



SEPIDEH AHMADI<sup>1</sup>, SASCHA F. GRITZBACH<sup>2</sup>,  
KATHRYN LUND-NGUYEN<sup>3</sup>, DEVITA MCCULLOUGH-AMAL<sup>4</sup>

## **Rolling Stock Rotation Optimization in Days of Strike: An Automated Approach for Creating an Alternative Timetable**

<sup>1</sup> Lappeenranta University of Technology, P.O.Box 20, FI-53851 Lappeenranta, Finland

<sup>2</sup> Universität Ulm, Fakultät für Mathematik und Wirtschaftswissenschaften, Helmholtzstraße 18, 89081 Ulm, Germany - sascha.gritzbach@uni-ulm.de

<sup>3</sup> Temple University, Department of Mathematics, Rm 638 Wachman Hall, 1805 N. Broad Street, Philadelphia, PA 19122 - 6094, USA - lund-nguyen@temple.edu

<sup>4</sup> Virginia Tech, Grado Department of Industrial and Systems Engineering, 1145 Perry Street, Blacksburg, VA 24061, USA - mccullod@vt.edu

Herausgegeben vom  
Konrad-Zuse-Zentrum für Informationstechnik Berlin  
Takustraße 7  
D-14195 Berlin-Dahlem

Telefon: 030-84185-0  
Telefax: 030-84185-125

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

# Rolling Stock Rotation Optimization in Days of Strike: An Automated Approach for Creating an Alternative Timetable

---

Sepideh Ahmadi  
Sascha F. Gritzbach  
Kathryn Lund-Nguyen  
Devita McCullough-Amal

August 20, 2015

## 1 ABSTRACT

The operation of a railway network as large as Deutsche Bahn's Intercity Express (ICE) hinges on a number of factors, such as the availability of personnel and the assignment of physical vehicles to a timetable schedule, a problem known as the rolling stock rotation problem (RSRP). In this paper, we consider the problem of creating an alternative timetable in the case that there is a long-term disruption, such as a strike, and the effects that this alternative timetable has on the resulting vehicle rotation plan. We define a priority measure via the Analytic Hierarchy Process (AHP) to determine the importance of each trip in the timetable and therefore which trips to cancel or retain. We then compare our results with those of a limited timetable manually designed by Deutsche Bahn (DB). We find that while our timetable results in a more expensive rotation plan, its flexibility lends itself to a number of simple improvements. Furthermore, our priority measure has the potential to be integrated into the rolling stock rotation optimization process, in particular, the Rotation Optimizer for Railways (ROTOR) software, via the cost function. Ultimately, our method provides the foundation for an automated way of creating a new timetable quickly, and potentially in conjunction with a new rotation plan, in the case of a limited scenario.

## 2 INTRODUCTION

DB Fernverkehr AG is one of the largest providers of intercity passenger trains in the world with around 340,000 passengers every day.<sup>1</sup> In order to run their timetables, so-called rolling stock rotations are used to assign physical vehicles to scheduled trips in the timetables. Rail vehicles and their operations are one of the most expensive assets for a railway company such as DB [1]. Hence, a cost-saving vehicle assignment, which obeys technical and legal requirements, is one of the major goals in the planning process.

In order to optimize their rotation plans in a robust, mathematical fashion, DB Fernverkehr AG has partnered with the Zuse Institute Berlin (ZIB) to develop optimization software for the rolling stock rotation problem (RSRP). The Rotation Optimizer for Railways (ROTOR)<sup>2</sup> software is a result of this partnership, and it is currently used by DB on a regular basis. ROTOR uses a periodic time approach for the planning, referred to as the standard week.

However, due to public holidays, construction, maintenance, and other limitations such as strikes, there are deviations from the standard week which need to be accounted for in the RSRP. Particularly in the case of a strike, a number of trips or trains need to be cancelled, resulting in limited passenger service. In this report, we propose an approach for deciding which scheduled trips to cancel in a limited scenario while still maximizing passenger service. We employ the Analytic Hierarchy Process (AHP) (see [2], [3], [4], and [5]) to rank scheduled trips with a priority value. This process is described in more detail in Section 4.

In Section 5 we describe the data provided to us by DB as well as the methods developed for processing them. In Section 6 we analyze the timetable resulting from the AHP and compare it with DB's manually created limited timetable. Given the short-term nature of the project, we also give brief descriptions of a number of ideas for improving the AHP-based timetable and future work in Section 7, as well as a summary of what was accomplished in Section 8.

## 3 THE LIMITED SCENARIO

On July 1, 2015 DB and the labor union Gewerkschaft Deutscher Lokomotivführer (GDL) signed a collective labor agreement, which ended a ten-month long trade dispute.<sup>3</sup> During those ten months, not only did the long-distance passengers suffer, but Deutsche Bahn also faced many challenges in reconciling its reduced workforce with steady passenger demand.

Once a strike is announced, planners at DB produce an altered timetable which accounts for limitations during a strike, known as a *limited scenario*. To the best of our knowledge, this limited scenario is created manually by the planners from their experience, without the application of a large-scale mathematical model. They modify trips by cancelling them altogether

---

<sup>1</sup><http://www.deutschebahn.com/de/konzern/geschaeftsfelder/dbbahnfernverkehr/2190874/dbfernverkehr.html?start=0>

<sup>2</sup><http://www.zib.de/projects/modal-raillab>

<sup>3</sup>[http://www.deutschebahn.com/de/presse/presseinformationen/pi\\_k/9611862/h20150701.html](http://www.deutschebahn.com/de/presse/presseinformationen/pi_k/9611862/h20150701.html)

or partially, and by adding stops to allow for more connections.

## 4 DEFINING PRIORITY AND THE ANALYTIC HIERARCHY PROCESS

For our model, we assume that we are given a restriction on the number of trips that can run. For instance, suppose a certain number of workers go on strike, and DB tells us that there are enough workers to run approximately one-third of the trips each day of the week. Our goal is to rank all the trips for a given day by some sort of priority measure and then run only the top third of them. The challenge, then, is defining this priority measure.

Before moving forward with the definition of the priority measure, it is important to note that we are now ignoring the notion of the train and focusing only on the trip. As the train is a collection of trips, each valid on different days of the week, we cannot compare one train to another. Consequently, it can happen that a trip runs on one day of the week but is cancelled on another. And if all the trips for a given train are cancelled, then the train too is cancelled.

The basis for our priority measure is the Analytic Hierarchy Process (AHP) (see [2], [3], [4], and [5]), developed by Thomas Saaty. The AHP consists of two comparison stages. First, we identify criteria that describe different aspects of a trip and then weigh the importance of these criteria with respect to each other in a pairwise fashion. Secondly, for a given criterion, we weigh each trip with respect to each other, again in a pairwise fashion. These two steps are visualized in Figure 4.1. The criteria used in the process are defined as follows, each for a given trip on a given day of the week:

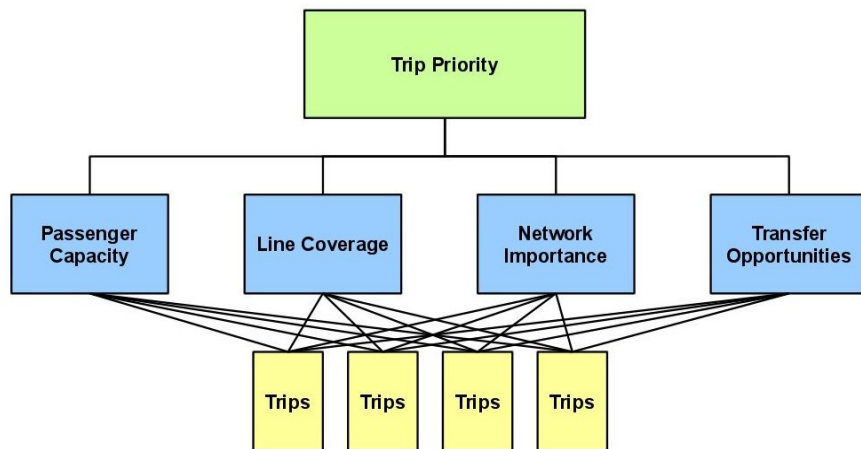


Figure 4.1: Map of Hierarchy Structure

- Passenger Capacity (PC): the maximum number of passengers a given trip can carry
- Line Coverage (LC): a ratio of the number of stations covered by a trip versus the maximal number of stations covered by the corresponding line

- Network Importance (NI): median of the numbers of lines per station among all the stations covered by a given trip
- Transfer Opportunity (TO): median of the numbers of trips per station among all the stations covered by a given trip

In the first stage of the procedure, we generate a pairwise comparison matrix as in Table 4.1, in which each entry corresponds to a comparison value between a pair of criteria. In particular, the value indicates how much more important we determine a criterion is over the other. We use a standard scale running from 1 to 9, with 1 representing "equally important" and 9 "extremely more important" [5]. For example, in the third row of the first column of Table 4.1, we have decided that network importance is much more important than passenger capacity, assigning a value of 7 from the standard scale. To remain consistent, we then must assign a value of  $\frac{1}{7}$  to the first row of the third column.

More generally, the pairwise comparison matrix is structured as follows: given any two criteria with corresponding indices  $i, j = 1, \dots, 4$ , where, without loss of generality, criterion  $i$  is more important than criterion  $j$  by a value of  $x$ , we assign the value  $x$  to the  $(i, j)$ -entry and  $\frac{1}{x}$  into the  $(j, i)$ -entry. Naturally, the diagonal entries take on the value of 1. The resulting matrix is strictly positive in each entry, ensuring the existence of a unique normalized eigenvector with positive entries, as known as the Perron vector [5]. In the AHP, this vector is used for weighting the criteria with respect to each other, so we therefore call it the *priority vector*. This vector determines in a straightforward fashion the importance of each criterion with respect to the others. Given that the pairwise comparison matrix is so small, computing the Perron vector is computationally inexpensive.

Table 4.1: Pairwise comparison matrix and priority.

	PC	LC	NI	TO	Priority vector
PC	1	1	$\frac{1}{7}$	$\frac{1}{6}$	0.0705
LC	1	1	$\frac{1}{4}$	$\frac{1}{3}$	0.0961
NI	7	4	1	3	0.5494
TO	6	3	$\frac{1}{3}$	1	0.2840

Consistency Ratio: 0.0495, Consistency Index: 0.056

Due to the reciprocal nature of the pairwise comparison matrix, very few decisions have to be made in order to fill up the matrix. In our case of four criteria, only six decisions need be made, and our reasoning for these decisions is as follows:

Generally, we want to ensure some coverage of the main lines while trips on less frequented and less important lines are cancelled. Also, even though we want trips that are retained in the limited scenario to carry a high number of passengers (PC); and even though we also want that those trips cover most of their respective line to ensure some overall network coverage (LC); we reason that it is more important to keep trips with highly frequented stations operational (NI, TO). Furthermore, we value NI over TO, since we want to emphasize line connections.

Before continuing with the second step of the AHP, we would like to highlight some of the properties of the process thus far. Ideally, the comparison decisions should obey some kind of transitivity property, e.g., if property B is slightly more important than property A and if property C is moderately more important than B, then C should at least be moderately more important than A. For this, Saaty [5] proposed a consistency ratio (CR) defined as

$$\text{CR} := \frac{\lambda_{\max} - n}{n - 1}, \quad (4.1)$$

where  $\lambda_{\max}$  is the principal (Perron) eigenvalue of the pairwise comparison matrix and  $n$  is the number of criteria. The resulting CR is then divided by the mean CR of a random positive consistent matrix of size  $n \times n$ , which is 0.89 for  $n = 4$  [5]. This ratio gives the so-called consistency index CI, here  $\text{CI} = 0.056$ . Saaty suggests that for continuing the process, a consistency index of less than 0.09 is suitable for  $n = 4$ . Otherwise, some adjustments to the decisions have to be made which are outlined in [5].

With the first stage complete and a priority vector in hand, we now move on to the next stage of the AHP, in which trips are compared in a pairwise manner for a given criterion. Each criterion comes with its own natural scale, i.e., the value of the criterion itself for a given trip. The corresponding comparison matrix for a given criterion, then, is simply a weight matrix generated by the criterion values. More specifically, for any two trips  $i, j = 1, \dots, n$ , where  $n$  is the number of trips, with criterion values  $w_i$  and  $w_j$ , the  $(i, j)$ -entry  $a_{ij}$  of the comparison matrix is given by  $a_{ij} = \frac{w_i}{w_j}$ .

Indeed, such a weight matrix is fully consistent. That is,  $a_{ij} = \frac{a_{ki}}{a_{kj}}$  for  $i, j, k = 1, \dots, n$ . The theory gives us that this matrix has rank 1, and that its only nonzero and hence principal eigenvalue is  $n$  (see [4]). The corresponding positive eigenvector is  $[w_i]_{i=1}^n$ . By the Perron-Frobenius Theorem (as in, for example [6]), this vector is a scalar multiple of the Perron vector for the weight matrix, which, as in the first stage, is used for ranking the trips with respect to the given criterion.

From a computational point of view, there is therefore no need to set up the criterion-specific comparison matrices nor calculate their Perron vectors, since the theory tells us already what they are. While such matrices would not have been so large in the first place ( $O(n^2)$ ), this still saves us computational time and effort and is a fortunate consequence of our particular problem.

Finally, we combine the priority vectors from the first and second stages to globally rank the trips. Let  $[c_j]_{j=1}^4$  denote the priority vector from stage one, and  $[w_i^{(j)}]_{i=1}^n$  with  $j = 1, \dots, 4$ , the collection of criterion-specific priority vectors from the second stage. Our cumulative priority vector  $[p_i]_{i=1}^n$  is then given as a linear combination and weighted average of the criterion-

specific priority vectors, as follows:

$$p_i := \sum_{j=1}^4 c_j w_i^{(j)}, \quad (4.2)$$

We then sort this final priority vector to determine which trips to keep or eliminate.

## 5 SOLUTION PROCESS: CLEANING THE DATABASE AND RUNNING THE AHP

Before applying our priority-based model, we must process and transform the database provided to us by DB, which consists of two timetables. The first is the complete timetable from a standard week in XML format, consisting of all scheduled trips for the ICE fleets 401, 402, 403, 406, 407, 411, 415 and 605. Note that these fleets also include vehicles that can be operated in the railway networks of Schweizerische Bundesbahnen (SBB), Nederlandse Spoorwegen (NS), Société Nationale des Chemins de fer Français (SNCF), and Danske Statsbaner (DSB). The second is a spreadsheet of a limited timetable created manually by DB in May 2015. The spreadsheet is an abbreviated timetable, and does not include such information as intermediate stops for a given trip.

In the following, we detail the process for editing, cleaning, and transforming our data so that the AHP routine, implemented in Matlab, could be applied and so that comparisons among the original timetable, the AHP-limited timetable, and the DB-limited timetable could be carried out. See Figure 5.1 for a comprehensive structure of our solution process.

### 5.1 THE CLEAN TIMETABLE AND REFERENCE ROTATION PLANS

In the first step, we remove from the original (raw) timetable all trips which only pass through stations abroad and all deadhead trips to obtain a Germany-centric (or “clean”) timetable of passenger trips. There are 902 trains in the raw timetable, and only 549 after the cleaning, which amounts to on average 400 trips per day. This cleaning is executed in an automated fashion via routines written in Matlab<sup>4</sup>.

After the clean XML file is generated, we divide it into five files corresponding to different sets of fleets, and then run each file in ROTOR to create reference solutions and rotation plan data. These reference solutions are necessary for running the limited timetables through ROTOR, in order to ensure that we have as little deviation from the original rotation plan as possible.

### 5.2 THE DB-LIMITED TIMETABLE AND ROTATION PLANS

We then create a DB-limited timetable file, based off the clean timetable and the DB-limited scenario spreadsheet. This step is currently the most tedious and most prone to error, since it

---

<sup>4</sup>See the documentation *Matlab Routines for Processing Timetable Data, Implementing the AHP, and Visualizing Results* of our code package.



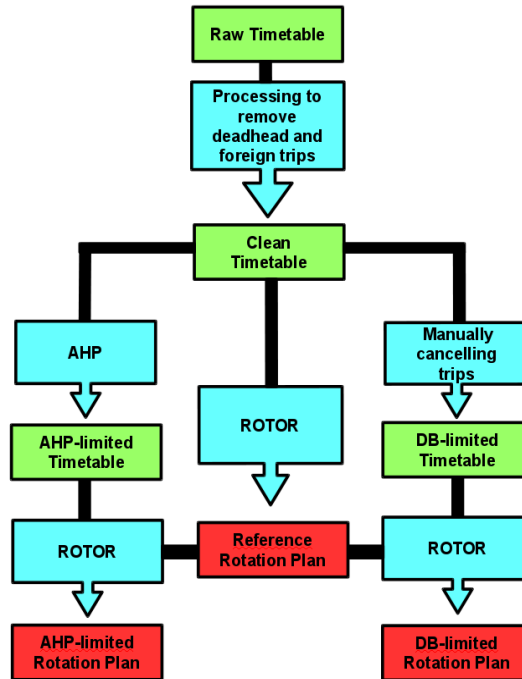


Figure 5.1: A flow chart of our solution process, including the data-processing stages and the generation of rotation plans.

had to be conducted by hand. However, the process led to a new feature in ROTOR, in which one can cancel a trip (German: *Fahrplan*) by setting the boolean "Ausfall" property to "true", and an entire train (German: *Zug*) by setting the boolean "ZugAusfall" property to "true". Then ROTOR will simply ignore the train or trip.

Due to the vast amount of data, the manual creation of the DB-limited timetable was carried out by several persons. Furthermore, there are a number of inconsistencies between the timetable and the spreadsheet. In order to standardize the process, the following assumptions and rules were established:

- In the case of differences in departure or arrival times, the information from the clean timetable should be retained.
- If the DB-limited spreadsheet included additional stops, they are not included in the DB-limited timetable.
- Partial cancellations from the DB-limited spreadsheet are included in the DB-limited timetable. If, after accounting for the partial cancellation, fleet configurations differ at a stop other than the first in the city of origin or the last in the city of destination, then these differences should be ignored.

- The days of validity from the DB-limited spreadsheet are used.
- Trips not located in the DB-limited spreadsheet are not included.

After obtaining the DB-limited timetable in XML format, we again divide the file into five smaller files, each corresponding to a set of fleets, and then run each file in ROTOR along with the reference solution to obtain a DB-limited rotation plan and data.

### 5.3 THE AHP-LIMITED TIMETABLE AND ROTATION PLANS

After creating the DB-limited timetable, some additional Matlab routines are run in order to determine how much of the clean timetable the DB-limited timetable retains. In fact, the DB-limited timetable retains 242 trains, or on average 200 trips per day. We therefore run the AHP on each day of the week separately, keeping on average half of the trips per day, in order to end up with a timetable that can compare fairly with the DB-limited timetable.

As with the DB-limited timetable, we then divide up the AHP-limited timetable into five files, each file corresponding to a set of fleets, and then run these files in ROTOR with the corresponding reference solution in order to obtain AHP-limited rotation plans and data.

### 5.4 COORDINATES DATA

A final small, yet important, component of our database is the geographical coordinates for each station. We were given a coordinates file containing latitudes and longitudes for each station; however, at least thirty stations were missing, and there were some issues with the formatting of station names that included spaces<sup>5</sup> These issues were rectified by hand, using Google maps and personal knowledge of Germany and bordering countries.

## 6 RESULTS AND ANALYSIS OF THE AHP-LIMITED SCENARIO

There are two main outputs for our problem: the timetable and the rotation plan. We will first analyze aspects of our resulting timetable and compare it with the DB-limited timetable, and then we will look at cost outputs from the rotation plans.

### 6.1 TIMETABLE DATA

To assess how well our limited timetable compares with that of DB, we examine how the criteria used to defined priority differ across all three scenarios.

First, we take a look at the average line coverage in Table 6.1. In general, it varies little across each scenario for a given day of the week, suggesting that cancelling trips does little to alter line coverage. Consequently, one could argue that line coverage is not an important criterion

---

<sup>5</sup>In fact, we would like to recommend to DB that in the future, they use underscores rather than spaces for their station names.

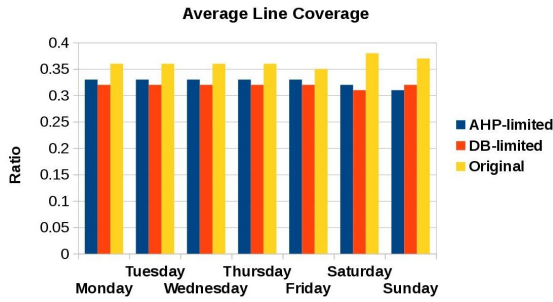


Table 6.1: Average line coverage across each scenario.

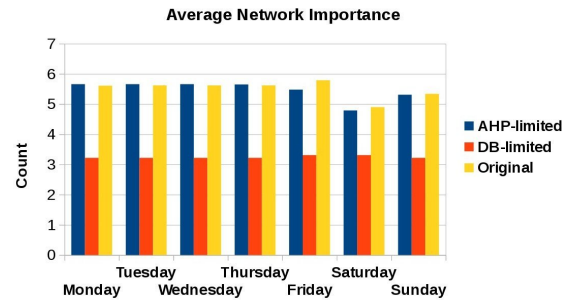


Table 6.2: Average network importance across each scenario.

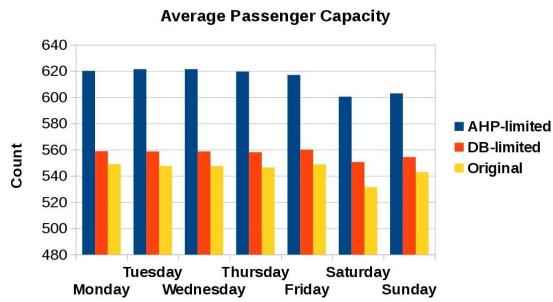


Table 6.3: Average passenger capacity across each scenario.

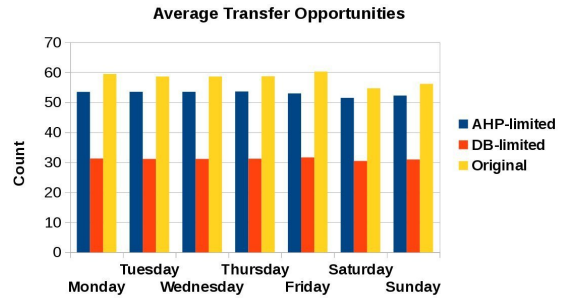


Table 6.4: Average transfer opportunity across each scenario.

in determining which trips to cancel; or rather, if we subjectively view it as important, then we should increase its importance in the pairwise comparison matrix in Table 4.1. If line coverage is to be retained as a useful criteria, then one should also consider adding stops to trips so that a single trip covers more of the line. This is discussed further in Section 7.

With regards to the average network importance in Table 6.2, the AHP-limited timetable retains the same values as in the original timetable, whereas the DB-limited timetable drops significantly. However, as one can see in Figure 6.1, it is not necessarily a good thing that the AHP-limited scenario retains the average network importance. Effectively, what has happened is that the AHP has eliminated trips with stations that are only serviced by one line, thus reducing the connectedness of the network. Despite having lower network importance, the DB-limited timetable retains a more connected network. This situation indicates that perhaps one should change the relation on network importance. In our model, the relation is positive linear: as network importance goes up, so does the priority of the trip. An inverse or perhaps log relation may be better at returning a more desirable timetable.

The average transfer opportunity in Table 6.4 points to similar issues as does the average network importance. Again, the AHP-limited timetable has relatively the same values as the

original scenario, while the DM-limited timetable drops significantly. However, if we look at the related map in Figure 6.2, the DB-limited scenario again has a more connected network, retaining at least one trip for stations the AHP determined to be unimportant.

Finally, passenger capacity as seen in Table 6.3 increases drastically for the AHP-limited scenario—in fact, it is even higher than that of the original. This is likely because the AHP cancelled trips with low passenger capacity, so the average of the remaining trips is quite high. The DB-limited timetable also ends up with higher passenger capacity than the original timetable, likely for the same reason.

As alluded to already, two maps were generated to compare how the AHP and DB methods altered the original connectedness of the network. The first, Figure 6.1, plots circles whose areas vary with the number of unique lines running through each station on a Monday. The second, Figure 6.2, plots circles whose areas vary with the number of trips running through each station on a Monday. (These plots could of course be generated for any day of the week, but given how similar the results are for each day, we arbitrarily focus on Monday.)

One quickly notes that while the maps represent different quantities, they are qualitatively very similar. In both maps of the AHP-limited timetable, coverage is very high around major cities and essentially non-existent elsewhere. The DB-limited timetable, however, retains at least one line or trip through most stations.

In particular, both the AHP and the DB methods cancel trips to Denmark (see also Subsection 6.2, where we find that all trains in fleet 605 have been cancelled). However, the AHP generally cancels more foreign trips than DB. For example, all trips to Austria (east of Munich) and to Switzerland (in the south/southwest) have been cancelled in the AHP timetable, while DB keeps at least one trip running through those regions.

Overall, it seems that criteria we have used to implement the AHP gives strong preference to trips with large passenger capacity passing through heavily trafficked stations and lines. This is precisely what we had set out to do with the AHP. However, as noted in the maps, our method sacrifices network connectedness by eliminating trips running through "unimportant" stations. As will be discussed in Subsection 6.2, these unimportant stations may be vital for reducing costly deadhead trips, and the elimination of them will therefore be a disadvantage of our AHP method.

## 6.2 ROTATION PLAN COSTS

We now turn to data from the rotation plans generated by ROTOR. Note that all trips pertaining to fleet 605 have been cancelled in the AHP-limited timetable as well as the DB-limited timetable, so we do not have rotation plan data for this fleet.

As mentioned in Subsection 5.3, roughly half of the trips were cancelled in the DB-limited and hence also in the AHP-limited timetable. These cancellations are reflected in the trip distance covered, which is also approximately half for both limited scenarios, with the DB-

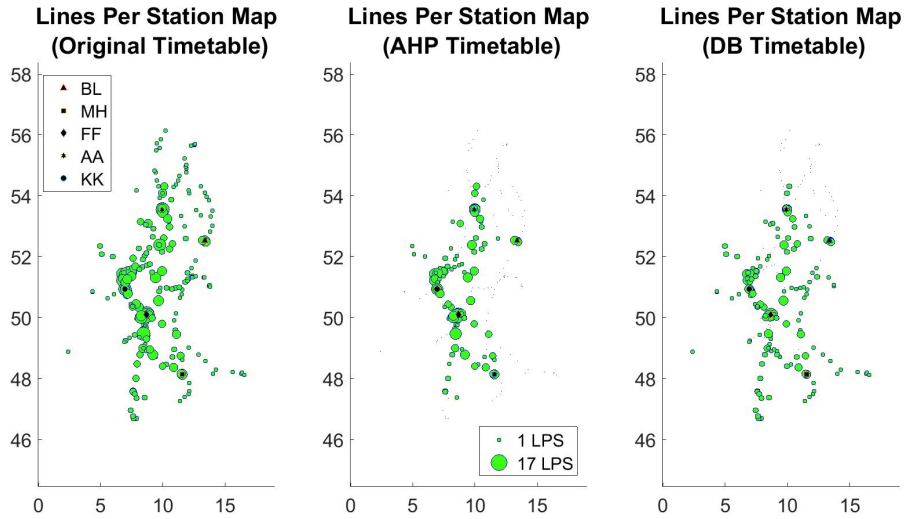


Figure 6.1: A map-based comparison of the number of lines running through each station for each scenario.

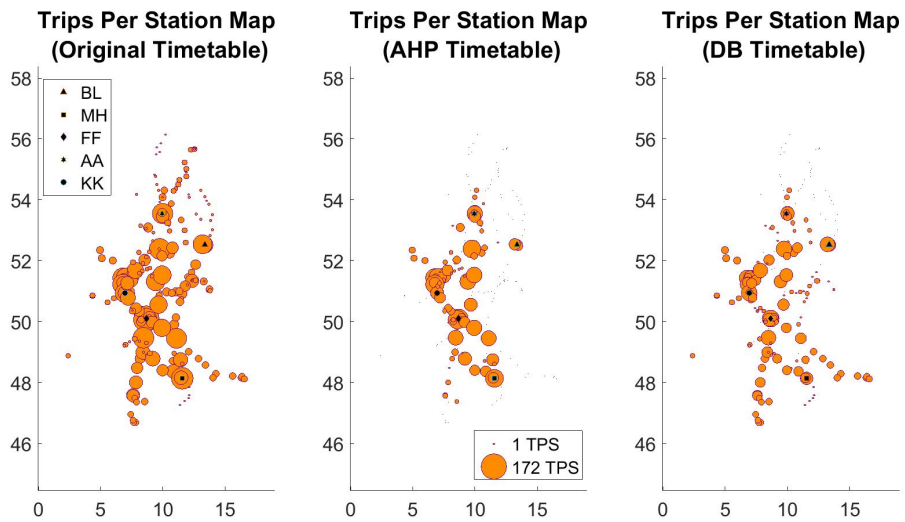


Figure 6.2: A map-based comparison of the number of trips running through each station for each scenario.

limited scenario having slightly more than the AHP-limited scenario. This slight discrepancy likely occurs because the AHP-limited timetable does not branch out as much as the DB-limited timetable (see again Figure 6.1 or Figure 6.2). Considering that the DB-limited scenario has a lower total cost, this puts the AHP-limited scenario at a slight disadvantage.

Table 6.5: Rotation plan summary across all fleets

	Original	DB-limited	AHP-limited
Operational costs [Euro]	$19 \times 10^6$	$10 \times 10^6$	$12 \times 10^6$
# deviating trips	377	200	258
# vehicles	216	114	126
trip distance [km]	1,622,473.93	779,345.26	748,784.81
deadhead trip distance [km]	27695.69	51571.62	138461.99
# deadhead trips	1006	693	694
# turn around trips	62	226	43
turn time violation [min]	2138.67	507.79	1086.56

Interestingly, the deadhead trip distance in both limited scenarios is rather high, with the AHP-limited scenario faring the worst. However, both limited scenarios have roughly the same number of deadhead trips, so the AHP-limited scenario makes longer deadhead trips than the DB-limited scenario. Either way, both limited scenarios increase the total deadhead distance over the original scenario. This is particularly unfavorable, considering that the reason for creating a new timetable in the first place is to reduce the railway traffic in proportion to the reduced resources causing the limitation. A possible solution for the increased deadhead trip distance would be to integrate the trip-cancelling regime with the rolling stock rotation optimization regime, which we discuss further in Section 7.

As for overall costs, the DB-limited scenario is about half as expensive as the original scenario, with the AHP-limited scenario about 20% more expensive than the DB-limited scenario. The main reasons for the higher costs in the AHP-limited scenario are clearly the higher deadhead distance, higher number of vehicles, higher amount of deviating trips, and the higher number of minutes below target turn time. The latter quantity is especially disturbing. Considering the inherent stress of a limited-resource situation, imposing lower target turn times on trips could strain resources which are already stretched thin. We are not immediately aware of ways to reduce the number of minutes below target turn time, but we believe longer turn times would result from a better connected network with fewer deadhead trips.

Finally, we note that the AHP-limited scenario has more deviating trips than the DB-limited scenario. This is not so much a resource issue as an assignment one. DB has defined a notion of regularity that they want imposed on their rotation plans. One of the aspects of regularity is that a vehicle retain its orientation. Loosely speaking, orientation is a binary assignment of either "tick" or "tack," i.e., that a vehicle point in one direction of the line or the other. If the rotation plan demands that a vehicle change its orientation from one day to the next, then regularity is violated; a deviating trip is recorded; and the associated cost is added to the tally. Even though our plan includes more deviating trips than DB's, these deviations could be resolved again, perhaps, by adding a few trips outside the main routes so that changes in orientations are avoided.

## 7 FUTURE WORK

Given the short-term nature of the project, we had more ideas than time to implement them. Here we discuss some of those ideas, as well as suggestions for improving what we were able to implement.

**IMPROVED DATA PROCESSING ROUTINES.** Processing the data was the most involved aspect of our project, and many issues arose while trying to ensure we had a consistent database. Furthermore, as mentioned in Section 5, some of the data processing was conducted by hand— in particular, creating XML files that matched the DB-limited scenario spreadsheet. In the future, it would be better to have an automated routine that creates a properly formatted XML timetable out of such a spreadsheet. Indeed, such a routine could be generalized to create timetables in general from spreadsheets, which would be helpful when one needs small instances for testing in ROTOR or future optimization software.

**ADDITIONAL CRITERIA TO CONSIDER IN THE AHP.** Thus far, we only considered four criteria—passenger capacity, line coverage, network importance, and transfer opportunity. In an earlier version of our model, we had also considered trip duration, giving priority to shorter trips. However, we later realized that this conflicted with the fact that we wanted to cover as much of each line as possible, which would inherently require longer trips. Other criteria one may wish to consider are the physical distance covered by each trip, how often similar trips occur the day, or whether a trip passes through a station that no other trip passes through. One should also consider revising our definitions for such things as line coverage or network importance to include the direction in which a line is traversed. Line direction is important in the case that a line is covered by a trip in one direction but possibly not in the other.

Furthermore, there are subjective notions for determining the importance of a trip which we did not consider, but would likely be important in practice. For example, one can see in Figure 6.1 and Figure 6.2 that our method did not regard trips running through stations in the southeast of Germany as important, but DB preserved these. A subjective criteria that allows a DB expert to "force" certain trips to have high priority may rectify such discrepancies. Alternatively, as noted in Section 6, coverage for these stations may be lacking because data for foreign trips was excluded. One should then attempt to incorporate data from these foreign trips into the decision-making process without including them in the timetable itself.

Finally, one might also consider different relations on the criteria values, e.g., inverse or logarithmic relations, instead of linear ones. As noted in Section 6, it appears that having, for instance, high network importance or transfer opportunity as in the AHP-limited scenario did not lead to widespread coverage of the map. Therefore a different relation may help correct this issue.

**OTHER DECISION-MAKING ALGORITHMS AND POSSIBLE PITFALLS WITH THE AHP.** While the AHP is elegant and easy to implement, one should also consider similar decision-making

procedures and how they compare to the performance of the AHP, as is done in [7]. As noted by [8], the AHP has a number of potential short-comings which we did not consider. Before going further with the AHP, one should ensure that the method is suitable for the limited timetable problem and, in particular, that it is numerically stable.

**EVALUATION AND VISUALIZATION TOOLS.** Evaluating and visualizing results from our problem is inherently difficult, since timetables and rotation plans for real-life data are large and complex. We found a map visualization to be particularly interesting, but there is much room for improvement. In fact, another reason to consider line direction, or at least an ordered set of the all the stations composing a line, is for creating maps of results related to the lines of the network. We had hoped to generate a map similar to the one DB uses for its ICE lines <sup>6</sup>. However, without an ordered set of stations for each line or a functioning routine to extract such information, such visualizations were impossible.

Aside from visualizations, there are other interesting tools one could use to evaluate our results. For example, for each pair of important stations, one could calculate how long it would take a passenger to travel between them. Such a measure is especially useful for predicting passenger dissatisfaction during limited scenarios. It could prompt DB to either change the timetable or to provide extra resources to calm passengers who would end up waiting for extended periods. One could also check whether there are any pairs of stations that are no longer connected after trips have been removed. Results from this measure could prompt local railway networks to provide extra service for routes that the ICE would no longer cover. Finally, such tools would be useful in comparing limited scenarios resulting from different methods, to determine which one would best service passengers.

**INTEGRATION OF THE TRIP-CANCELLING PROCESS INTO ROTOR.** ROTOR contains a number of useful features that we did not have time to explore, in particular, slack costs. A *slack* is a trip that cannot be feasibly covered by any heuristic in ROTOR. There are high penalty costs associated with such slacks, as it is obviously undesirable to leave trips uncovered in the timetable. In addition to the Analytic Hierarchy Process, we had also considered manipulating slack costs in ROTOR to force it to cancel trips along with returning an optimized rotation plan. We think that once appropriate modifications are made to the priority measure, that it would be well suited to determine these slack costs, which is an important first step in an integrated routine that cancels trips and optimizes the rotation plan all at once.

**ADJUSTMENTS OF MODEL ASSUMPTIONS.** Treating trips as monolithic items to either include or exclude is an inherent limitation in our model. That is, the only changes permitted to the timetable is the entire cancellation (or scheduling) of a trip. However, in DB's limited scenario, some trips were cancelled partially, or stops were added or deleted. These additional options are likely what give their scenario an advantage over ours. A natural next step for our model, then, would be to account for sections of trips and addition or deletion of stations.

---

<sup>6</sup>[http://www.bahn.de/p/view/mdb/bahnintern/fahrplan\\_und\\_buchung/streckenplaene/mdb\\_176845\\_ice\\_linienetz\\_2015.pdf](http://www.bahn.de/p/view/mdb/bahnintern/fahrplan_und_buchung/streckenplaene/mdb_176845_ice_linienetz_2015.pdf)  
Accessed 14.08.2015



Furthermore, we question whether limiting the number of trips is the correct approach to the limited scenario. In the case of a strike, for instance, the number of workers is limited first, which indirectly limits how many trips can run. However, the issue then is not which trips should run, but rather how to divide up the worker resources to run as many trips as possible. This is, of course, an assignment problem in and of itself and leads to a more complicated scenario, but we believe it to be the more correct perspective on the problem.

## 8 CONCLUSION

We have developed an approach for creating an alternative timetable for DB's ICE-network using the Analytic Hierarchy Process (AHP). To achieve this end, we developed a package of code that processes the large timetable files and runs the AHP in a matter of a few minutes. Overall, our implementation of the AHP is automated and fast, which is a major advantage over the manual process regularly implemented by DB, since typically a limited scenario must be designed on short notice. We successfully created a preliminary timetable which retains trips between the most important transit hubs of the ICE-network. The flexibility of our method also lends itself to number of simple modifications that could lead to future improvements. Ultimately, we believe our work is a first step towards a fair and quick method for providing passengers and DB with alternative timetables during times of unexpected interruptions to service.

## REFERENCES

- [1] M. Reuther, R. Borndorfer, T. Schlechte, and S. Weider, "Integrated optimization of rolling stock rotations for intercity railways," ZIB, Takustr.7, 14195 Berlin, Tech. Rep. 12-39, 2012.
- [2] T. L. Saaty, "A scaling method for priorities in hierarchical structures," *Journal of Mathematical Psychology*, vol. 15, pp. 234–281, 1977.
- [3] —, "Rank according to Perron: a new insight," *Mathematics Magazine*, vol. 60, no. 4, pp. 211–213, 1987.
- [4] —, "How to make a decision: the Analytic Hierarchy Process," *European Journal of Operational Research*, vol. 48, pp. 9–26, 1990.
- [5] —, "On the measurement of intangibles. a principal eigenvector approach to relative measurement derived from paired comparisons," *Notices of the AMS*, vol. 60, no. 2, pp. 192–208, February 2013.
- [6] F. R. Gantmacher, *The theory of matrices*. New York: Chelsea Publ., 1960.
- [7] M. Velasquez and P. T. Hester, "An analysis of multi-criteria decision-making methods," *International Journal of Operations Research*, vol. 10, no. 2, pp. 56–66, 2013.

- [8] J. S. Dyer, "Remarks on the analytic hierarchy process," *Management Science*, vol. 39, no. 3, pp. 249–258, 1990.