solution. Letting \(F\) and \(A\) be the set of fixed and alternative constraints (arcs), \(t_0\) be the starting time of traffic prediction, and letting \(t_i, i=1,\ldots,*\), be the set of variables, the alternative graph formulation is as follows:

\[
\begin{align*}
\min & \quad t_* - t_0 \\
\text{s.t.} & \quad t_v - t_u \geq f_{uv} \\
& \quad (t_j - t_h \geq a_{hj}) \text{ OR } (t_i - t_h \geq a_{hi}) \\
& \quad ((k,j),(h,i)) \in A
\end{align*}
\]

The alternative graph allows to easily and efficiently check the feasibility of a solution (i.e., a solution is feasible if there are no positive length cycles, which corresponds to an event strictly preceding itself), as well as the quality of a solution.

Figure 1 shows a simple network with four trains (A, B, C, D). The events to be considered are the entrance time of train A in resources 4-5-7-8-out, train B in resources 2-4-5-7-8-out, train C in resources 9-7-6-4-3-1, and train D in resources 7-6-4-3-1.

![Figure 1: Example of traffic situation](image)

The four train schedules are individually feasible if each variable satisfies the fixed constraints depicted in the event graph of Figure 2 (diagonal arcs are forced by the network topology). Here, each horizontal arc is weighted with the traversing time of the associated resource, diagonal arcs are weighted with the blocking times, arcs outgoing node 0 are weighted with the remaining traversing time of the current resource and arcs entering node * are used to compute the quality of a schedule (still infeasible so far since conflicts between red and green trains are still possible).
The alternative graph formulation in Figure 3 includes the eight alternative pairs of arcs needed to ensure global feasibility, i.e., to solve potential conflicts among red and green trains on block sections 4 and 7. The two alternative arcs in each pair are depicted with the same color. The weight on each alternative arc equals the associated blocking times minus the traversing time of the associated resource. More details are reported in e.g. [Cormanetal09,10,12a,12b, D’Arianoetal07a,07b,08].

A solution is obtained by choosing an arc from each alternative pair in such a way that the resulting graph contains no positive length cycles. The graph of a feasible schedule given in Figure 4. The train schedule is then obtained by assigning to each variable \( t_i \) the length of the longest path in the graph from node 0 to node \( i \).

By adding suitable weights on the arcs entering node *, the length of the longest path from 0 to * may represent a max-type performance of the schedule. For example, if \( D_{A_{\text{out}}}, D_{B_{\text{out}}}, D_{C_{1}}, D_{D_{1}} \) are the prescribed pass through times at the exit of the last resource traversed by each train, the weights \(-D_{A_{\text{out}}}, -D_{B_{\text{out}}}, -D_{C_{1}}, -D_{D_{1}}\) for arcs \((A_{\text{out}}, \ast), (B_{\text{out}}, \ast), (C_{1}, \ast), (D_{1}, \ast)\)
respectively, allow to represent the maximum delay of the schedule as the length of the longest path from 0 to * in a train schedule. Clearly, the individual delay at any subset of the nodes can be obtained and combined to form a different objective function. The formulation can be extended to include alternative routes for each train. A route in the railway network is then associated with a path of fixed arcs in the graph from node 0 to node *. Several paths are therefore associated with each train, and once a specific path is selected for each train, it is possible to formulate the alternative graph described above, as in [D’Ariano et al. 08, Cormano et al. 10].

### 2.3 MILP formulations

A natural MILP formulation of the problem with fixed routes can be obtained from the alternative graph formulation by translating each alternative pair into a pair of constraints and by introducing a binary variable representing the choice of a constraint (M is a very large number, e.g. the sum of all arc weights):

\[
\begin{align*}
\min & \quad t_* - t_0 \\
\text{s.t.} \quad & t_v - t_u \geq f_{uv} & (u,v) \in F \\
& t_j - t_k + Mx_{khi} \geq a_{kj} \\
& t_i - t_h + M(1-x_{khi}) \geq a_{hi} & ((k,j),(h,i)) \in A \\
& x_{khi} \in \{0,1\}
\end{align*}
\]

Here, \(x_{khi}=1\) means that arc \((h,i)\) has been selected from pair \(((k,j),(h,i))\), i.e., constraint \(t_j - t_k + Mx_{khi} \geq a_{kj}\) is always satisfied. Variable \(t_i\), \(i=1,\ldots,*,\) represents the starting time of the \(i\)-th operation. An operation corresponds to the entrance of the associated train in the associated resource. \(t_0\) represents the starting of the prediction, also called \(t_{now}\).

The formulation can be extended to the problem with routing flexibility by enlarging sets \(F\) and \(A\) to contain all possible arcs for all possible train routes, and by adding for each alternative route variables \(y_{eb} \in \{0,1\}\) and \(y_{cd} \in \{0,1\}\) equal to 1 if route \(e\) is chosen for train \(b\) (resp., if route \(c\) is chosen for train \(d\)), and 0 otherwise. In this case, alternative arcs are associated to all resources shared by routes \(e\) and \(c\). Letting \(n_T\) be the number of trains and \(rb\) be the number of routes of train \(b\), the formulation becomes:

\[
\begin{align*}
\min & \quad t_* - t_0 \\
\text{s.t.} \quad & t_v - t_u + M(1-y_{eb}) \geq f_{uv} & (i,j) \in F \\
& t_j - t_k + M(1-y_{eb}) + M(1-y_{cd}) + Mx_{khi} \geq a_{kj} \\
& t_i - t_h + M(1-y_{eb}) + M(1-y_{cd}) + M(1-x_{khi}) \geq a_{hi} & ((k,j),(h,i)) \in A \\
& \sum_{e=1,\ldots,rb} y_{eb} = 1 \\
& y_{eb}, y_{cd} \in \{0,1\} & b=1,\ldots, n_T
\end{align*}
\]

In principle the MILP formulation can be easily modified. In fact, the quality of a schedule may involve several indices reflecting the interests of the different actors involved in railway traffic management, such as the train punctuality, the utilization level of railway resources, the costs incurred by different train operating companies in terms of delays, broken connections and energy consumption, and so on. All these indices, if expressed in terms of operation starting times \(t_i\), \(i=1,\ldots,*,\) are taken or can be taken into account in the construction of the MILP formulation through the definition e.g. of a suitable objective
function.

2.4 AGLIBRARY system

The traffic control procedure implemented in AGLIBRARY computes a first feasible schedule in which each train follows its default route. Then, the procedure looks for better solutions, in terms of delay minimization, by changing the route for some trains. Its architecture is described in Figure 5. Specifically, AGLIBRARY combines the Branch and Bound (BB) algorithm of [D’Ariano et al. 07a] and the Tabu Search (TS) algorithm of [Corman et al. 10] for solving the train scheduling and re-routing problem.

![Figure 5: Train scheduling and re-routing scheme [D’Ariano08]](image)

3 Computational experiments

The algorithms presented in the previous sections have been tested on a Intel Core 2 Duo E6550 (2.33 GHz), 2 GB di RAM, Windows XP. The MILP formulation is solved by IBM LOG CPLEX MIP 12.0. The test bed is a line nearby the city of London (see Figure 6). In this campaign of experiments the solvers have to deal with strongly disrupted traffic situations in which some trains have speed restrictions and others are re-routed.

![Figure 6: Microscopic layout of the East Coast Main Line (source: ICONIS)](image)

Table 1 presents average data on 29 instances: 10 small (15-minute), 9 medium (30-
minute) and 10 large (60-minute). We note that a small instance is not necessarily an easy instance since its complexity depends on the ordering and routing variables.

Table 1: Quantitative information on the tested instances

<table>
<thead>
<tr>
<th>Instance Type</th>
<th>Time Horizon</th>
<th>Number of resources per train</th>
<th>Number of viable routes</th>
<th>Number of trains (jobs)</th>
<th>Number of alternative pairs</th>
<th>Number of arcs</th>
<th>Number of nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>15</td>
<td>18.6</td>
<td>11.0</td>
<td>36.1</td>
<td>1136.0</td>
<td>3852.3</td>
<td>1331.2</td>
</tr>
<tr>
<td>Medium</td>
<td>30</td>
<td>15.8</td>
<td>14.8</td>
<td>45.9</td>
<td>3408.3</td>
<td>9454.8</td>
<td>2249.3</td>
</tr>
<tr>
<td>Large</td>
<td>60</td>
<td>11.3</td>
<td>23.6</td>
<td>65.1</td>
<td>12946.7</td>
<td>30791.1</td>
<td>4216.6</td>
</tr>
</tbody>
</table>

In what follows, the results on the three set of instances are reported comparing the MILP formulation solved by CPLEX within 1 hour of computation (let to the algorithm, i.e., excluding the time needed to generate the formulation) and the alternative graph formulation solved by AGLIBRARY within 30 seconds of computation (let to the solver, i.e., including the time needed to generate the formulation):

- MILP formulation solved by IBM LOG CPLEX MIP 12.0:
  6 fails, 22 optimum, avg computation time (algorithm) best solution 1011.7 sec.
- AGLIBRARY: Branch & Bound [D’Ariano et al. 07], Tabu Search [Corman et al. 10]:
  0 fails, 21 optimum, avg computation time (algorithm) best solution 9 sec.

In Figures 7, 8 and 9, detailed results are shown regarding the comparison AGLIBRARY versus CPLEX. For each optimal solution (i.e. minimum maximum consecutive delay), an asterisk is inserted in Figure 7 nearby the numerical value (in seconds). Every time CPLEX does not find a solution in the given computation time a failure is shown. The time to compute the best solution by each algorithm or solver are reported for each instance in Figures 8 and 9. We note that the solver time includes the computation time of the algorithms plus all other times required to process the problem instance.

Figure 7: CPLEX versus AGLIBRARY: solution quality
From the results on this set of instances, we conclude that AGLIBRARY computes near-optimal solutions in a short time of computation, compatible with real-time application.

4 On-going research

We are currently investigating a number of possible system improvements, including:

- Formulation and impact of additional problem constraints;
- Different objective functions (e.g. number of late trains, weight of broken connections, passenger delay minimization) and their combinations;
- Advancement of the train scheduling and routing algorithms (e.g. for dealing with specific disruption scenarios) in terms of reduced computation time and better solution
quality (with respect to various performance indicators);
• Study of alternative MILP formulations and MILP-based solution approaches;
• Extensions of the model by incorporating further relevant practical aspects.

References


