Zuse Institute Berlin

Takustr. 7
14195 Berlin
Germany

Stanley Schade, Ralf Borndörfer, Matthias Breuer, Boris Grimm, Markus Reuther, Thomas Schlechte, Patrick Siebeneicher

Pattern Detection For Large-Scale Railway Timetables

ZIB Report 17-17 (April 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

Pattern Detection For Large-Scale Railway Timetables*

Stanley Schade [a,1], Ralf Borndörfer [a],
Matthias Breuer [b], Boris Grimm [a], Markus Reuther [a,c],
Thomas Schlechte [c], Patrick Siebeneicher [b]
[a] Zuse Institute, Takustr. 7, 14195 Berlin
[1] E-mail: schade@zib.de
[b] DB Fernverkehr AG
[c] Dres. Löbel, Borndörfer & Weider GbR

Abstract
We consider railway timetables of our industrial partner DB Fernverkehr AG that operates the ICE high speed trains in the long-distance passenger railway network of Germany. Such a timetable covers a whole year with 364 days and, typically, includes more than 45,000 trips. A rolling stock rotation plan is not created for the whole timetable at once. Instead the timetable is divided into regular invariant sections and irregular deviations (e.g. for public holidays). A separate rotation plan with a weekly period can then be provided for each of the different sections of the timetable. We present an algorithmic approach to automatically recognize these sections. Together with the supplementing visualisation of the timetable this method has shown to be very relevant for our industrial partner.

Keywords
railway timetables, visualization, pattern detection

1 Introduction

This paper deals with the analysis of timetables as a preprocessing step for the rolling stock rotation planning. The timetables in question cover a whole year with 45000 and more trips. They stem from our industrial partner DB Fernverkehr AG (DBF). DB Fernverkehr AG operates the long distance passenger railway services in Germany. Unlike for the vehicles of the regional services, their vehicles do not commute between two terminal stations in general. Instead they frequent stations all over the country. In this way, idle times are reduced, which leads to a more efficient use of the available vehicles and an increased offer. However, this also greatly increases the requirements in the forward planning. A rotation plan should use as few vehicles and as few empty trips as possible to save financial and environmental resources. Also, the timely maintenance of the vehicles has to be ensured and regularity requirements assure the functioning of operational processes. Indeed the process of rolling stock rotation planning starts as early as five years before a timetable takes effect.

Here we consider the tactical planning stage, i.e. the processing that we propose happens 18 to 6 months before the timetable takes effect. At this point the timetable is split into two

*This work was funded by the RailLab of the Research Campus MODAL (Mathematical Optimization and Data Analysis Laboratories) funded by the German Ministry of Education and Research (BMBF).

seasons and contains variations that reflect public holidays and construction works. Operational aspects, such as technical and organisational problems that lead to delays, cancellations, re-routeing, and, ultimately, to a change of the rotations in the real-time management are not in the scope of our study. We aim to identify invariant sections of the timetable for which a cyclic rotation plan is adequate. These timetable patterns, the so called standard weeks, are also a result of the current planning process. However, we present a fully automated approach for their identification. As a result, they are available several months earlier in the planning process. The planning department of DBF wants to include our results into their current planning process and is already using it for testing purposes as it has several benefits. In fact, given two timetables it is now possible to identify changes immediately while the manual inspection of timetables could require several days. Furthermore, since each standard week requires a separate rotation plan, an automatic recognition of standard weeks in an early stage allows to estimate the effort for the planning in advance, which helps to evenly distribute the workload. Also, it helps to quantify the workload and, thus, to compare the workloads of different concepts. It should be noted that the recognition of standard weeks from the timetable has limitations. For example, it may be necessary to bridge the transition from one standard week to another to ensure that minimum turn times and maintenance intervals are met. This can only be discovered by the time the rotation plans of the standard weeks are available.

In addition, we present a visualisation of the timetable including the recognised standard weeks. Since we want to visualise the timetable for a whole year with several hundred trains, it is not feasible to make such a representation using existing graphic timetable diagram as for example in OpenTimeTable (Nash and Ullius (2004)). Instead we again rely on the identification of periodic patterns.

In the following section, we provide background information on the planning process at DBF and rotation planning in general. Then, we present our model for the recognition of standard weeks, which we call the *timetable pattern detection problem*. Subsequently, we present some computational results and recap our findings.

2 Prerequisites

2.1 Timetable

The timetables that we consider start on the Sunday of the second December weekend and end on the Saturday of the second December weekend in the subsequent year. The timetable does not simply give all the trips that are operated on a day. Instead the trips are organised as *trains*. A train is a set of recurring trips. It does *not* refer to a physical vehicle. In order that two distinct trips count as the same train, they have to lie on different dates, they have to cover a similar route and their departure and arrival times along this route have to be similar. As the assignment of trips to trains is given by the timetable, we need not go into more detail. Typically, a train has the same trips from Monday to Friday, but the trips on the weekend differ slightly, e.g. because they cover additional or fewer stops. Hence, the departure and arrival times may differ by a few minutes. At times this induces a period of 7 days on the timetable. An example is given by Figure 1, which shows train 685 that goes from Hamburg to Munich on Friday, Saturday and Sunday (indicated by a lighter blue colour), but already terminates in Nuremberg on the other days of the week (indicated by the darker blue colour).

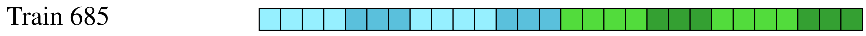

Figure 1: The image shows the timetable of train 685 from Monday, July 4 to Sunday, 31 July 2016. Different colours of the boxes are used to indicate different flavours of the trips.

Breaks in the timetable occur, for example, in the summer when parts of the track are closed down for several weeks of construction works. We can see in Figure 1 that the colours of the boxes change from blue to green. This indicates a change of the trips. Here, train 685 was rerouted due to construction works between Hanover and Göttingen in the summer 2016.

Input Format

As we have already seen in the example of train 685, it is not necessary to consider all details of the timetable for our study. We do not need to know that it is rerouted between Hanover and Göttingen, but it is sufficient to know that its trips change at a certain point in time. Hence, to reduce the complexity of the input, we will not consider a trip of the timetable with all of its stops, departure and arrival times, and so on. Instead we assume that the trips of a train come in different flavours (or, considering Figure 1, colours). We associate a number with each flavour of a train and, using this number, the timetable specifies which flavour a train has on any day of the timetable period. Algorithmically, we obtain this reduced form of the timetable by a simple preprocessing.

2.2 Rolling Stock Rotation Planning

The creation of rolling stock rotation plans is beyond the scope of this article. However, it is useful to discuss a few key concepts. The general problem is to assign a series of activities to every physical vehicle, such that all trips from the timetable are covered and maintenance constraints and organisational constraints are met. The objective is to minimise the operational costs. We refer to Borndörfer et al. (2016) for more detailed information on this topic. It is noteworthy that, during the tactical planning phase at DB Fernverkehr AG, the majority of the year is covered using so-called *standard weeks* as rotation plans. A standard week is a cyclic rotation plan with a period of one week. If a break in the timetable occurs, it is often possible to transition from one standard week to another. Periods where standard weeks *cannot* be used include public holidays in Germany like Christmas and Easter. They are called *deviations* and require a non-cyclical rotation planning approach. Also, if two consecutive standard weeks differ profoundly, it may be necessary to introduce a deviation to transition between these standard weeks. For technical reasons, the length of any deviation is always a multiple of 7.

2.3 Planning Process

The planning processes in railway companies are very diverse and exist in several time scales. On the one hand, some decisions, e.g. on the extension of the infrastructure, have effects for the decades to come and take years to implement. On the other hand, some decisions like the re-routeing of a train to avoid broken track affect the immediate future. Accordingly, the stages of the planning process are often categorised into strategic, tactical and operational stages (see Haahr et al. (2015)). The creation of rotation plans, as considered in this paper, can be classified as part of the tactical planning stage. The earliest occurrence of a rotation plan is five years ahead of the timetable taking effect. Of course, this first draft is rather sketchy. It is stated in more detail 18 months before the timetable takes effect, when the first standard week is constructed (i.e. a cyclic rotation plan with a period of seven days, that it is intended to cover the whole year). At this level, other departments such as the marketing may request changes. This way, the timetable gets more detailed progressively. Therefore, it is necessary to develop the rotation plan in parallel and to adapt it to changes made to the timetable frequently. The changes of the timetable include, e.g., special events like public holidays and construction works. Therefore, it is no longer possible to cover the whole timetable with one standard week. Instead, the need to identify invariant time periods in the timetable that can be covered by one standard week each arises. This problem is addressed in the subsequent section.

3 Model

3.1 Timetable

In this section we describe formally how we determine the timetable patterns that are the basis for the visualisation. As noted in Section 2.1 most details of the timetable are discarded by a preprocessing step, essentially leaving a sequence of flavours, which is represented by numbers. In the following we regard a timetable as the schedule for one train while, usually, one calls the schedules of all trains the timetable. According to our definitions the schedules of all trains constitute the *combined timetable*.

Definition 1 *A* timetable $(t_i)_{i=1,\dots,n}$ *is a sequence of n natural numbers, where n is a multiple of 7.*

We start with a notion of a timetable pattern which is as general as possible. We will then let our algorithm sieve undesired patterns. Since the timetable has periodic components, it is important to keep track of the weekdays.

Definition 2 *A* timetable pattern *is a subsequence of 7 consecutive elements of a timetable:* $t_k, \dots, t_{k+6} \subseteq (t_i)$.

Definition 3 *Let $\%$ denote the modulo operator. An element t_k of a timetable pattern or a timetable is called* Sunday *if and only if $k\%7 = 1$. Accordingly, any element of a timetable is called a* day *of the timetable.*

Corollary 4 *Every timetable pattern has exactly one Sunday.*

Definition 5 *Let $(t_k, \dots, t_{k+6})$ be a timetable pattern and t_{k+i} ($i \in \{0, \dots, 6\}$) be one of its elements. Then, we call $t_{k+((i\pm1)\%7)}$ its* successor $s(e)$ *or* predecessor $p(e)$*, respectively. Let b be the Sunday of a timetable sequence B. Then, we call the sequence $(b, s(b), s^{(2)}(b), \dots, s^{(6)}(b))$ the* normal form *of B. Two timetable sequences A and B agree, by definition, under the equivalence relation (E) if and only if their normal forms are identical. We shall write $A \sim B$ to denote the equivalence of A and B under (E).*

Definition 6 *Let A be a timetable pattern of a timetable T and S be a sequence of at least 8 consecutive days of T. We say that A covers S if every timetable pattern B in S is equivalent to A, i.e. $A \sim B$, under (E).*

According to Definition 1 a timetable is a simple sequence of numbers. However, the input gives such a timetable for every train. From the timetables of the individual trains, we compile a timetable of all trains, which can be used to seek standard weeks.

Definition 7 *Given l timetables $(t_{1,i})_{i=1,...,n}, \ldots, (t_{l,i})_{i=1,...,n}$ of length n, the* combined timetable $(\hat{t}_i)_{i=1,...,n}$ *is a timetable with the following property: $\hat{t}_i = \hat{t}_{i+7k} \Leftrightarrow t_{j,i} = t_{j,i+7k} \forall j = 1, \ldots, l$ for any $i, k \in \mathbb{N}$ with $i < n$ and $i + 7k \leq n$.*

In other words, Definition 7 says that two days in the combined timetable are equal if they are equal in all train timetables. This is illustrated in Table 1. The Sundays of the three timetables t_1, t_2 and t_3 agree. Hence, they also agree in the combined timetable $\hat{t}$. The second timetable t_2 differs on the second Wednesday and t_1 and t_3 differ on the last Wednesday. Therefore, the combined timetable has a different number for all Wednesdays. Note that we generally compare two days of the timetable only if they correspond to the same day of the week. For example, Friday, Saturday and Sunday differ from the other days in the first week in Table 1, but this is not reflected in the combined timetable. Also, the choice of numbers is arbitrary to some extend. In Table 1, the smallest unused number was chosen for every entry of the combined timetable. While this is a reasonable convention, several other numberings would also conform to Definition 7.

Table 1: Example of a Combined Timetable

Week	Timetable	Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	t_1	1	2	2	2	2	1	1
	t_2	1	1	1	1	1	1	1
	t_3	2	1	1	1	1	1	2
	$\hat{t}$	1	1	1	1	1	1	1
2	t_1	1	2	2	2	2	1	1
	t_2	1	1	1	2	1	1	1
	t_3	2	1	1	1	1	1	2
	$\hat{t}$	1	1	1	2	1	1	1
3	t_1	1	3	3	3	3	4	4
	t_2	1	1	1	1	1	1	1
	t_3	2	1	1	3	1	1	2
	$\hat{t}$	1	2	2	3	2	2	2

3.2 Pattern Detection

Preprocessing

To determine the patterns that are suggested to the planning department we determine all patterns that cover a part of the combined timetable as defined in Definition 6. Algorithmically, this is achieved by scanning a timetable from the beginning to the end and saving the normal form of every pattern that is found to a list. In a second run every day is equipped

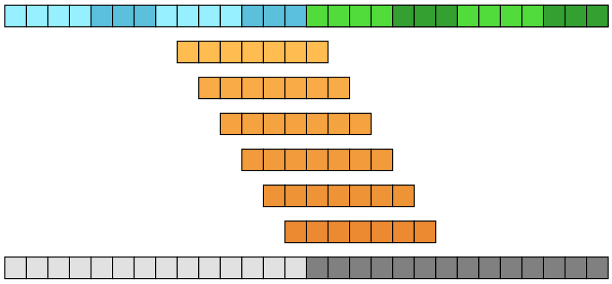

Figure 2: The image shows the train 685 and several patterns that could be found during the preprocessing step.

with a list that contains all patterns that occur during this day. This way we obtain possible patterns for every train and the combined timetable. The patterns may overlap. They can occur for only seven days or up to several weeks. A pattern need not occur without interruption.

Figure 2 shows the timetable of train 685, which was introduced in Section 2.1. Recall that 685 is rerouted due to construction works, which is indicated by a change of the trip flavours from blue to green. Obviously, the trips before and after the break form two separate patterns. These patterns are depicted in grey. However, combining Tuesday to Sunday before the break with Monday after the break, also gives a pattern. Hence, we obtain the six orange patterns by combining trips from before and after the break. These patterns are examples of patterns that we do not want to present to planners. They occur only once and they do not help to understand the structure of the timetable in anyway. This is only one situation where undesired patterns occur. Similar patterns occur, if we have a regular pattern, but the timetable has a deviation on a single day. Some of these undesired patterns can be removed by adding filtering rules to the preprocessing. For example, it is reasonable to remove patterns that cover only exactly seven days. Still, some ambiguity remains after the preprocessing. In the following, we show the two strategies we used to determine eligible patterns and to neglect undesired patterns. The first strategy is applied to the combined timetable, while the second strategy is applied to the individual train timetables.

Choosing Patterns from the Combined Timetable

We associate a binary variable y_j with pattern $j(j = 1, \ldots, m)$ that was found by the preprocessing step. This variable is one if j is eligible or 0 if the pattern should be neglected. Each day $i(i = 1, \ldots, n)$ is associated with a variable x_i that determines if i is covered by one of the eligible patterns $(x_i = 1)$ or whether it is not covered $(x_i = 0)$. In the latter case, the day will be marked as a deviation.

We use the following mixed integer linear program, which we call the *timetable pattern* *detection problem*, to determine a set of eligible patterns.

$$\begin{aligned} \min \quad & -\sum_{i=1}^{n} x_i + 8\sum_{j=i}^{m} y_j & & (1) \\ \text{s.t.} \quad & x_i - \sum_{j:j \text{ covers } i} y_j \leq 0 & \forall i = 1,\dots,n \\ & x_i \in [0,1] & \forall i = 1,\dots,n \\ & y_j \in \{0,1\} & \forall j = 1,\dots,m \end{aligned}$$

The idea is that the use of patterns induces costs which can be compensated, and hopefully overcompensated, by the days that are covered by this pattern, since every covered day induces a benefit. Arguably the cost benefit ratio of 8 is somewhat arbitrary. It was chosen because the period of the timetable is one week. Hence, one needs to consider at least 8 consecutive days in order to obtain a repetitive day. Shorter patterns are generally not desirable and should not be detected as eligible.

Mathematically the above mixed integer linear program is closely related to the set cover problem. If we enforce the constraints $x_i = 1 (i = 1,\dots,n)$, i.e. all days have to be covered, we obtain the set cover problem. To get back to (1), one moves the newly introduced constraints to the objective function as in a Lagrangian relaxation approach. This leads to a penalty for days that are not covered.

Alternative Analysis of Train Timetables

The sole visualisation of the suggested standard weeks or patterns is only of limited use if a planner cannot identify the cause of a deviation or a change of the pattern right away. Hence, the timetables of the individual trains should also be visualised. Naturally, the train timetables have much more regularity than the combined timetable, as the latter is a composition of all train timetables. In most cases, a train timetable has a regular main pattern. This pattern may be (temporarily) replaced by another pattern, e.g. for a middle-term construction site. In other cases there may be single deviations for a period up to two weeks, e.g. for Christmas and New Year. Although it may be possible to form new patterns with these deviations as it would be done for the combined timetable, it is desirable to make these deviations more transparent to the planner. Hence, we use a different approach to detect eligible patterns for the train timetables. For each pattern (cf. Definition 6), we count the days covered by the pattern. If they cover more than three weeks of the timetable, they are considered as long patterns, otherwise as short patterns. The long patterns are always considered as eligible patterns. The short patterns are compared to the long patterns. If they agree on at least 4 out of 7 days with a long pattern, they are removed. (In the visualisation, they will be represented as the long pattern with up to three deviations.) Undesired patterns as in Figure 1 are avoided by enforcing that at least one day is covered uniquely by a pattern in the preprocessing step. Patterns that do not satisfy this criterion are discarded.

4 Computations and Results

We used six sample timetables provided by DB Fernverkehr AG to test our algorithm, which we implemented in Python. As can be seen from Table 2, the running time of the procedure is negligible. The same holds for the memory consumption. It is noteworthy that the majority of the running time stems from the I/O operations, the parsing of the XML timetable, and the creation of the visualisation as a spreadsheet, but not from the analysis of the timetable (which is accomplished in less than a second for the biggest timetable). The tests were performed on a standard workstation running Linux. The mixed integer linear program was solved using the python interface of the academic MIP solver SCIP (Maher et al. (2016); Gamrath et al. (2016)).

Table 2: Test Instances

Instance	Trains	Trips	Wall-clock time
2016_402	248	46782	5.94s
2016_411-415	272	47134	7.05s
2017_402	174	43950	4.74s
2017_403-406	255	54357	6.93s
2017_403-406-407	295	65175	8.31s
2017_407	43	11900	1.06s

Because of its size it is difficult to present the visualisation in a printed format. Generally, our algorithm produces the timetable visualisation as a spreadsheet. An alternative is the presentation in PDF, which is sketched in Figure 3 for the 2016_402 instance. Although, it is not possible to recognise details in this sketch, it is still possible to extract some information from it. The x-axis contains all dates from 13 December 2015 to 10 December 2016. The y-axis shows all trains of the 402 series (ICE 2). Different colours represent different timetable patterns. Grey is used to represent the first timetable pattern of the year, blue is used for the second pattern and so on. Red is used to represent deviations and yellow is used for trains that do not run on any day of the week. Every 20th row (including the first row) is separated from the other rows by blank rows and contains the patterns which were determined for the combined timetable.

On the very left, one can recognise an accumulation of red dots within two vertical black lines. These are deviations of the trains during the Christmas season. The black lines mark the beginning and the end of a deviation period that our algorithm determined for the combined timetable. Over the course of the timetable there are several such accumulations, which are mostly caused by German public holidays. On the right-hand side of the sketch, one can clearly recognise an extended period where several trains change from a grey to a blue pattern. During this time several trains are rerouted due to construction sites. To enable the reader who does not have an electronic copy of this document, which allows zooming, to see more details, a magnified excerpt of the upper left part of the sample visualisation is given in Figure 4 in the appendix.

Furthermore, the patterns obtained for the combined timetable including the patterns as used by DB Fernverkehr AG are given in Table 3 in the appendix. The patterns are presented as different letters, where deviations are denoted by an A. Sundays are set in boldface. The time ranges from 13 December 2015 to 15 December 2016. Although there appear several differences between the patterns of DB Fernverkehr AG and our algorithm, most of them are easily explainable. It is important to notice that no one-to-one comparison between the two data sets is not possible. Our algorithm can only be used with a timetable that serves as an input for the rotation planners. As the planning progresses the data is processed within

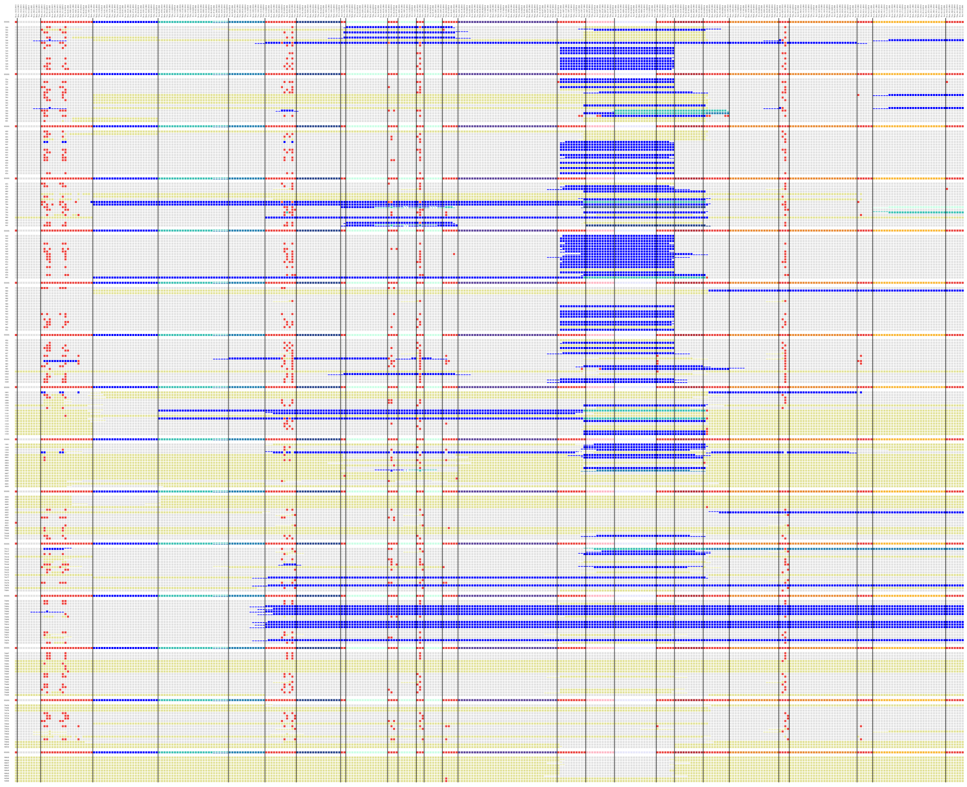

Figure 3: Timetable visualisation: Detailed detected patterns in the appendix. Every column corresponds to one day. Every row corresponds to one train. The rows separated by blank rows show the standard weeks obtained from the combined timetable. The black vertical lines are used to indicate the end of a standard week or deviations. Red is used for deviations and yellow is used for trains that do not run on any day of the week. The other colours, starting with grey and blue, are used to indicate distinct patterns/standard weeks.

different internal data processing systems, but the timetable is not maintained in its original file format. Also, as mentioned before, our algorithm's pattern detection is solely based on the timetable. Hence, it cannot detect if a deviation is necessary to bridge two standard weeks. Such a deviation occurs for example in the transition from pattern K to L. For the transition from F to G, our algorithm detects an overlap of the patterns, which is denoted by an asterisk (*). This is because there is just one small change. Train 984 starts running on Fridays beginning with 4 March 16. Therefore, it is up to free choice whether the other days between the two patterns are assigned to the former or to F or G.

For technical reasons all deviations need to have a length of seven or a multiple of seven (if this is not the case in Table 3, the reason is that the patterns from DBF were compiled from three rotations), while our algorithm makes deviations as short as possible. This effect is even increased if deviations are put such that they begin on a Wednesday and end on a Tuesday, which recently has been established as a best practice for process-related reasons. A good example of this is the transition from pattern N to O.

While we only have only access to this one example of patterns from DB Fernverkehr AG and a comparison with our algorithm's output seems intricate, our visualisation was tested by DB Fernverkehr AG themselves and according to the planning department it is deemed useful throughout the department.

5 Résumé

We presented parts of the planning processes of our industrial partner which lead to the timetable pattern detection problem. Due to its small size this problem does not lead to computational difficulties and it is possible to detect standard weeks and deviations of a large-scale timetable within a reasonable time. Together with the visualisation of the timetable that we propose, this seems to be of great value for our industrial partner. It is now possible to inspect timetables and to track changes within minutes while a time-consuming manual inspection was necessary beforehand. Since the timetables and the corresponding rotation plans are developed in parallel, rotation planners frequently obtain revised timetables and our tool can be of meaningful assistance every time. This proves again that mathematics is an adequate tool to support and improve processes in real-world business applications.

Appendix

Table 3: Comparison of patterns (for the combined timetable) from DB Fernverkehr AG and our algorithm for the instance 2016_402. Sundays are emphasised in boldface. The absence of a timetable pattern (a deviation) is marked by a capital A. For the cells marked with an asterisk (*), the algorithm recognised the patterns F and G.

DB F	**C**	C	C	A	A	A	A	**A**	A	A	A	A	A	A	**A**	A	A	A	A	A	A	**A**	A	A	A	A	A	A
ALGO	**A**	D	D	D	D	D	D	**D**	D	D	A	A	A	A	**A**	A	A	A	A	A	A	**A**	A	A	A	A	A	A
DB F	**A**	A	A	E	E	E	E	**E**	E	E	E	E	E	E	**E**	E	E	E	E	E	E	**E**	E	E	E	E	E	F
ALGO	**A**	A	E	E	E	E	E	**E**	E	E	E	E	E	E	**E**	E	E	E	E	E	E	**E**	E	E	E	E	E	F
DB F	**F**	F	F	F	F	F	F	**F**	F	F	F	F	F	F	**F**	F	F	F	F	F	F	**F**	G	G	G	G	G	G
ALGO	**F**	F	F	F	F	F	F	**F**	F	F	F	F	F	F	**F**	F	F	F	F	F	*	*****	*	*	*	*	G	G
DB F	**G**	G	G	G	G	G	G	**G**	G	G	G	G	A	A	**A**	A	A	A	A	A	A	**A**	A	A	H	H	H	H
ALGO	**G**	G	G	G	G	G	G	**G**	G	G	G	G	A	A	**A**	A	A	A	A	A	A	**A**	A	A	H	H	H	H
DB F	**H**	H	H	H	H	H	H	**H**	H	H	H	H	H	?	**I**	I	I	I	I	I	I	**I**	I	I	I	I	I	I
ALGO	**H**	H	H	H	H	H	H	**H**	H	H	H	H	H	A	**A**	I	I	I	I	I	I	**I**	I	I	I	I	I	I
DB F	**I**	A	A	A	A	A	A	**A**	A	A	A	A	A	A	**A**	A	A	I	I	I	I	**I**	I	I	A	A	A	A
ALGO	**I**	I	I	A	A	A	A	**I**	I	I	I	I	I	I	**A**	A	A	I	I	I	I	**I**	I	I	A	A	A	A
DB F	**A**	A	A	X	X	X	X	**X**	X	X	A	A	A	A	**A**	A	A	J	J	J	J	**J**	J	J	J	J	J	J
ALGO	**A**	A	J	J	J	J	J	**J**	J	J	J	J	J	J	**J**	J	J	J	J	J	J	**J**	J	J	J	J	J	J
DB F	**J**	J	J	J	J	J	J	**J**	J	J	A	A	A	A	**A**	A	A	Y	Y	Y	Y	**Y**	K	K	K	K	K	K
ALGO	**J**	J	J	J	J	J	J	**J**	J	J	J	J	A	A	**A**	A	A	A	A	A	A	**A**	A	K	K	K	K	K
DB F	**K**	K	K	A	A	A	A	**A**	A	A	L	L	L	L	**L**	L	L	A	A	A	A	**A**	A	A	A	A	A	A
ALGO	**K**	K	K	K	K	K	L	**L**	L	L	L	L	L	L	**L**	L	L	L	L	L	L	**L**	A	A	A	A	A	A
DB F	**A**	A	A	A	A	A	A	**A**	A	A	A	A	A	A	**A**	A	A	A	A	A	A	**A**	A	A	N	N	N	N
ALGO	**A**	M	M	M	M	M	M	**M**	M	M	M	M	A	A	**A**	A	A	A	A	A	A	**A**	N	N	N	N	N	N
DB F	**N**	N	N	N	N	N	N	**N**	N	N	A	A	A	A	**A**	A	A	N	N	N	N	**N**	N	N	N	N	N	N
ALGO	**N**	N	N	N	N	N	N	**N**	N	N	N	N	N	A	**A**	A	A	N	N	N	N	**N**	N	N	N	N	N	N
DB F	**N**	N	N	N	N	N	N	**N**	N	N	A	A	A	A	**A**	A	A	A	A	A	A	**A**	A	A	O	O	O	O
ALGO	**N**	N	N	N	N	N	N	**N**	N	N	N	N	N	N	**N**	A	A	A	A	A	A	**O**	O	O	O	O	O	O
DB F	**O**	O	O	O	O	O	O	**O**	O	O	O	O	O	O	**O**	O	O	O	O	O	O	**O**	O	O	O	O	O	O
ALGO	**O**	O	O	O	O	O	O	**O**	O	O	O	O	O	O	**O**	O	O	O	O	O	O	**A**	A	A	A	A	A	A

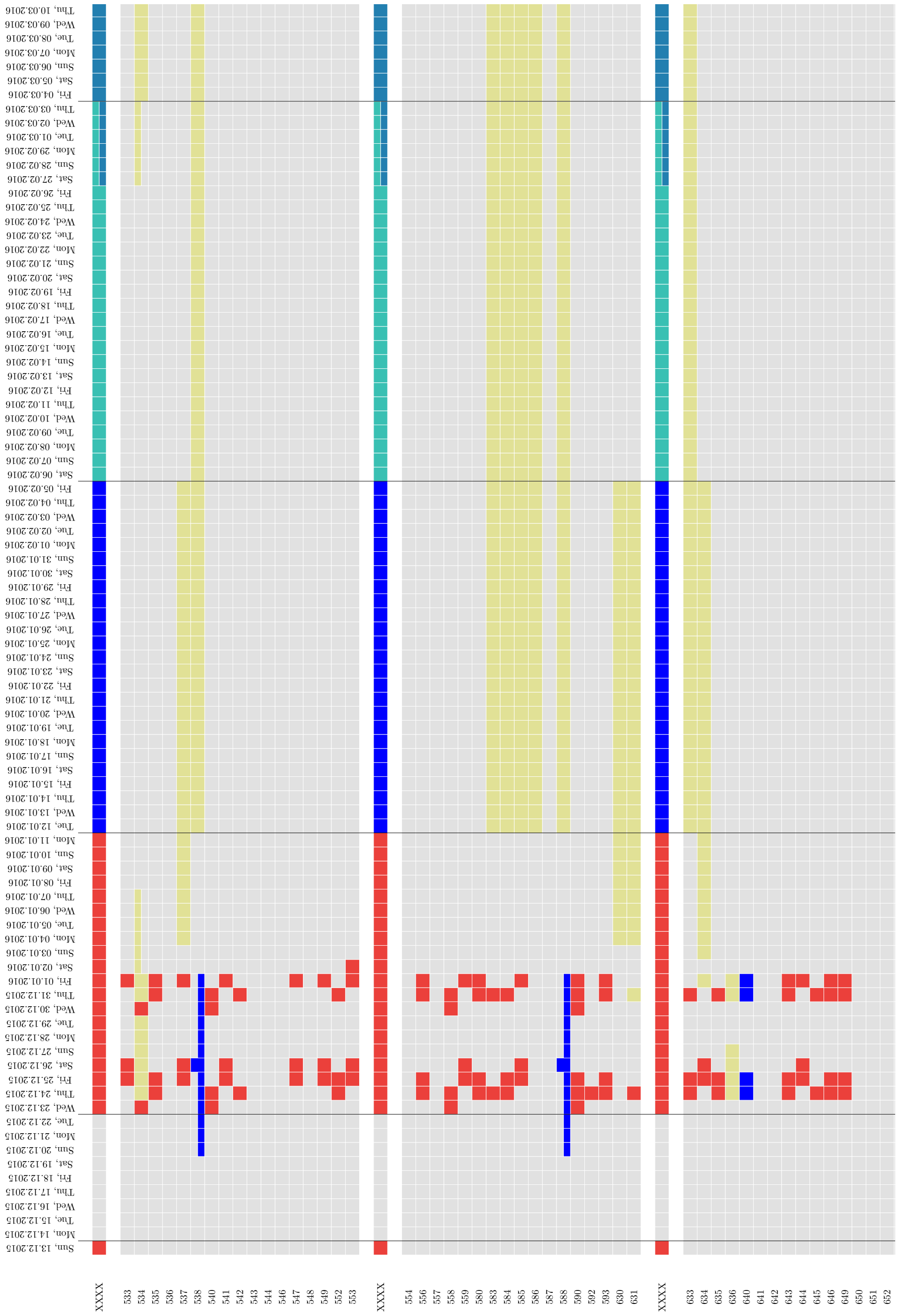

Figure 4: Excerpt of PDF timetable visualisation

References

Borndörfer, R., Reuther, M., Schlechte, T., Waas, K., Weider, S., 2016. “Integrated Optimization of Rolling Stock Rotations for Intercity Railways.” *Transportation Science*, vol. 50(3), pp. 863–877.

Gamrath, G., Fischer, T., Gally, T., Gleixner, A.M., Hendel, G., Koch, T., Maher, S.J., Miltenberger, M., Müller, B., Pfetsch, M.E., Puchert, C., Rehfeldt, D., Schenker, S., Schwarz, R., Serrano, F., Shinano, Y., Vigerske, S., Weninger, D., Winkler, M., Witt, J.T., Witzig, J., 2016. The SCIP Optimization Suite 3.2., ZIB Technical Report 15-60, ZIB, Takustr. 7, 14195 Berlin.

Haahr, J.T., Pisinger, D., Larsen, J., 2015. Reactive Robustness and Integrated Approaches for Railway Optimization Problems, PhD thesis, DTU Management Engineering.

Maher, S.J., Miltenberger, M., Pedroso, J.P., Rehfeldt, D., Schwarz, R., Serrano, F., 2016. Mathematical Programming in Python with the SCIP Optimization Suite. In *Mathematical Software – ICMS 2016*, vol. 9725, pp. 301–307.

Nash, A., Ullius, M., 2004. “Optimizing railway timetables with OpenTimeTable.” *WIT Transactions on The Built Environment*, vol. 74.