

RALF BORNDÖRFER, MATTHIAS BREUER, BORIS GRIMM,  
MARKUS REUTHER, STANLEY SCHADE, THOMAS  
SCHLECHTE

## **Timetable Sparsification by Rolling Stock Rotation Optimization**

Zuse Institute Berlin  
Takustr. 7  
14195 Berlin  
Germany

Telephone: +49 30-84185-0  
Telefax: +49 30-84185-125

E-mail: [bibliothek@zib.de](mailto:bibliothek@zib.de)  
URL: <http://www.zib.de>

ZIB-Report (Print) ISSN 1438-0064  
ZIB-Report (Internet) ISSN 2192-7782

# Timetable Sparsification by Rolling Stock Rotation Optimization

Ralf Borndörfer<sup>1</sup>, Matthias Breuer<sup>3</sup>, Boris Grimm<sup>1</sup>, Markus Reuther<sup>2</sup>,  
Stanley Schade<sup>1</sup>, and Thomas Schlechte<sup>2</sup>

<sup>1</sup> Zuse Institute Berlin, Takustr. 7, 14195 Berlin, Germany  
lastname@zib.de, WWW: <http://www.zib.de>

<sup>2</sup> LBW Optimization GmbH, Obwaldener Zeile 19, 12205 Berlin, Germany  
lastname@lbw-berlin.de, WWW: <http://www.lbw-berlin.de>

<sup>3</sup> DB Fernverkehr AG, Im Galluspark 15-19, 60326 Frankfurt Germany

**Abstract.** Rolling stock optimization is a task that naturally arises by operating a railway system. It could be seen with different level of details. From a strategic perspective to have a rough plan which types of fleets to be bought to a more operational perspective to decide which coaches have to be maintained first. This paper presents a new approach to deal with rolling stock optimisation in case of a (long term) strike. Instead of constructing a completely new timetable for the strike period, we propose a mixed integer programming model that is able to choose appropriate trips from a given timetable to construct efficient tours of railway vehicles covering an optimized subset of trips, in terms of deadhead kilometers and importance of the trips. The decision which trip is preferred over the other is made by a simple evaluation method that is deduced from the network and trip defining data.

**Keywords:** mixed integer programming, railway rolling stock optimization, operations research

## 1 Facing Capacity Limitations

Planning rolling stock rotations in industrial railway applications is a long-term process that starts with a coarse plan and gains accuracy the closer the day of operation comes. This process is affected by all kinds of unusual events such as natural disasters (floods or snow), technical problems (track or fleet breakdowns), or man-made impediments (strikes). For example, during autumn 2014 and spring 2015, Germany's largest union of train drivers called for not less than nine strikes of varying intensities. In Germany it is possible that different unions for the same class of employees exist such that only a subset of such a class is actually on strike where the other part is still working. Consequently, a strike of a single union is a heavy decrease of capacity than a complete lock down of the railway system. Such events have widespread repercussions on the operation of a railway system: The timetable, the rolling stock rotations, the maintenance plans, and the crew schedules for the personnel in trains and maintenance facilities all have to be changed.

Finding new or revised rolling stock rotations, i.e., cyclic tours of rolling stock vehicles covering parts of the timetable, after disruptions is a well studied topic in the literature on railway optimization, see [1] for an overview.

In this paper we consider a different, more integrated approach which, to the best of our knowledge, has not been described in the literature before. The idea is to compute revised rolling stock rotations in order to “sparsify” a given undisturbed timetable. The goal is to construct rolling stock rotations that have minimum operational costs while using the limited capacities, in case of a strike the train drivers, as efficient as possible. The balance between these two objectives is controlled by an Analytic Hierarchy Process (AHP) that was developed in cooperation with our industrial partner DB Fernverkehr AG. The AHP can be seen as a key performance indicator (KPI) of the trips in the railway network, which is widely used in economy and operations research. [2] and [3] are examples for applications of KPIs in airline tail assignment. Using the train drivers as efficiently as possible directly leads to a decrease of deadhead trips and deadhead kilometers, since drivers for these kind of movements could not be used for passenger trips.

The paper is organized as follows. The next section deals with the evaluation process of the trips via the Analytic Hierarchy Process (AHP). The main contribution of this paper, the concept to sparsify the timetable according to ensure optimal rotations via mixed integer programming is part of Section 3. In Section 4 the performance of the algorithm is demonstrated via a case study for the strike period in May 2015 in Germany. Finally, we summarize the results in Section 5.

## 2 Defining Priorities by an Analytic Hierarchy Process

Before tackling the problem how to construct optimized rolling stock rotations, we deal with a subproblem of our optimization procedure. Recall that we want to choose the subset of trips to be operated from all trips of the timetable. Hence, some kind of criterion or evaluation of the trips is necessary to choose the right ones. The idea is to guide the sparsification of the timetable by a prioritization of each trip in terms of certain criteria. Afterwards optimal rolling stock rotations are constructed that cover (i.e., collect) as many trips as possible taking the trip priorities into account.

We use the *Analytic Hierarchy Process* (AHP) by [4] in order to compute trip priorities as described in [5]. The AHP involves several steps. First, criteria that describe different aspects of a trip are identified. Then weights for the importance of one criterion over every other are defined. This information is used to construct a weighting of the criteria that is used to prioritize the trips. In [5] a set of such criteria including weights for their pairwise comparison was defined as well as a sequential approach. The results were reviewed by our industrial partner DB Fernverkehr AG.

The input criteria for the AHP are defined as follows: The *passenger capacity* of the planned railway vehicle for the operation of the trip; the *line coverage* ratio

of stops of the trips and stops of the line the trip belongs to; the median of the number of lines that pass each stop of the trip called *network importance*; and the median of the number of *transfer opportunities* at each stop in an time interval after the stop.

These four criteria have the big advantage that they are completely independent from other data sources. Furthermore, it is possible to deduce them directly from existing timetable and network data. The final priority of the trip is then given by  $p_t \in \mathbb{Q}^+$  for all trips  $t \in T$ .

### 3 Trip Collecting Rolling Stock Rotation Optimization

In this section we consider the *Rolling Stock Rotation Problem* (RSRP) and extend a hypergraph-based integer programming formulation to our setting. We focus on the main modeling ideas and refer the reader to the paper [6] for technical details including the treatment of maintenance and capacity constraints.

We consider a cyclic planning horizon of one *standard week*. The set of timetabled passenger trips is denoted by  $T$ . Let  $V$  be a set of *nodes* representing timetabled departures and arrivals of vehicles operating passenger trips of  $T$ . Trips that could be operated with two or more vehicles have the appropriate number of arrival and departure nodes. Let further  $A \subseteq V \times V$  be a set of directed standard arcs, and  $H \subseteq 2^A$  a set of *hyperarcs*. Thus, a hyperarc  $h \in H$  is a set of standard arcs and includes always an equal number of tail and head nodes, i.e., arrival and departure nodes. A hyperarc  $h \in H$  *covers*  $t \in T$  if each standard arc  $a \in h$  represents an arc between the departure and arrival of  $t$ . Each of the standard arcs  $a$  represents a vehicle that is required to operate  $t$ . We define the set of all hyperarcs that cover  $t \in T$  by  $H(t) \subseteq H$ . By defining hyperarcs appropriately, vehicle composition rules and regularity aspects can be directly handled by the model. Hyperarcs that contain arrival and departure nodes of different trips are used to model deadhead trips between the operation of two (or more if couplings are involved) trips. The RSRP *hypergraph* is denoted by  $G = (V, A, H)$ . We define sets of hyperarcs coming into and going out of  $v \in V$  in the RSRP hypergraph  $G$  as  $H(v)^{\text{in}} := \{h \in H \mid \exists a \in h : a = (u, v)\}$  and  $H(v)^{\text{out}} := \{h \in H \mid \exists a \in h : a = (v, w)\}$ , respectively. Let finally  $k \in \mathbb{N}$  denote a capacity and  $\delta_t$  the respective capacity consumption of a trip  $t \in T$ , e.g., a maximum number of trips allowed to be included in the sparsified timetable, a maximum number of aggregated kilometers, or hours of length of the included trips. This number results from the estimate how many employees might be not on strike and thus could drive a train. The *Trip Collecting Rolling Stock Rotation Problem* (TCRSRP) is to find a cost minimal set of hyperarcs  $H_0 \subseteq H$  such that the capacity  $k$  is not exceeded by the trips  $t \in T$  covered by a hyperarc  $h \in H_0$  and  $\bigcup_{h \in H_0} h \subseteq A$  is a set of *rotations*, i.e., a packing of cycles (each node is covered at most once).

Using a binary decision variable for each hyperarc and a slack variable for each trip, the TCRSRP can be stated as an integer program as follows:

$$\min \sum_{h \in H} c_h x_h + \sum_{t \in T} p_t s_t, \quad (1)$$

$$\sum_{t \in T} \sum_{h \in H(t)} \delta_t x_h \leq k, \quad (2)$$

$$\sum_{h \in H(t)} x_h = 1 - s_t \quad \forall t \in H, \quad (3)$$

$$\sum_{h \in H(v)^{\text{in}}} x_h - \sum_{h \in H(v)^{\text{out}}} x_h = 0 \quad \forall v \in V, \quad (4)$$

$$x_h \in \{0, 1\} \quad \forall h \in H, \quad (5)$$

$$s_t \in \mathbb{Q}_+ \quad \forall t \in T. \quad (6)$$

The objective function of model (1) minimizes a sum consisting of the total cost of the chosen hyperarcs and the priorities of the uncovered trips. For each trip  $t \in T$  the covering constraints (3) assign one hyperarc of  $H(t)$  or a slack variable to  $t$ . Inequality (2) stipulate the capacity consumption of operated trips. (4) are flow conservation constraints for each node  $v \in V$  that induce a set of cycles of arcs of  $A$ . Finally, (5) and (6) state the domains of the decision variables.

The RSRP, and therefore also the TCRSRP, is  $\mathcal{NP}$ -hard, even if constraints (3) are trivially fulfilled, i.e.,  $|H(t)| = 1$  for all trips  $t \in T$ , see [7].

## 4 A Case Study at DBF: Strike Period 2015

The proposed model was implemented in our algorithmic framework ROTOR (see [6]) that is integrated in the IT environment of DB Fernverkehr AG. The implementation makes use of the commercial mixed integer programming solver Gurobi 6.5 as an internal LP solver to support a customized column generation and branch and bound procedure. The computations are stopped a after optimality is proved, a fixed number of branching nodes is reached or the LP-IP gap is below 1%.

Our implementation is tested on real-world instances provided by our industrial partner. There are four instances related to the 2014–2015 strike each representing a different fleet of ICE trains, i.e., *ice1*, *ice2*, *ice3*, and *iceT*. Each fleet has different sizes, vehicle characteristics, and different underlying networks which cover wide parts of Germany. To compare our solution approach we run ROTOR without the trip cancelling approach on instances that contain a limited number of trips of the normal DBF timetable. This list of trips was created by planners of DBF with a rough guess which drivers are on strike to offer a maximum customer friendly timetable as possible. Although this list is the result of the planning at DBF there were some changes made before really operating the trips during that period. Reasons for that are a larger number of employees on strike than expected or fine tuning of the rotations by adding additional passenger trips to reduce deadhead kilometres. Nevertheless, these rotations are

very close to the operated ones and therefore a most appropriate candidate to compare to. Table 1 shows the main characteristics of the solution process and its outcome. The first three columns show the instance name, respectively fleet, the number of trips, and hyperarcs that were required to model all possible train movements, couplings, and deadhead trips in the hypergraph model. Columns four and five give the sum of the trip and deadhead trip distance of all used vehicles of the solution. Since the costs are confidential column *Cost* shows only a factor of the operational cost of the computed solution. The next two columns *Gap* and *CPU* present the LP-IP gap and the run time of the optimization process. The last column gives the sum of the  $p_t$  values for all trips included in the solution.

**Table 1.**  $ice^{DB}$ : Instance with  $\approx 50\%$  manually cancelled trips by planners of DBF.

| Name        | $ T $ | $ H (\times 10^6)$ | $\sum \delta_t$ (km) | Dh (km) | Cost( $\times 10^x$ ) | Gap(%) | CPU(s) | $\sum p_t$ |
|-------------|-------|--------------------|----------------------|---------|-----------------------|--------|--------|------------|
| $ice1^{DB}$ | 379   | 0.9                | 296094               | 8777    | 1.74                  | 0.14   | 70     | 1.82       |
| $ice2^{DB}$ | 456   | 4.8                | 165906               | 13506   | 1.00                  | 0.04   | 622    | 2.42       |
| $ice3^{DB}$ | 335   | 1.6                | 186653               | 6279    | 1.42                  | 0.11   | 489    | 2.41       |
| $iceT^{DB}$ | 232   | 1.9                | 131899               | 9370    | 0.69                  | 0.47   | 441    | 1.16       |

Table 2 shows the results of the optimization runs with integrated timetable sparsification. We applied a capacity limit for each instance, respectively fleet, equal to the aggregated trip length of all trips included in the corresponding instance with manually canceled trips. Hence, optimized rotations of both approaches have an amount of comparable working hours of the train drivers. Again, columns four and five give the sum of the operated trips and deadhead trips kilometres of all used vehicle of the solution. The aggregated deadhead trip length of the optimized solutions save between  $\approx 41\%$  and  $\approx 74\%$  of the aggregated deadhead km. Also the operational costs of the optimized solutions decrease which is a consequence of the decreased number of deadhead kilometres. Comparing the last columns of the two tables shows that the approach with the included trip cancelling leads to better values for the sum of the  $p_t$  values over the trips contained in the solution. Note that the solutions found in the  $ice$ -case are most likely not in the solution space of the  $ice^{DB}$  instances, whereas solutions of the  $ice^{DB}$  instances are potential solutions for the  $ice$ -instances. The reason for that is the preselection of trips in the  $ice^{DB}$  case. In [5] it was shown that a preselection via the ordering computed with the AHP but without integration into the MIP approach is not sufficient.

**Table 2.** Instances with AHP priorities and integrated trip cancelling no vehicle cost.

| Name        | $ T $ | $ H (\times 10^6)$ | $\sum \delta_t$ (km) | Dh (km) | Cost( $\times 10^x$ ) | Gap(%) | CPU(s) | $\sum p_t$ |
|-------------|-------|--------------------|----------------------|---------|-----------------------|--------|--------|------------|
| <i>ice1</i> | 700   | 1.4                | 299154               | 2314    | 1.71                  | 0.21   | 519    | 2.15       |
| <i>ice2</i> | 973   | 5.2                | 155219               | 6470    | 0.95                  | 1.21   | 5381   | 2.46       |
| <i>ice3</i> | 922   | 3.4                | 166250               | 3676    | 1.03                  | 1.00   | 494    | 2.22       |
| <i>iceT</i> | 915   | 3.1                | 132798               | 4413    | 0.64                  | 1.00   | 2116   | 1.32       |

## 5 Conclusion

We presented the integration of a timetable sparsification method into a mixed integer programming approach to solve the TCRSRP. The timetable sparsification is guided by a fast and from external data independent evaluation of the trips. The proposed approach leads to promising results for situations with an heavily decreased offer of passenger railway trips, like strike periods.

## Acknowledgements

This work has been developed within the Research Campus MODAL [8] funded by the German Ministry of Education and Research (BMBF).

## References

1. V. Cacchiani, D. Huisman, M. Kidd, L.G. Kroon, P. Toth, L. Veelenturf, J.C. Wagenaar.: An overview of recovery models and algorithms for real-time railway rescheduling. In: Transportation Research Part B: Methodological, vol. 63, pp. 15–37, 2014.
2. E.K. Burke, P. De Causmaecker, G. De Maere, J. Mulder, M. Paelinck, G. Vanden Bergh.: A multi-objective approach for robust airline scheduling. In: Computers & Operations Research, 2010.
3. J.M. Rosenberger, E.L. Johnson, G.L. Nemhauser. A robust fleet-assignment model with hub isolation and short cycles. In: Transportation Science, vol. 38(3), 2004.
4. T.L. Saaty.: How to make a decision: the Analytic Hierarchy Process. In: European Journal of Operational Research, 1990.
5. S. Ahmadi, S.F. Gritzbach, K. Lund-Nguyen, D. McCullough-Amal.: Rolling Stock Rotation Optimization in Days of Strike: An Automated Approach for Creating an Alternative Timetable. Preprint available at: [https://opus4.kobv.de/opus4-zib/files/5642/ZR\\_15-52.pdf](https://opus4.kobv.de/opus4-zib/files/5642/ZR_15-52.pdf).
6. M. Reuther, R. Borndörfer, T. Schlechte, S. Weider. Integrated optimization of rolling stock rotations for intercity railways. In: Proceedings of the 5th International Seminar on Railway Operations Modelling and Analysis, Copenhagen, Denmark, 2013.
7. O. Heismann. The Hypergraph Assignment Problem. PhD thesis, Technische Universität Berlin, 2014.
8. Research Campus MODAL (Mathematical Optimization and Data Analysis Laboratories) <http://www.forschungscampus-modal.de/>