The track formulation for the Train Dispatching problem

Leonardo Lamorgese, Carlo Mannino

Applied Mathematics
SINTEF ICT
Oslo, Norway

Abstract

With few exceptions, train movements are still controlled by human operators, the dispatchers. They establish routes and precedence between trains in real-time in order to cope with normal operations but also to recover from deviations from the timetable, and minimize overall delays. Implicitly they tackle and solve repeatedly a hard optimization problem, the Train Dispatching Problem. We recently developed a decomposition approach which allowed us to solve real-life instances to optimality or near optimality in times acceptable for dispatchers. We present here some new ideas which appear to significantly reduce computational times while solving to optimality even large instances.

Keywords: Rescheduling, dispatching, disjunctive formulation

1 Introduction

Railways all over Europe are becoming ever more congested and punctuality and reliability are rapidly deteriorating. Still, in Europe the volume of people

1 Email: leonardo.lamorgese(carlo.mannino)@sintef.no
and freight transported on railway is increasing (see [3]), increasing congestion evermore. A fast and economically viable way to increase capacity is by improving the efficiency of daily operations so as to be able to control a larger number of running trains. Today, nearly all train traffic control is carried out “by hands”, namely by human operators called dispatchers. When trains fail to follow the official timetable, dispatchers establish new routes and new schedules in real-time trying to restore the original timetable, or to approximate it as much as possible. We call the resulting optimization problem the Train Dispatching Problem (TD).

In a by now classical approach to this type of problems, the movements on the rail of each train are represented as a path (route). The coordinated movements of all trains are then further represented in a single graph which contains all routes. To avoid conflicts on the use of the resource, decisions on ”which train goes first” are represented in this model by disjunctive arcs (see [6]). The TD problem is finally represented as a (typically very large) disjunctive optimization problem. The latter may be very hard to solve in practice and typically one resorts to heuristic approaches.

In [4],[5] we introduce a decomposition approach which allows to solve the TD problem to optimality (or nearly) for a number of real-life instances from the Italian railway system. We identify two sub-problems, the Line Dispatching problem (LD) and the Station Dispatching problem (SD), which naturally arise when controlling a railway line. Both are modeled as Mixed Integer Linear Programs (MILPs) and the overall TD problem is solved in a master-slave fashion. In particular, the Line Dispatching problem acts as the master problem which is solved iteratively; at each iteration a solution to the Station Dispatching problem is generated (for every station of the line) and, if necessary, new constraints (in the variables of the master problem) are added to the Line Dispatching problem which is then resolved again. This approach allows to drastically reduce the number of variables with respect to the ”general” disjunctive formulation and model station layouts with increasing complexity.

In [4], [5] the major decision variables of the master problem are associated with every pair of trains and every point on the line where trains can meet or pass each other. This choice mimics the actual behaviour of dispatchers which take such decisions in order to avoid conflicts. In this paper we introduce a new modeling approach which allowed us to further reduce the computational effort and solve previously unsolved instances. In particular, the new variables are somehow complementary to the original ones, in the sense that they model where a pair of trains will not meet on the line. We assume that the railway resources of a line can be ordered so that any train runs through either
(monotonically) increasing or (monotonically) decreasing resources. It follows that, for any given pair of trains, if they do not meet in a specific region \( p \), then one of the following conditions must be satisfied i) the trains meet before \( p \); ii) the trains meet after \( p \). This simple idea actually allowed for several algorithmic enhancements. In particular, we devised a delayed row generation mechanism which is a natural extension of this definition and corresponds to generating recursive partitions of the line. Preliminary computational results appear very promising.

2 The Train Dispatching problem on single-track lines

Railway networks can contain very complex stations and several parallel tracks between stations, plus other infrastructures such as sidings and cross-overs. However in this paper we focus on single-tracks lines, where stations are connected by one track and trains in both directions will alternate on it. To understand the relevance of such lines, consider (e.g.) that in Italy they represent almost 60% of the entire network, and in Norway almost 95%! We also use a simple model for the stations, which applies in most cases and to all stations considered in our real-life test-bed. In any case, the concepts here discussed for single-track lines and simple stations can be readily extended to multiple tracks and more complex stations.

2.1 Single-Track lines

A single-track line is an alternating sequence of stations and tracks. Let \( S = \{1, \ldots, q\} \) be the set of stations and \( B = \{1, \ldots, q-1\} \) be the set of tracks, with track \( k \) connecting station \( k \) and station \( k+1 \). At most one train at the time can occupy a track, while each station \( s \in S \) can accommodate up to \( c_s \) trains, where \( c_s \) is the number of platforms in the station (or station capacity).

Let \( R = S \cup B \) be the set of railway resources (in our case stations and tracks) and let \( T \) be the set of trains. The route of train \( i \in T \) is a sequence of stations and tracks, namely a directed path \( G^i = \{v_1^i, (v_1^i, v_2^i) \ldots, (v_{l(i)}^i, v_{l(i)}^i)\} \), with node set \( V^i \) and arc set \( A^i \). Node \( v \in V^i \) represents the occupation of a resource in \( R \), either a station or a track. Arc \( (u,v) \in A^i \) models that train \( i \) visits \( u \) right before \( v \) and we let \( L_{u,v}^i \geq 0 \) be the minimum time to go from \( u \) to \( v \). So, if \( u \) is a station (node), then \( v \) is the next track on the route and \( L_{u,v}^i \) is the minimum time train \( i \) spends in the station before departing. If \( u \) is a track (node), then \( L_v^i \) is the track’s running time for train \( i \).
2.2 The graph of routes

The train routes are exploited in the construction of a new "logical" graph, which represents the movements of all trains. We call this graph the graph of routes $G_T = (V, A)$, which is the union of each route graph $G^i$ plus an additional root node. Namely, we let $V = \{r\} \cup \{v \in V^i : i \in T\}$ and $A = \{(r, v^i_i), i \in T\} \cup \{(u,v) \in A^i : i \in T\}$. The new node $r$ is a source, connected to the first node of each train route $G^i$, and is used to model the start of the time horizon. Now, we introduce the schedule $t \in \mathbb{R}_{+}^V$. If $v \in V$ corresponds to node $v^i_k$ in the route of train $i$, then $t_v$ is the earliest time at which train $i$ can reach the $k$-th resource on its path. Also, we let $t_r = 0$. Indeed, if $v$ corresponds to a station on route $G^i$, then $t_v$ represents the earliest arrival time for train $i$ in such station. Similarly, if $v$ is a track on route $G^i$, then $t_v$ represents the departing time from the station that precedes the track on $G^i$. With every possible schedule $t$ we associate a cost function $c(t)$, which we chose (after discussions with railway engineers) to be a convex and piece-wise linear function of arrival delays, so that larger delays have non-decreasing marginal cost. We are now able to state more formally the Train Dispatching problem for single-track lines:

**Problem 2.1** Given a set of stations $S$, a set of tracks $B$, and a set of trains $T$ find a schedule $t$ for the trains in $T$ so that $c(t)$ is minimized, no two trains occupy simultaneously the same track and no more than $c_s$ trains meet in station $s \in S$.

3 Modelling the Train Dispatching problem

We assume hereby that the set of single-track railway resources $R = S \cup B$ can be ordered so that any train runs through either (monotonically) increasing or (monotonically) decreasing resources. Given trains $i, j \in T$, we will refer to $i$ meeting $j$ before (after) a given railway resource $r \in R$ if $i$ and $j$ meet in any railway resource $r' \in R$ such that $r' \prec r$ ($r' \succ r$). Accordingly, we have an increasing direction (when trains run from smaller resources to larger ones) and a decreasing direction.

We model now that trains cannot meet on a track $b$, that is they meet either before $b$ or after $b$. Let train $i$ run in increasing direction, and train $j$ run in decreasing direction. Assume they meet before track $b$. Then train

---

2 This is always the case in single-track lines. However, for more complex lines, we may content ourselves with a milder assumption as, indeed, it suffices that, for any given pair of trains, the property holds for the common sub-network.
\( j \) arrives in station \( b \) before train \( i \) leaves it, and the precedence constraint \( t_v - t_u \geq 0 \) holds, where \( t_v \) and \( t_u \) are, respectively, the exit and arrival times of \( i \) and \( j \) in station \( b \). Similarly, the constraint \( t_q - t_p \geq 0 \) holds if \( i \) and \( j \) meet after \( b \), with \( t_q \) and \( t_p \) being, respectively, the exit and arrival times of \( j \) and \( i \) in station \( b + 1 \). So, every feasible schedule must satisfy the disjunction: \( t_v - t_u \geq 0 \lor t_q - t_p \geq 0 \). Similar disjunctions hold for pair of trains running in other combinations of directions. Disjunctive precedence constraints can be represented on the graph of routes \( G^T \) by a pair of directed arcs \( \{\alpha(i,j,b), \beta(i,j,b)\} \), where \( \alpha(i,j,b) = (p,q) \) and \( \beta(i,j,b) = (u,v) \), with \( p,q,u,v \in V \); such pair is called disjunctive arc (See (1)).

For every pair of distinct trains \( \{i,j\} \) and every track \( b \in B \) we introduce a binary variable \( w_{ij}^b \) which is 1 if \( i \) and \( j \) meet before \( b \) and 0 if \( i \) and \( j \) meet after \( b \). It is well known that by introducing a suitable large coefficient \( M \) the above disjunction can be expressed by a (conjunctive) pair of linear constraints \( t_v - t_u \geq M(w_{ij}^b - 1) \) and \( t_q - t_p \geq -Mw_{ij}^b \).

Now, recall that no more than \( c_s \) trains can meet simultaneously in a station \( s \in S \). To express this constraint we introduce for every pair of distinct trains \( i,j \) and every station \( s \in S \) a binary variable \( y_{ij}^s \) which is 1 if \( i \) and \( j \) meet in \( s \) and 0 otherwise. In [5] we show that the station capacity constraint is satisfied if and only if, for every subset of trains \( K \subseteq T \), with \( |K| = c_s + 1 \), the following constraint holds:

\[
\sum_{(i,j) \in K} y_{ij}^s \leq \frac{1}{2}(c_s + 1)c_s - 1
\]

Clearly, trains \( i \) and \( j \) meet in \( s \in S \) if and only if they meet after track \( s - 1 \) and before track \( s \). This can be expressed by the linear constraint \( y_{ij}^s \geq w_{ij}^s - w_{ij}^{s-1} \). So, the TD problem for single-track lines writes as shown in (2).

Note that since \( c(t) \) is piece-wise linear and convex, it can be easily linearized. The formulation can be strengthened in various ways. First, observe that if train \( i \) and \( j \) meet before \( b \), then they meet before \( b + 1, b + 2, \ldots \). Then, for \( \{i,j\} \subseteq T \) and \( k \in B \), we have the following valid constraints (not obtainable as conic combinations of other constraints): \( w_{ij}^k - w_{ij}^{k-1} \geq 0 \). Also, since \( w_{ij}^k - w_{ij}^{k-1} \geq 0 \), then it is not difficult to see that for every feasible integral \( w,y \) we have \( y_{ij}^k = w_{ij}^k - w_{ij}^{k-1} \). Then the variable \( y_{ij}^k \) can be dropped from the formulation. Again, observe that this equality holds necessarily only for integral values, that is the equation is not directly implied by the other linear inequalities.

The above MILP is the basis of our master-slave algorithm for solving the
\[
\begin{align*}
\min \quad & c(t) \\
\text{s.t.} \quad & (i) \quad t_v - t_u \geq L_{uv} \quad (u, v) \in A \\
& (ii) \quad t_v - t_u \geq M(u^{ij}_b - 1) \quad \{i, j\} \subseteq T, b \in B, (u, v) = \beta(i, j, b) \\
& (iii) \quad t_q - t_p \geq -Mw^{ij}_b \quad \{i, j\} \subseteq T, b \in B, (p, q) = \alpha(i, j, b) \\
& (iv) \quad y^ij_s \geq w^{s}_{ij} - w^{s-1}_{ij} \quad \{i, j\} \in K, s \in S \setminus \{1\} \\
& (v) \quad \sum_{\{i,j\} \subseteq K} y^ij_s \leq \frac{1}{2}(c_s + 1)c_s - 1 \quad s \in S, K \subseteq T, |K| = c_s + 1 \\
& (vi) \quad t \in \mathbb{R}^V, w, y \text{ binary}
\end{align*}
\]

TD problem, and is solved by delayed row generation. Namely, we start by solving a problem with a subset of the overall constraints and we generate more constraints only if necessary. We follow a path somehow intermediate between the classical Benders’s decomposition algorithm (see, e.g., [7]) and the combinatorial variant suggested by Codato and Fischetti[2]. We limit to separating constraints from (ii) to (v) only in branching nodes corresponding to integer solutions. In this way, the separation of (v) becomes easy (see [5]) as it reduces to computing a maximum clique on an interval graph.

4 Preliminary Computations

Our tests were run on an Intel(R) Core(tm) i7-2640M CPU 870 2.80GHz machine and we used the commercial solver ILOG CPLEX. Although a very preliminary implementation, results show that this approach allows to successfully tackle the Train Dispatching problem for single-track instances and find solutions within the stringent time window required by a real-time application. In our computations we solved real-life instances from the Trento-Bassano (T-BG), Terontola-Foligno (T-F), Foligno-Orte (F-O) and Falconara-Foligno (F-F) lines in Italy. In Table 1 we confront the performance of the new formulation against the precedent ”station meeting” formulation (see [5]) for nine instances of T-BG. TC stands for Track Conflict, while SM stands for Station Meeting. Results show that TC appears competitive with the former. Indeed, computation time is reduced significantly, as computing optimal solutions for TC requires separating and adding, in the general case, a far smaller set of
In Table 2, we also compare our new approach with the dispatching algorithm currently in operation on the lines in Table 2 in terms of the average distribution of delayed trains. Trains were clustered in macro-ranges according to the difference between expected and actual arrival time at destination.

As emerges from Table 2, in all cases the percentage of trains on time increased tangibly with respect to the current practice. This was not only due to slightly delayed trains arriving on time, but also to a clear decrease in the number of trains running severely late. Also, TC’s performance is only very slightly affected by larger instances, as it is still able to produce, in the vast majority of cases, high quality solutions within the time limit. In conclusion, these preliminary results appear very promising and confirm evermore the potential impact of effective exact methods on Train Dispatching.

3 Actually, two of these are mixed single/double-track lines but the extending the model was straightforward
<table>
<thead>
<tr>
<th>Line</th>
<th>#</th>
<th>Horizon (hrs)</th>
<th>Trains</th>
<th>On time</th>
<th>Late between 3 and 6 mins</th>
<th>Later than 6 mins</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>H</td>
<td>TC</td>
<td>H</td>
</tr>
<tr>
<td>F-O</td>
<td>318</td>
<td>12</td>
<td>43</td>
<td>85%</td>
<td>94%</td>
<td>9%</td>
</tr>
<tr>
<td>T-BG</td>
<td>365</td>
<td>9</td>
<td>31</td>
<td>64%</td>
<td>86%</td>
<td>15%</td>
</tr>
<tr>
<td>T-F</td>
<td>634</td>
<td>11</td>
<td>32</td>
<td>65%</td>
<td>90%</td>
<td>25%</td>
</tr>
<tr>
<td>F-F</td>
<td>834</td>
<td>12</td>
<td>44</td>
<td>75%</td>
<td>86%</td>
<td>10%</td>
</tr>
</tbody>
</table>

Table 2
Punctuality distribution for four lines in Italy. Average figures. H stands for Heuristic. Column "#" refers to the number of instances solved.

References


