Zuse Institute Berlin

Takustr. 7
14195 Berlin
Germany

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

A Propagation Approach to Acyclic Rolling Stock Rotation Optimization

ZIB Report 17-24 (May 2017)

Zuse Institute Berlin
Takustr. 7
14195 Berlin
Germany

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

E-mail: bibliothek@zib.de
URL: http://www.zib.de

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

A Propagation Approach to Acyclic Rolling Stock Rotation Optimization

B. Grimm [a,1], R. Borndörfer [a], M. Reuther [a,b], S. Schade [a], and T. Schlechte [b]
[a] Department of Mathematical Optimization, Zuse Institute Berlin
Takustr. 7, 14195 Berlin, Germany
[1] E-mail: grimm@zib.de
[b] Dres. Löbel, Borndörfer & Weider GbR

Keywords
Railway rolling stock optimization, integer programming.

Abstract
The rolling stock, i.e., railway vehicles, are one of the key ingredients of a running railway system. As it is well known, the offer of a railway company to their customers, i.e., the railway timetable, changes from time to time. Typical reasons for that are different timetables associated with different seasons, maintenance periods or holidays. Therefore, the regular lifetime of a timetable is split into (more or less) irregular periods where parts of the timetable are changed. In order to operate a railway timetable most railway companies set up sequences that define the operation of timetabled trips by a single physical railway vehicle called (rolling stock) rotations. Not surprisingly, the individual parts of a timetable also affect the rotations. More precisely, each of the parts brings up an acyclic rolling stock rotation problem with start and end conditions associated with the beginning and ending of the corresponding period. In this paper, we propose a *propagation* approach to deal with large planning horizons that are composed of many timetables with shorter individual lifetimes. The approach is based on an integer linear programming formulation that propagates rolling stock rotations through the irregular parts of the timetable while taking a large variety of operational requirements into account. This approach is implemented within the rolling stock rotation optimization framework ROTORused by DB Fernverkehr AG, one of the leading railway operators in Europe. Computational results for real world scenarios are presented to evaluate the approach.

1 Introduction

The rolling stock, i.e., railway vehicles, are one of the key assets of a running railway system. In order to operate all trips of a timetable a sufficient number of railway vehicles is required. In most railway companies several different types of vehicles that may be coupled together. Therefore trips could be operated by different vehicle configurations composed of a single or a multiple number of vehicles or vehicle types. The implementation of a timetable by a rolling stock fleet must be done in a most efficient way to be in the black, e.g., deadhead trips of vehicles between two trips to change configurations are highly undesired and expensive in practice.

The offer of our industrial partner DB Fernverkehr AG (DBF) is split into two major seasons, the summer and the winter timetable. In an early planning stage, each week of the summer or winter timetable consists of the same set of offered trips, respectively. We call this week or, more precisely, the set of trips during this week a *standard week*. In order to have a master plan for the standard week *rolling stock rotations* are developed. The rolling stock rotations are the cycles along which the distinct rail vehicles go in order to cover all trips of the timetable. That is, the rotations decide on what happens to a certain rail vehicle after it has covered a timetabled trip (,e.g., it could cover another trip, undergo a maintenance or it could be parked over night). Each of these decisions is crucial for the operational efficiency and must absolutely comply with several intricate conditions: vehicle composition rules, maintenance constraints, and infrastructure capacities. This work is done around 18 months before the day of operation of a timetable and leads to ideal rolling stock rotations for each fleet, i.e., type, of rolling stock vehicles. This variety of requirements gives rise to a very challenging competition on rolling stock rotation planning. Our *productive optimization software* ROTORparticipates in this competition for DB Fernverkehr AG, one of the leading railway operators in Europe.

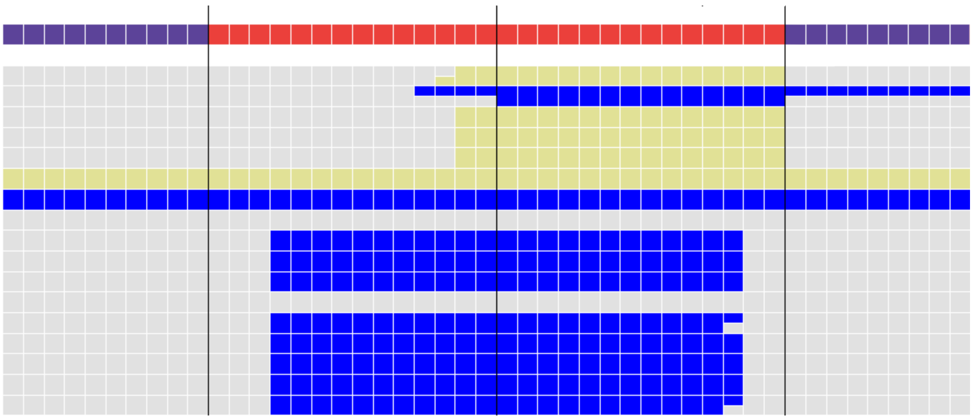

Figure 1: Operational differences of trips of a timetable period.

Besides summer and winter, there are many more situations in practice where standard week timetables are considered. Typical examples are public holidays, suddenly necessary maintenance or long term bad weather periods. For these situations the standard timetables for winter and summer are split into finer periods. Each period is associated with multiple weeks such that the trips of each part repeat from week to week, i.e., the summer and winter timetables are refined into several standard week timetables. For each of the arising periods so called *standard* rolling stock rotations (i.e., cycles) for the corresponding standard weeks are created during the planning process of DBF.

Figure 1 shows such periods. More precisely, it shows a small piece of the output of a visualization tool to analyse the structure of a railway timetable. This tool was developed for DBF to compute a minimal partition of the timetable into standard weeks, for more detail see Schade et al. (2017). The picture shows a "headline", i.e., the first row separated from the rest by the empty row and a matrix, i.e., the last 17 rows. At first focus on the rows and columns of the matrix. Each column identifies a certain day of operation, i.e., a fixed calender day. Each row of the matrix corresponds to a timetabled trip. The idea about the entries of the matrix is the following: Every color of an entry of the row stands for a different way to operate this trip. A box with two colors stands for the two possible options to operate the trip. For example the trip of the first line of the matrix containing only gray colored entries is operated exactly the same on each day of operation. Whereas the second row shows a trip that could be operated into two ways, beginning with 20 days of option one, follow by four days were both options are possible, succeeded by 14 days of option two and ending with nine days were again both options are possible. As mentioned before this is only a small subset of the complete timetable, imagine some columns on the left and the right hand side of the picture and much more rows as the timetable at DBF contains thousands of trips and not only 17. For the complete timetable at DBF one can find lots of regularity in it, i.e., a huge number of trips are operated every day. Mostly they are the same trip except for the day of operation, in other cases there exist some modifications (e.g., cancellation, additional stops, rerouteing, or different desired vehicle configurations). Let us focus on the first row or "headline". This row is somehow special. Colors of the boxes others than red indicate if all trips appear similarly in a standard week timetable. In this case the standard rotations for their standard week are assumed to be feasible, i.e., need not be changed as long as the color does not change. A red box means that an appropriate standard week does not exist. Reoccurring colors show that the same way of operation is possible again. In the particular case or time period that is visualized by Figure 1 one sees that the trips of the timetable are unchanged for the first ten days, after that there are 28 days where many changes happen leading to (at least) nine days of a new period where no trip changes its way of operation.

These pictures give a good advice that sometimes the standard rotations have to be revised to integrate the changes of the underlying timetable. Especially, smooth transitions between succeeding standard week periods and, most importantly, "rotations" for periods where no standard week exist have to be created.

This leads to planning problems with given start and end conditions for the rolling stock vehicles. For example, the timetabled trips associated with the second and third fourth of the time horizon of Figure 1 compose such problems. There are start conditions for the vehicles operating the offered timetable in these periods resulting from the planned vehicle rotations of the standard week for the first fourth weeks of the time horizon. Additionally, there are end conditions to set up the vehicles in the right spots and conditions to operate the last fourth (and maybe longer lasting) of the time horizon of 1.

Obviously, the start and end conditions make the individual planning problems interdependent. But there is another equally important requirement that links them – the regularity. It is highly desirable that the vehicles covering a trip that is repeated on several days of the time horizon always have the same vehicle composition. The vehicle composition is determined by the number of vehicles that cover a trip and their orientations. Furthermore, the order of the vehicles and their type is part of the vehicle composition. Another aspect of regularity is the regularity of turns. Once a vehicle has carried out a trip, there are often several options which trip the vehicle could carry out subsequently. Here, it is also desirable that the same option is chosen again if possible. Note that regularity should be maintained for the whole time horizon (and not only for the individual standard weeks). For example, the trips associated with the sixth and seventh row of Figure 1 should be operated similarly at all days of the time horizon.

Having a closer look at two different standard weeks of two adjacent timetable periods reveals that they still share a lot of similarities, i.e., trips that are identical except for their day of operation or trips that are only slightly changed in sense of arrival and departure times or platforms. Optimizing rotations for both weeks separately could lead to undesired properties of the rotations like different vehicle compositions in each single week for trips that are identical in both weeks.

The aim of the regularity requirement is to provide a regular operation of the timetable to the passengers. Therefore, regularity for rolling stock rotations can be seen as an analogue of periodicity in timetables, see Liebchen (2007).

Obviously, the main goal in rolling stock rotation optimization is to minimize the operational cost. Besides less decisive properties, the operational costs are mostly implied by the number of vehicles used to cover a given timetable and by deadhead trips that must be allocated to change the location of railway vehicles whenever it is necessary or efficient. Regularity seems to be contradicting when minimizing the cost of rolling stock rotations, but it is a key property to transfer rolling stock rotations into operation and can not be neglected.

Especially for the last decade there has been some research on long term railway planning problems that includes track or vehicle maintenances. There are different solution approaches like rolling horizon fashioned as in Nielsen et al. (2012) or iterative approaches like Vansteenwegen et al. (2016). In this paper, we consider a similar problem setting with a major difference, i.e., there is no way to modify the maintenance tasks or timetabled trips. Both of them are more the hidden reasons for the different periods of the timetable.

The paper deals with rolling stock rotation optimization for periods spanning multiple weeks. We present an iterative approach to compute optimized rolling stock rotations considering many technical details including regularity aspects.

The paper is organized as follows: Section 2 gives the mathematical formulation of the Acyclic Rolling Stock Rotation Problem (ARSRP). In Section 3 a propagation approach for the ARSRP with focus on long time periods is presented. This method is evaluated on real world instances for the timetables and fleets of DBF which is part of Section 4. Finally, we conclude our results in Section 5.

2 Acyclic Rolling Stock Rotation Problem

The acyclic rolling stock rotation problem can be described as follows. We consider a planning horizon, i.e., a set $\mathbb{D}$, of $|\mathbb{D}| =: n \in \mathbb{N}$ consecutive days. The set of timetabled passenger trips is denoted by T. Let V be a set of *nodes* representing departures or arrivals at respective origin and destination stations of vehicles operating passenger trips of T. There are copies of the arrival and departure nodes of the same trip if it could be operated by different fleets, vehicle compositions, and vehicle orientations. The sets $S \subset V$ and $D \subset V$ define the set of start and destination nodes at the beginning and the end of the planning horizon, respectively. For each node $s \in S$ there exists a railway vehicle that starts its rotation at s and for each node $d \in D$ a vehicle is required to end in d. Let $A \subseteq V \times V$ be a set of directed standard arcs, and $H \subseteq 2^A$ a set of *hyperarcs*. We consider an *arc-based* hypergraph definition in contrast to other variants in the literature, see Gambini et al. (1997). The reason is simply that this allows a direct definition of which vehicles are coupled together and are jointly operating trips. Thus, a hyperarc $h \in H$ is a set of standard arcs representing the movements of the vehicles that it models.

The ARSRP *hypergraph* is denoted by $G = (V, A, H)$. The 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. We define the set of all hyperarcs that cover $t \in T$ by $H(t) \subseteq H$. By defining hyperarcs many

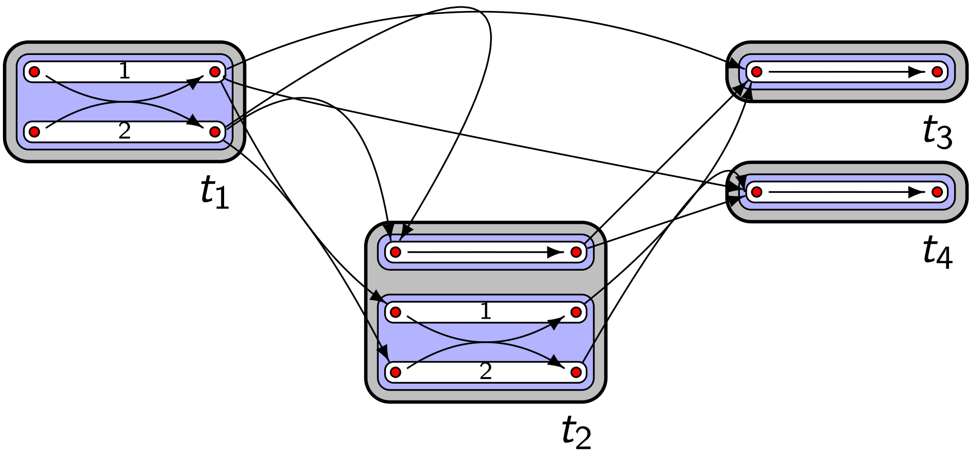

Figure 2: A sub hypergraph $G = (V, A, H)$ with $|T| = 4$ and $|V| = 14$ consisting of 5 standard arcs and 6 hyperarcs (bended arcs).

technical requirements such as vehicle configuration and regularity aspects can be handled directly by the hypergraph model, see Borndörfer et al. (2015). The ARSRP is to find a cost minimal set of hyperarcs $H^\star \subseteq H$ such that each timetabled trip $t \in T$ is covered by exactly one hyperarc $h \in H^\star$ and $\bigcup_{a \in h \in H^\star} a$ is a set of *rotations*, i.e., a set packing of s-d-paths (each node is covered at most one time). Hence, a rotation is a union of hyperpaths in G that runs through the planning horizon.

Figure 2 shows a small example to illustrate the relations between trips, nodes, and hyperarcs. All red circles are nodes of the ARSRP hypergraph. In particular the departure node at the origin station of a trip is on the left hand side within a white box and the arrival node at the destination is on the right hand side, respectively. Each white box models a vehicle operating a trip. Blue boxes show a vehicle configuration that is possible to operate a trip, i.e, blue boxes containing two white boxes model a configuration requiring two vehicles. Finally, gray boxes include the possible configurations to operate a trip. Arcs define the consecutive operation of activities. Trip t_1 is an example for a trip that has to be operated with two vehicles, for t_2 it is possible to operate it with one or two vehicles whereas t_3, t_4 require a single vehicle. Therefore, hyperarcs exist to operate t_2 after t_1 with the same configurations containing two vehicles or to decouple the vehicles to operate t_2 with a single vehicle.

The cost function $c : H \mapsto \mathbb{Q}$ is considered very detailed. It sums up the cost for the required vehicles, the energy consumption to operate the train, additional cost for deadhead movements, and artificial cost for relatively short transitions between two trips, as it is shown in 7.

$$\left.\begin{pmatrix} c_1(h) \\ c_2(h) \\ c_3(h) \\ c_4(h) \\ c_5(h) \end{pmatrix} \begin{array}{l} \ldots \text{ vehicles} \\ \ldots \text{ trip distance} \\ \ldots \text{ deadhead distance} \\ \ldots \text{ desired turn time} \\ \ldots \text{ regularity} \end{array} \right\} \text{greenfield optimization} \tag{1}$$

The individual value that c assigns to the hyperarc $h \in H$ is defined as:

$$c(h) := \sum_{i=1}^{5} c_i(h).$$

Vehicle costs are measured as a price that each minute of vehicles usage costs. Trip and deadhead distance cost are pure energy cost to move the vehicle(s) the required distance. Turn time and regularity costs are artificial cost parameter that are defined and tuned by planners at DBF. In case of the desired turn time cost a turn, i.e., the operation of one trip after an other, has a desired time gap between the arrival of the first trip and the departure of the second. An insufficient gap is penalised by certain costs. The exact numbers of the cost parameter are confidential, but the individual cost factors are ordered in decreasing order of magnitude.

Remark, that the construction of the hyperarcs allows to define hyperarcs that cover more than one trip or more than one transition between two trips by a single hyperarc. A hyperarc that covers two or more trips that are identical except of their days of operation is rewarded by the cost function. Hence, it is beneficial to choose a hyperarc that covers more than one almost identical trips than choosing a set of hyperarcs that cover the same trips. This feature models all regularity aspects of the problem. For example, imagine t_1 and t_2 of Figure 2 would be the same trip except for the day of operation, then there would be a hyperarc in the hypergraph containing both two vehicle configurations as a single hyperarc. The cost parameter to reward regular hyperarcs, i.e., that model the identical operation of trips that repeat on multiple days of operation, is user controlled and could be changed to compute solutions that are more or less regular. As a remark already a small penalty for irregular hyperarcs leads to a significantly more regular solution.

We define the sets of hyperarcs that go into and out of the node $v \in V$ as $H(v)^{\text{in}} := \{h \in H \,|\, \exists\, a \in h \,:\, a = (u, v)\}$ and $H(v)^{\text{out}} := \{h \in H \,|\, \exists\, a \in h \,:\, a = (v, w)\}$, respectively. Introducing a binary decision variable x_h, which is equal to one if h is selected in a solution, and its cost c_h for each hyperarc $h \in H$, then the ARSRP can be stated as an integer program as follows:

$$\min \sum_{h \in H} c_h x_h, \tag{IP}$$

$$\sum_{h \in H(t)} x_h = 1 \qquad \forall t \in T, \tag{2}$$

$$\sum_{h \in H(s)^{\text{out}}} x_h = 1 \qquad \forall s \in S, \tag{3}$$

$$\sum_{h \in H(d)^{\text{in}}} x_h = 1 \qquad \forall d \in D, \tag{4}$$

$$\sum_{h \in H(v)^{\text{in}}} x_h = \sum_{h \in H(v)^{\text{out}}} x_h \quad \forall\, v \in V \setminus \{S \cup D\}, \tag{5}$$

$$x_h \in \{0, 1\} \qquad \forall \in H. \tag{6}$$

The objective function of model (IP) minimizes the total cost of operating the given timetable. For each trip $t \in T$ the covering constraints (2) assign exactly one hyperarc of

$H(t)$ to t. The constraints (3) and (4) ensure that for each start node an outgoing hyperarc and for each destination node an ingoing hyperarc is chosen, respectively. The equalities (5) are flow conservation constraints for each node $v \in V \setminus \{S \cup D\}$ that imply the set of cycles in the arc set A. Finally, (6) state the integrality constraints for the decision variables.

3 Propagation Approach

The next section deals with a Propagation Approach to compute solutions to the ARSRP. The approach is motivated by the following observation. Recall that there exists an arc in G between every pair of an arrival and a departure node of V. Hence, the number of arcs $|A|$ in G grows quadratically with respect to the number of nodes V. Moreover, there are hyperarcs that cover subsets of trips that are almost identical trips, i.e., which only differ in their day of operation. Including all of these arcs respectively subsets would lead to an exponential growth of arcs with respect to the days of operation of the trip. Even adding only maximum subsets with respective hyperarcs for all identical trips at once, (i.e., a single hyperarc to operate a repeating trip with the same fleets, composition, and orientations) leads to a huge growth of hyperarcs. In all our real world instances from DBF, the case that almost identical trips are trips operated on most of all days is the rule rather than the exception. The growth of the hyperarcs is further enhanced by the fact that we are interested in scenarios that cover several weeks. This motivates the idea of our Propagation Approach which is an algorithm that decomposes the planning horizon into shorter subsets, computes solutions for the subproblems and, finally, combines these solutions to a solution of the original problem.

```
PropagationAlgorithm( G, \mathbb{D} ) // Input: Hypergraph G, planning horizon \mathbb{D}
{
  // split the horizon weekwise
  \mathbb{D} = \bigcup_{i:=0}^{k} \mathbb{D}_i := weeklyDecompose(\mathbb{D});

  // Solve the ARSRP reduced to \mathbb{D}_0
  \mathbb{S}_0 := solveARSRP( G|_{\mathbb{D}_0}, \emptyset );

  for i = 1 to k do:
  {
    // Solve the ARSRP reduced to \mathbb{D}_i with reference solution \mathbb{S}_{i-1}
    \mathbb{S}_i := solveARSRP( G|_{\mathbb{D}_i}, \mathbb{S}_{i-1} );
  }
  //Output: Combined solution for the ARSRP of \mathbb{D}
  return composeSolution( \bigcup_{i:=0}^{k} \mathbb{S}_i );
}
```

Algorithm 1: Propagation Algorithm

Algorithm 1 illustrates the propagation approach. In more detail, it starts with a weekwise decomposition of the planning horizon $\mathbb{D}$ into subsets of days $\mathbb{D}_i$ with $|\mathbb{D}_i| \leq 7$. After that, a solution of the ARSRP reduced to the planning horizon $\mathbb{D}_0$ is computed. This is done via the hypergraph $G|_{\mathbb{D}_0}$ which is constructed from G by deleting all nodes and arcs that do not correspond to an event during $\mathbb{D}_0$ and adding an artificial destination node d for each destination node $d \in D$ that was deleted. This is followed by an iteration over the remaining subsets of the decomposed planning horizon $\mathbb{D}_1, \ldots, \mathbb{D}_k$. In each iteration i the ARSRP is solved for $G|_{\mathbb{D}_i}$ with two differences:

- New start nodes s are added $G|_{\mathbb{D}_i}$ with respect to the solution $\mathbb{S}_{i-1}$ of the ARSRP of the previous iteration.
- The cost function $c_{\mathbb{D}_i} : H|_{\mathbb{D}_i} \mapsto \mathbb{Q}$ of the hyperarcs in $G|_{\mathbb{D}_i}$ is updated according to the solution of the previous iteration.

$$\begin{pmatrix} c_1(h)\\ c_2(h)\\ c_3(h)\\ c_4(h)\\ c_5(h)\\ c_6(h)\\ c_7(h)\\ c_8(h)\\ c_9(h) \end{pmatrix} \quad \begin{array}{l} \left.\begin{array}{l} \ldots\text{ vehicles}\\ \ldots\text{ trip distance}\\ \ldots\text{ deadhead distance}\\ \ldots\text{ desired turn time}\\ \ldots\text{ regularity} \end{array}\right\}\text{greenfield optimization}\\ \left.\begin{array}{l} \ldots\text{ deviating configurations}\\ \ldots\text{ deviating fleets}\\ \ldots\text{ deviating connections}\\ \ldots\text{ deviating orientations} \end{array}\right\}\text{solution dependent} \end{array} \tag{7}$$

The individual value that c assigns to the hyperarc $h \in H$ is defined as:

$$c_{\mathbb{D}_i}(h) := \sum_{i=1}^{9} c_i(h).$$

The costs of all hyperarcs $h \in H|_{\mathbb{D}_i}$ are modified by penalties that are added to the original cost if some characteristics of the solution of the previous iteration are not preserved. Hyperarcs that include trips which exist in the problem of the previous iteration, if their day of operation would be shifted by one week, get penalised, in case the hyperarc models not the same vehicle configuration, fleet, orientation, or succeeding trip as the according hyperarc of the solution of the previous iteration.

If a hyperarc $\hat{h} \in \mathbb{S}_{i-1}$ exists such that all nodes and arcs of h are equal to the ones in $\hat{h}$, but with shifted operational days for one week, then $c(\hat{h}) = c_{\mathbb{D}_i}(h)$.

By this construction it becomes beneficial to retain decisions made in the previous iterations as long as it does not become too costly. Thus, decisions made for compositions or orientations are propagated to the next iterations. Finally, a solution of the ARSRP for $\mathbb{D}$ is composed from the sub-solutions $\mathbb{S}_i, i = 0, \ldots, k$. This is possible due to the propagation of the start conditions in each iteration.

4 Computational Results

In this section, we report on a set of computational experiments carried out to assess computational effects of the Propagation Algorithm within the rolling stock optimizer ROTOR, see Borndörfer et al. (2012). We evaluate the propagation approach w.r.t. the computational tractability and the practical quality of the produced solutions. Furthermore, we compare the solutions provided by the full model with the propagation approach and with a reduced propagation approach, where the problem is decomposed, but the solutions of the foregoing subproblems are not used to solve the following subproblems.

Table 1: Results for decomposed optimization with solution propagation (PA).

instance		$\lvert H\rvert$ ($\approx \cdot 10^6$)			obj. ($\cdot 10^x$)		time (in s)		
	$\lvert T\rvert$	sum	max	FM	PA	FM	max	sum	FM
401_1	491	0.3	0.2	0.9	202	200	25	40	73
401_2	734	0.5	0.2	1.8	291	290	27	67	178
401_3	978	0.6	0.2	3.0	382	381	27	88	602
401_4	485	0.3	0.2	0.9	199	198	21	36	87
401_5	728	0.5	0.2	1.7	289	289	21	55	195
401_6	487	0.3	0.2	0.9	200	200	22	36	85
402_1	2258	9.1	4.6	16.0	486	486	740	1467	7325
402_2	3432	14.0	4.9	23.6	626	624	790	2258	29005

Our implementation makes use of the commercial mixed integer programming solver `GUROBI 6.0` as internal LP solver. All computations were performed on computers with an Intel(R) Core(TM) i7-4790 CPU with 3.60GHz, 8 MB cache, and 32 GB of RAM in multi-thread mode using four cores. The algorithm presented in Borndörfer et al. (2015) with additional constraints for start and destination nodes is used to compute the solutions to the ARSRP subproblems. In a nutshell the solution approach is a column generation procedure to generate a sufficient set of hyperarcs to solve the root LP. Followed by a sophisticated branch and bound scheme to compute integer solutions.

The set of instances that have been chosen for the experiments emerge from reasonable real-world applications that our cooperation partner DBF is facing. All instances cover a major part of the German high speed railway network for different planning horizons between two to four weeks and deal with timetable changes resulting from holidays. The 401 instances are scenarios covering Easter holidays and the 402 instances are scenarios covering Christmas holidays both in 2015. In these cases several trips have increased running times and increased number of stops, additional orientation changes during some trips, changed origin or destination stations in different weeks, or do not occur in all weeks.

For each instance we ran three experiments, the full model approach (FM), i.e., solving the ARSRP without any decomposition of the planning horizon, the propagation approach (PA) as described in Section 3, and a decomposed approach without propagation of the subproblem's optimized solutions (DA). The second columns of Tables 1 and 2 list the number of trips of the instances. The next two columns of each table show the combined and the maximum number of hyperarcs for the subproblems of PA, respectively DA, compared to the number of hyperarcs in the full model, shown in the fifths column. Note that these numbers are not equal to the number of variables in the models, since a column generation procedure is used to generate only a subset of variables. To have a more fair comparison in case of FM no arcs where constructed that connect two trips that are separated by more than seven operational days. The next block of columns headlined with *obj.* give the objective function value of PA, respectively DA, and the optimal objective computed by FM. The last three columns of both tables show the maximum, the aggregated runtime of the sub problems solved by PA, respectively DA, and for the last column the time that was required to provide the first integer solution that is better than one of the solutions of the corresponding decomposed approach. Note that this leads to slightly different running times of FM in the two tables as it sometimes takes a bit longer to beat the better solution.

Table 2: Results for decomposed optimization without propagation (DA).

instance	$\|T\|$	$\|H\|$ ($\approx \cdot 10^6$) sum	max	FM	obj. ($\cdot 10^x$) DA	FM	time (in s) max	sum	FM
401_1	491	0.3	0.2	0.9	204	200	66	81	73
401_2	734	0.5	0.2	1.8	300	290	66	110	178
401_3	978	0.6	0.2	3.0	393	381	66	145	602
401_4	485	0.3	0.2	0.9	204	198	66	95	87
401_5	728	0.5	0.2	1.7	297	289	66	130	195
401_6	487	0.3	0.2	0.9	202	200	35	85	85
402_1	2258	9.1	4.6	16.0	496	486	2258	3008	7192
402_2	3432	14.0	4.9	23.6	637	624	4513	7521	29005

Observing the values of Table 1 one can easily, and not surprisingly, see that the number of hyperarcs of the subproblems solved in PA is more or less constant for instances of both instance sets. Whereas the numbers for FM, especially for the last two rows, grow rather quickly. Focusing on the objective function values of the full model and the propagation approach reveals the gap between both objective function values is always $\leq 1\%$ of the FM's objective function value. In sense of run times of both approaches the last three columns show clearly that the PA outperforms the FM approach in the aspect of finding high quality integer solutions.

Taking a look to the important columns of Table 1, namely the values for the objective function values and the run time of the algorithms, shows that propagation of the solutions of the previous iterations has an impact on both costs of the rotations and runtime of the algorithm. All solutions computed with DA have a worse objective function value than the ones computed by PA. The gap between the objective function values of DA's solutions and the ones computed by FM rises slightly up to a worst case of 3.1%. The impact on the runtime of the algorithm seems to be much stronger. Not propagating the subproblem's solutions results in run times that are in the worst case 3.3 times longer compared to the run times of PA. The quality of the solutions for the decomposed approaches in means of the objective function values is slightly unexpected. The approaches definitely benefit from the fact that there is some kind of connection between different planning stages at DBF. Timetables are constructed in a way that they could be operated with a similar number of vehicles. Also many connections, especially between larger cities, do not change that much. The effect of the propagated solution on the runtime is not that surprising, since there is much symmetry in the solution space in all instances, e.g., solutions that only differ in orientation of vehicles or positions of single vehicles in compositions with more than one vehicle. The propagated solutions define preferences for some of these decisions which results shorter run times.

5 Conclusion

In this paper we introduce the acyclic rolling stock rotation problem and present an integer linear programming approach to solve it for large scale railway rolling stock rotation instances. We developed an Propagation Algorithm which is able to solve large real world instances in an adequate time and with high quality. Due to its iterative structure the algorithm is able to compute solutions for problems with very large time horizons. This is far beyond the scope of the full model approach which we also implemented for benchmarking reasons.

Acknowledgement

This work has been developed within the Research Campus MODAL (Mathematical Optimization and Data Analysis Laboratories) funded by the German Ministry of Education and Research (BMBF).

References

Liebchen, C., Möhring, R.H.,2007. ”The Modeling Power of the Periodic Event Scheduling Problem: Railway Timetables — and Beyond“ In: Geraets, F, Kroon, L., Schoebel, A, Wagner, D, Zaroliagis, C,D.(eds.), *Lecture Notes in Computer Science: Algorithmic Methods for Railway Optimization* vol. 4359, pp. 3–40, Springer, Berlin Heidelberg.

Borndörfer, R, Reuther, M, Schlechte, T, Waas, K, Weider, S. ”Integrated Optimization of Rolling Stock Rotations for Intercity Railways“, *Transportation Science*, vol. 0, pp, 1–15, 2015.

Nielsen, L. K.; Kroon, L., Maróti, G. ”A rolling horizon approach for disruption management of railway rolling stock“, *European Journal of Operational Research*, vol. 220, pp. 496–509, 2012.

Vansteenwegen, P., Dewilde, T., Burggraeve, S., Cattrysse, D. ”An iterative approach for reducing the impact of infrastructure maintenance on the performance of railway systems“, *European Journal of Operational Research*, vol. 252, pp. 39–53, 2016.

Gambini, R., Gallo, G., Scutellà, M.G. ”Flows on Hypergraphs”, *Mathematical Programming, Series B*,vol. 78, pp. 195–217, 1997.

Borndörfer, R., Reuther, M., Schlechte, T., Weider, S., ”Vehicle Rotation Planning for Intercity Railways”, *Proceedings of Conference on Advanced Systems for Public Transport 2012 (CASPT12)*, 2012.

Schade, S., Borndörfer, R., Breuer, M., Grimm, B., Reuther, M., Schlechte, T., Siebeneicher, P., ”Pattern Detection For Large-Scale Railway Timetables”, *Proceedings of the 7th International Conference on Railway Operations Modelling and Analysis 2017 (RailLille2017)*, 2017.