Zuse Institute Berlin

FRANK FISCHER[1]
BORIS GRIMM[2]
TORSTEN KLUG[2]
THOMAS SCHLECHTE[2]

A Re-optimization Approach for Train Dispatching

[1] University of Kassel, Algorithmic Algebra and Discrete Mathematics, Germany frank.fischer@uni-kassel.de
[2] Zuse Institute Berlin (ZIB), Takustr. 7, D-14195 Berlin, Germany {grimm, klug, schlechte}@zib.de

A Re-optimization Approach for Train Dispatching

Frank Fischer, Boris Grimm, Torsten Klug, and Thomas Schlechte

Frank Fischer
Universität Kassel,
Email: frank.fischer@uni-kassel.de

Boris Grimm, Torsten Klug, Thomas Schlechte
Zuse Institute Berlin, Takustr. 7, 14195 Berlin,
Email: grimm@zib.de, klug@zib.de, schlechte@zib.de

Summary. The Train Dispatching Problem (`TDP`) is to schedule trains through a network in a cost optimal way. Due to disturbances during operation existing track allocations often have to be re-scheduled and integrated into the timetable. This has to be done in seconds and with minimal timetable changes to guarantee smooth and conflict free operation. We present an integrated modeling approach for the re-optimization task using Mixed Integer Programming. Finally, we provide computational results for scenarios provided by the INFORMS RAS Problem Soling Competition 2012.

1 Introduction

The Train Dispatching Problem (`TDP`) deals with the determination of a railway timetable by constructing train routes and corresponding arrival and departure times to operate train requests in a given railway network. Due to the complex operation rules, limited capacity, which is only upgradeable with massive financial effort, the infrastructure network builds a natural bottleneck. Thus, it is appreciable to utilize the existing infrastructure in the best way.

The `TDP` integrates several major requirements like safety system rules, train characteristics, blocking and headway times, timetable requirements, and infrastructure capacities. A detailed problem description and a Mixed Integer Programming formulation to solve this problem is described in detail in [5]. In this paper, we report on a *Re-optimization* or *Re-scheduling* approach for the `TDP` in a real time setting using a state-of-the-art MIP solver.

A general concept of re-optimization for any optimization problem is illustrated in Figure 1. The authors of [2] introduced this approach in the context of rolling stock rotation planning problems. In case of the `TDP`, we adapted it as follows: At some point in time a railway undertaking has to agree on a timetable, ideally, utilizing an optimization algorithm or by manual planning, see on the left hand side the box *Original Problem* and the box *Reference Solution*. Later in time this problem or aspects changes that much, such that the reference solution, in case of the `TDP` the timetable, becomes infeasible. Thus, a modified problem has to be solved. In contrast to the first process leading to the reference timetable the time limitations are in the second stage rather strict. Since an operator has only minutes or seconds for his decisions, the re-optimization algorithm has to calculate solutions within a real time
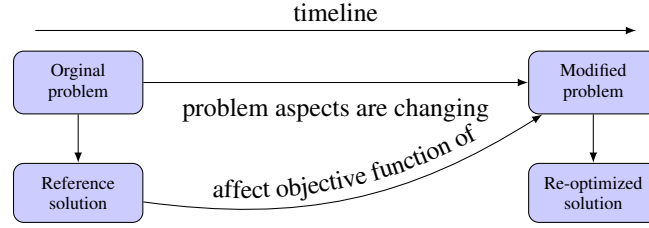
Fig. 1. Generalizing the concept of Re-optimization from [2].

management system. Another major goal is to change as few as possible in comparison to the original timetable. This should minimize the disturbance of the ongoing timetable because fewer changes are easier to communicate, easier to apply, and hence more reliable. Moreover, it is impossible for an operator to change many routes at the same time, because the running and blocking times highly depend on the routes and interaction between the trains. Changing a lot of routes and switching the orders of trains is very likely to result in an impractical timetable due to a significant amount of incertitude. In case of the timetable construction this is evaluated in detail by microscopic simulation which is not applicable in a real time setting. Therefore, the reference solution highly influences the objective function. The discussed re-optimization task occurs very often at a railway company. There are various causes that can lead to a situation where the implemented timetable becomes unexpectedly infeasible. Predictable and unpredictable construction sites and breakdowns that block a track must be integrated shortly into a timetable. In addition, delayed trains and modifications of speed limits may require an adjustment of the timetable. The paper contributes an adaption of the Mixed Integer Programming approaches presented in [5, 7] to re-optimize timetables. We show how to incorporate re-optimization requirements into the disjunctive graph based formulation, see [1, 6, 4, 3]. An iterative approach is used by [4] to solve real-time instances of the Dutch railway network. They use a branch-and-bound algorithm for sequencing train movements and improving the solution by locally rerouting some trains. The connection between adjacent dispatching areas is studied by [3]. [6] use a disjunctive graph formulation to model and solve a job-shop scheduling problem with blocking constraints. This paper is organized as follows. Section 2 defines the considered problem including an overview of the disjunctive based formulation. In Section 3 we present some real world scenarios, consider common re-optimization use cases for the TDP, and presents computational results. This indicates that the model and algorithmic approach produces high quality solutions in a very short time and is able to tackle the real time re-optimization setting.

2 A Re-Optimization Model for the TDP

Consider the following problem setting for the TDP. We model the infrastructure network by a directed graph $G = (V, A)$. The arcs correspond to track segments with fixed running times $\tau^r_{(v,w)}$ for each train r that is able to operate on track segment $(v, w) \in A$. For each track segment (v, w) and train pair r, r' exists a headway time $h^{r,r'}_{(v,w)}$ which is defined as the minimal time between two consecutive trains r and r' that use the same track segment (v, w), see details in [9]. The set of scheduled train requests is denoted by R. Each train $r \in R$ is associated with
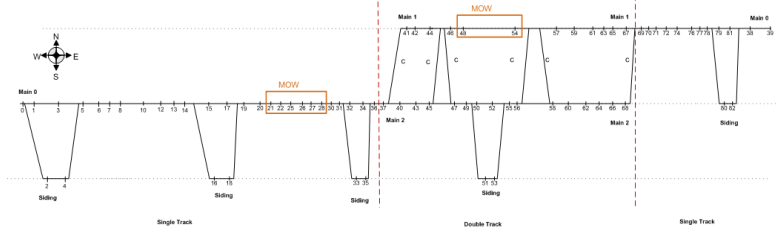
Fig. 2. Microscopic network of the RAS instances. The boxes (orange) show the disturbed areas.

an initial route $p^* \in P^r$, where P^r is the set of possible routes for request r. Additionally, there are essential stops $S^r \subset V$ for each train $r \in R$. Each stop $s \in S^r$ of train r have to be fulfilled during the time period $[\underline{\alpha}_s^r, \overline{\alpha}_s^r]$. A time unit of deviance from the scheduled departure of train r, denoted by α_s^r, is penalized by $\underline{c}_s^r$ if the actual departure time is before α_s^r, and $\overline{c}_s^r$ if the actual departure time is after α_s^r. Furthermore, we denote by δ_p^r the cost that occur from re-routing train r on route p instead of its initial route p^* with cost $\delta_{p^*}^r = 0$. By $\gamma^r \in \mathbb{R}^-$ we denote a (negative) profit value for not routing train r. Usually, this value should ensure that a maximal number of trains is scheduled. If meaningful data is available this could also be used to give the algorithm a priority estimation for each train depending on the demand. The set $B \subseteq A$ is the set of arcs (v, w) where some kind of disturbance takes place during the period of $[\underline{\beta}_{(v,w)}, \overline{\beta}_{(v,w)}]$.

A solution of the TDP has to associate each scheduled train $r \in R$ at most one route $p \in P^r$ with departure times for each node $v \in p$ under consideration of the headway constraints. The task of the model is to select a path for each train and to determine departure times t_v^r for each node v that is visited by train r on its path. For this, we enforce relations between different departure times w. r. t. the chosen paths and the order in which different trains traverse the same arc. In particular, we will make use of the following three types of decisions:

1. r uses (v, w), which is satisfied if and only if the selected path for r contains arc (v, w),
2. $r \prec_{(v,w)} r'$, which is satisfied if and only if r traverses (v, w) before r',
3. $r \prec b_{(v,w)}$ and $r \succ b_{(v,w)}$, which are satisfied if and only if r uses (v, w) before or after the disruption, respectively.

Depending on these conditions, we can formulate the following constraints for the departure times:

$$\textit{running times:} \qquad r \text{ uses } (v, w) \Rightarrow t_v^r + \tau_{(v,w)}^r \leq t_w^r, \qquad (1)$$

$$\textit{headway times:} \qquad r \prec_{(v,w)} r' \Rightarrow t_v^r + h_{(v,w)}^{r,r'} \leq t_v^{r'}, \qquad (2)$$

$$\textit{disruption times:} \qquad r \prec b_{(v,w)} \Rightarrow t_w^r \leq \underline{\beta}_{(v,w)}, \text{ and } r \succ b_{(v,w)} \Rightarrow t_v^r \geq \overline{\beta}_{(v,w)}. \qquad (3)$$

Note that using disjunctive constraints can easily be formulated using linear big-M constraints.

In particular, using the following binary variables

1. $z_p^r = 1 \iff r$ runs on $p \in P^r$,

2. $x^{r,r'}_{(v,w)} = 1 \iff r$ runs before r' on (v,w),
3. $b^r_{(v,w)} = 1 \iff r$ runs before disruption on (v,w),

a headway constraint (2) can, e. g., be written as

$$t^r_v + h^{r,r'}_{(v,w)} \le t^{r'}_v + M \cdot (1 - x^{r,r'}_{(v,w)}) \tag{4}$$

With this notation the TDP can be stated as a mixed integer program as follows:

$$\text{minimize} \quad \sum_{v \in S} (\underline{c}^r_v \underline{\Delta}^r_v + \overline{c}^r_v \overline{\Delta}^r_v) + \sum_{r \in R} \sum_{p \in P^r} \delta^r_p z^r_p + \sum_{r \in R} \gamma^r \sum_{p \in P^r} z^r_p \tag{5}$$

$$\text{subject to} \quad (1), (2), (3), \tag{6}$$

$$\sum_{p \in P^r} z^r_p \le 1, \qquad\qquad r \in R, \tag{7}$$

$$t^r_v - \overline{\Delta}^r_v \le \alpha^r_v, \qquad\qquad r \in R, v \in S^r, \tag{8}$$

$$t^r_v + \underline{\Delta}^r_v \ge \alpha^r_v, \qquad\qquad r \in R, v \in S^r, \tag{9}$$

$$t^r_v \in [\underline{\alpha}^r_v, \overline{\alpha}^r_v], \qquad\qquad r \in R, v \in S^r, \tag{10}$$

$$t^r_v, \underline{\Delta}^r_v, \overline{\Delta}^r_v \ge 0, \qquad\qquad r \in R, v \in S^r, \tag{11}$$

$$z, x, b \text{ binary} \tag{12}$$

In addition to the three types of binary variables, the continuous variables t^r_w model the departure time of train r at node w. The continuous cost variables $\underline{\Delta}^r_w$ and $\overline{\Delta}^r_w$ measure the deviation between the departure time of the reference timetable and re-allocated departures times of train r at node w. The linear objective function (5) minimizes the sum of the total costs for deviance at stops, costs for alternative routes, and costs for unscheduled trains. The constraints for the running times (1), the headway times (2) and the disruption times (3) are formulated as big-M constraints as mentioned above. The inequalities (8) and (9) ensure the correct values for the time deviation cost variables and constraints (7) ensure that at most one route is selected for each train. The departure time windows of the stops are modeled by (10).

If the trains have delays the model aims at pushing the trains back to its actual routing and timing. In some cases this is not desired since the new schedule may lead to a lot of modifications of the current timetable, which is not realizable. In this case the reference departure times could be adjusted to keep the delays at the current level. Of course, by setting the variables cost $\underline{c}^r_s, \overline{c}^r_s$ and δ^r_p to zero for all stops, paths and trains it is possible to calculate a timetable that is completely independent from the reference timetable.

3 Computational Study

We implemented the proposed re-optimization model in a C++-framework. This implementation takes use of the commercial mixed integer programming solver CPLEX 12.6. All our computations were performed on a desktop computer with an Intel(R) Xeon(R) CPU E3-1245 v3 with 3.40 GHz and 32 GB of RAM. The set of instances are scenarios derived from the INFORMS RAS Problem Solving Competition 2012, see [8].

The *RAS* instances include a microscopic infrastructure network containing 82 nodes and 184 arcs as shown in Figure 2. There are three different scenarios with increasing complexity,

Instance	$\|R\|$	disrupted Arcs	disrupted Routes	Planning Horizon (h:m:s)
RAS_1	12	2	5(41%)	17:58:47
RAS_2	18	2	6(33%)	18:07:47
RAS_3	20	12	18(90%)	16:32:15

Table 1. Key numbers of re-optimization scenarios from RAS.

i. e., in terms of larger number of trains and disturbances. Table 1 shows the corresponding sizes.

In all cases we chose $\gamma^r = -10^3$ for the profit value of routing train r. The parameter δ_p^r equals the number of deviating tracks between route p and reference route p^*. The cost parameters are set in such a way that the optimization goals are weighted in order of importance. First the number of cancelled trains should be minimal, second the number of route changes should be minimal and third the departure times should be as close as possible at the reference timetable. For the MIP solvers we set a time limit of 1800 seconds.

We limited the set of possible routes for each train since otherwise most of the trains have 192 possible routes which is far too much to handle. In addition, most of those ignored potential routes cannot be part of an optimal solution. An observation is that the model can be solved in a few seconds if the number of alternative routes per train is small. Therefore we sort accordingly to δ_p^r the alternative routes for each train and select the first 4, 8 and 16 alternative routes for each train, respectively. We use the cost parameters δ_p^r since there are the only costs that can be calculated in advance.

The computational results are in Table 2. The second and third column are the used MIP solver and the number of alternative routes for each train, respectively. It follows the number trains in the reference time table, the number of trains that are completely blocked by a disruption, the number of trains that are planned in the solution, the number of trains that are cancelled in the solution and the number of route changes. The last four columns are the running time, the final optimality gap, the optimality gap after 20 seconds and the final objective value of the solver. If the time limit was reached than this is indicated with TL in the running time column.

From the practical point of view even the restriction to four tours per train is more than a dispatcher can overlook in a couple of minutes or even seconds. We are able to solve the first two scenarios to optimality and solve the third with an optimality gap of at most 5.3%. It turns out that for the RAS instances the first four selected tours are sufficient to provide high quality solutions. Furthermore the optimality gaps after 20 seconds indicate that we are able to get good solutions fast.

4 Conclusion

We extended a well known MIP formulation for the TDP to be able to tackle re-optimization scenarios. Our computational study demonstrates that our re-optimization approach can be used to produce high quality solutions in reasonable computation time for a real time application.

Instance	Alt. Routes	Trains	Blocked	Planned	Cancelled	Route Changes	Running time (s)	Gap (%)	Gap after 20s (%)	Objective
RAS_1	4	12	0	12	0	0	3.0	0.0	0.0	-11852.72
RAS_1	8	12	0	12	0	0	17.0	0.0	0.0	-11852.72
RAS_1	16	12	0	12	0	1	51.0	0.0	21.3	-11993.00
RAS_2	4	18	0	18	0	2	12.0	0.0	0.0	-17869.33
RAS_2	8	18	0	18	0	2	89.0	0.0	>100.0	-17869.33
RAS_2	16	18	0	18	0	2	1492.0	0.0	>100.0	-17871.33
RAS_3	4	20	0	19	1	1	982.0	0.0	5.3	-18731.33
RAS_3	8	20	0	19	1	1	TL	5.3	11.4	-18731.33
RAS_3	16	20	0	19	1	4	TL	5.4	40.1	-18713.33

Table 2. Solutions of the RAS scenarios with 4, 8 and 16 alternative routes for each train.

References

1. E. Balas. Machine sequencing via disjunctive graphs. *Operations Research*, 17:941–957, 1969.
2. Ralf Borndörfer, Julika Mehrgardt, Markus Reuther, Thomas Schlechte, and Kerstin Waas. Re-optimization of rolling stock rotations. Technical Report 13-60, ZIB, Takustr.7, 14195 Berlin, 2013.
3. Francesco Corman, Andrea D'Ariano, Dario Pacciarelli, and Marco Pranzo. A bilevel rescheduling framework for optimal inter-area train coordination. In *ATMOS*, pages 15–26, 2011.
4. Andrea D'Ariano, Francesco Corman, Dario Pacciarelli, and Marco Pranzo. Reordering and local rerouting strategies to manage train traffic in real time. *Transportation Science*, 42(4):405–419, November 2008.
5. Carlo Mannino. Real-time traffic control in railway systems. In Alberto Caprara and Spyros Kontogiannis, editors, *ATMOS*, volume 20, Dagstuhl, Germany, 2011. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
6. Alessandro Mascis and Dario Pacciarelli. Job-shop scheduling with blocking and no-wait constraints. *European Journal of Operational Research*, 143(3):498–517, 2002.
7. Paola Pellegrini, Grégory Marlière, and Joaquin Rodriguez. Real Time Railway Traffic Management Modeling Track-Circuits. In *ATMOS*, volume 25 of *OpenAccess Series in Informatics (OASIcs)*, pages 23–34, Dagstuhl, Germany, 2012. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
8. INFORMS Railway Application Section (RAS). Problem-solving-competition 2012. https://www.informs.org/Community/RAS/Problem-Solving-Competition/2012-RAS-Problem-Solving-Competition, 2012.
9. Thomas Schlechte. *Railway Track Allocation: Models and Algorithms*. PhD thesis, TU Berlin, 2012.