Georg-August-Universität Göttingen

Computing Delay Resistant Railway Timetables

C. Liebchen, M. Schachtebeck, A. Schöbel,

S. Stiller, A. Prigge

Nr. 2007-24

Preprint-Serie des
Instituts für Numerische und Angewandte Mathematik
Lotzestr. 16-18
D - 37083 Göttingen

Computing Delay Resistant Railway Timetables*

Christian Liebchen Michael Schachtebeck
Anita Schöbel Sebastian Stiller André Prigge

November 11, 2008

Abstract

In the past, much research has been dedicated to compute optimum railway timetables. A typical objective has been the minimization of passenger waiting times. But only the planned nominal waiting times have been addressed, whereas delays as they occur in daily operations have been neglected. Delays have been rather treated mainly in an online context and solved as a separate optimization problem, called delay management.

We provide the first computational study which aims at computing delay resistant periodic timetables. In particular we assess the delay resistance of a timetable by evaluating it subject to several delay scenarios to which optimum delay management will be applied.

We arrive at computing delay resistant timetables by selecting a new objective function which we design to be somehow *in the middle* of the traditional simple timetabling objective and the sophisticated delay management objective. This is a slight extension of the concept of "Light Robustness" (LR) as it has been proposed by Fischetti and Monaci (2006). Moreover, in our application we are able to provide accurate interpretations for the ingredients of LR. We apply this new technique to real-world data of a part of the German railway network of Deutsche Bahn AG. Our computational results suggest that a significant decrease of passenger delays can be obtained at a relatively small *price of robustness*, i.e. by increasing the nominal travel times of the passengers.

*This work was partially supported by the Future and Emerging Technologies Unit of EC (IST priority - 6th FP), under contract no. FP6-021235-2 (project ARRIVAL) and by the DFG Research Center MATHEON in Berlin.

1 Introduction

1.1 Background

Timetabling is among the most important tasks for optimization in public transport. Not surprisingly the construction of timetables is a well studied problem in the literature and has been treated under various objective functions. Besides technical restrictions and optimization of costs, the main focus lies on finding timetables which are optimal from the passengers' point of view. However, in most papers only the nominal travel times are considered while the possibility of delays and stochastic changes is neglected. In reality delay is a considerable phenomenon in almost all public transport systems. It plays a particular role for customers' satisfaction.
There is of course a trivial way to come up with delay resistant timetables: Simply add huge time supplements on each trip. However, it is obvious that such a solution is unacceptable for passengers. The art of delay resistant timetabling hence is to achieve a certain level of robustness by a *minimum* increase in nominal travel times.
Recently, theoretical studies have been directed towards the problem of designing delay resistant timetables ([14, 21]). Kroon et al. consider single trains and non-periodic corridors in a sampling approach. In both papers the construction of delay resistant timetables is considered as a matter of most effectively placing a limited amount of slack time in the timetable. The authors present optimization techniques aimed at minimizing the expected delay for various topologies and network sizes. Although a variety of different methods and settings is considered, all of this is done under the following assumption: When the timetable is operated, the reaction towards delay will follow a simple pattern, i.e. either always to wait for delayed trains or always not to wait for delayed trains. Obviously, more sophisticated schemes are possible.
There exists some literature on how to deal with delays when they occur and find a disposition timetable which is as convenient as possible for the passengers under the given circumstances. This problem is called the *delay management problem* and includes to decide which connections between trains should be maintained and which not, see [30] and references therein.

1.2 Research objective

It would be desirable to optimize delay resistant timetables with respect to an optimal delay management. But as the latter is already a hard problem for a fixed plan and a fixed scenario of delays, a full integration of the two is out of computational reach. However, we can clarify the following question: Given a real-world data set for which optimal delay management is possible within practical computation time. How does a delay resistant timetable that has been optimized while assuming a simplified delay management behave? The question is whether such timetables keep their promise, namely that disturbances do not affect the quality of service to the passengers too much if good delay management strategies are finally applied.

Although this is a straightforward question, to the best of our knowledge, it has not been treated in the literature before. In this paper we bridge this gap between delay resistant timetabling and delay management. We consider a real-world railway network. In a first step, we apply a new technique to optimize periodic timetables for different degrees of robustness. This is a slight extension of the paradigm of "Light Robustness" as it has been proposed by Fischetti and Monaci [6]. The resulting timetables are in a second step confronted with different sets of disturbances. By solving the delay management problem for each of these delay scenarios, we obtain optimal disposition timetables from which we evaluate how much the passengers are affected by the disturbances. We hence obtain a new empirical measure for the quality of the original timetable.

This case study also allows to evaluate the power of optimal delay management. We will also calculate the delay experienced by the passengers if the same timetables and the same set of source-delays[1] are managed by very simple delay management policies.

In some sense we combine the expertise of state-of-the-art delay management and topical delay resistant timetabling techniques in order to overcome the shortcomings of both. On the one hand, the optimization of timetables cannot take into account a complicated delay management, but it optimizes with respect to *all* scenarios of delay. On the other hand, delay management can only be repeated for a small set of scenarios—in fact a very small set in relation to the set of all scenarios. It is, however, optimal for each of these.

It turns out that the effects of delays are less profound in the delay resistant timetables that we construct in our study than in a conventional timetable. In addition, it becomes clear that a good delay management provides even better disposition timetables than obtained by the simplest delay management policies. Finally, the delay resistant timetabling proves particularly effective against short delays.

[1]Other authors refer to what we call here source-delay as *primary delay*.

1.3 Related Work

The only comparable work we are aware of is due to Engelhardt-Funke and Kolonko, see [5]. Unlike ours their approach is able to integrate the delay management into the construction of the timetables by evolutionary algorithms. However, their evaluation is not based on an optimal delay management and solely relies on a limited number of scenarios. Our approach is not able to integrate timetabling and delay management, but solves both steps exactly: We construct *optimal* timetables with respect to the expectation over all scenarios and evaluate them with an *optimal* delay management policy for a limited number of scenarios.
Note that the objective function we use to *optimize* is similar to the one introduced by PTV AG, Germany, in its planning software VISUM in order to enrich their *evaluation* by some penalty for tight—and thus vulnerable—transfers.

1.4 Outline

The remainder of the paper is structured as follows. In Section 2 we define our models for periodic timetabling, stochastic disturbances, and delay management. We also describe in detail how event-activity networks can be used to model the periodic event scheduling problem as well as the delay management problem. Our integer programming approach to designing delay resistant timetables, using these as input for defining delay management problems, and solving these problems are developed in Section 3. In Section 4 we describe the real-world data we used for our case study. Finally, we present the results of the case study in Section 5 and give some suggestions for further research.

2 Models

We consider two mathematical optimization problems: In our first step we compute optimal periodic timetables with a (light) robustness against delays incorporated to the objective function; in the second step we solve the delay management problem, i.e. we determine a disposition timetable to react on a given set of (small) disturbances. Since both problems are timetabling problems, we start by recalling the concept of event-activity networks which can be used to model both of our problems.

2.1 Event-activity networks in timetabling

The basic graph model that we use is the Feasible Differential Problem (FDP), see [26]. In an instance $\mathcal{I} = (D, \ell, u)$ of FDP where $D = (V, A)$ is a directed graph and $\ell, u \in \mathbb{Q}^A$, a vector $\pi \in \mathbb{Q}^V$ is sought such that

$$\ell_a \leq \pi_j - \pi_i \leq u_a \quad \forall a = (i, j) \in A. \tag{1}$$

By shortest path (SP) techniques, feasibility of FDP can be easily checked: model lower and upper bounds as anti-parallel arcs and assign to π SP distance labels.
The model can be illustrated in an event-activity network $D = (V, A)$ in which the nodes V are called *events* and correspond to arrivals V_{arr} or departures V_{dep} of trains at stations. The vector π is called a *timetable*, the value π_i is the time instant at which the event i is scheduled.
The precedence constraints (1) are used to define the arcs (also called *activities*) of the network. Observe that the quantities $\pi_j - \pi_i$ are time durations. An activity $a = (i, j)$ makes sure that event j is scheduled after event i, in particular

$$\pi_j \in [\pi_i + \ell_a, \pi_i + u_a].$$

We distinguish the following five different types of activities:

$$A = A_{\text{drive}} \cup A_{\text{stop}} \cup A_{\text{transfer}} \cup A_{\text{turn}} \cup A_{\text{head}}.$$

A_{drive} models the driving of a train between two consecutive stations and A_{stop} models the waiting of a train within a station to let passengers get on or off. An arc $a = (i, j) \in A_{\text{transfer}}$ makes sure that passengers can transfer from a train arrival i to a train departure j. Similarly, the arcs in A_{turn} connect different trips of the same train and hence ensure that the vehicle schedules are respected.

A different concept is used for the arcs in A_{head}. The arcs in A_{head} model the limited capacity of the track system. They come in pairs $a = (i, j)$ and $a^{-1} = (j, i)$ usually linking two departure events i and j. It is required that for each pair, exactly one of the two corresponding precedence constraints, which only consist of a lower bound each, is respected. In more detail, either $\pi_i \geq \pi_j + \ell_{a^{-1}}$, meaning that event j is scheduled before event i, *or* $\pi_j \geq \pi_i + \ell_a$, where event j is scheduled after event i. The lower bounds $\ell_{a^{-1}}$ and ℓ_a model the headway that has to be respected between two consecutive (departure) events.
Here, one could be tempted to assume that simply reorienting one of these two antiparallel arcs, say a^{-1}, enables us to turn such a pair of disjunctive

constraints into a standard FDP constraint, because formally this is nothing but

$$\ell_a \leq \pi_j - \pi_i \leq -\ell_{a^{-1}}.$$

Rather, the problem is that under realistic assumptions such as $\ell_a > 0$ and $\ell_{a^{-1}} > 0$, this does obviously constitute some infeasible FDP constraint. Hence, we have to relax exactly one constraint of the two arcs a and a^{-1} in each pair of disjunctive headway constraints, hereby explaining the special role of A_{head}.

As an example, Figure 2 shows the (small) event-activity network corresponding to the public transportation network of Figure 1.

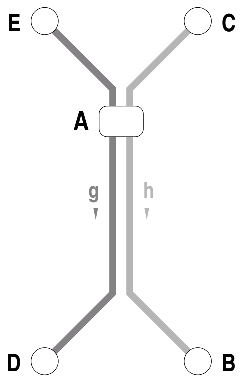

Figure 1: A part of a public transportation network with five stations and two trains. After their departure in A both trains have to use the same track.

In a *periodic timetable*, trains are grouped into lines which are required to be operated with some periodicity T. In the case of $T = 120$ minutes, this means that if one train of some fixed line starts its trip at 10:05, then there will be a train five minutes past every even hour. Throughout this paper, we assume that all lines have the same period T.

We obtain subsets of non-periodic events i_k that take place at time $\pi_{i_k} = \pi_i + kT$ for $k \in \mathbb{Z}$. Such a set represents the departure (or arrival) of all trains of the same line at a specific station and will in the following be represented by one periodic event i. We can hence reduce the event-activity network above to a periodic one,

$$\underline{D} = (\underline{V}, \underline{A}),$$

in which $\underline{V}$ consists of equivalence classes of events in V. For $i \in \underline{V}$ let us denote by $V(i)$ the set of non-periodic events belonging to a given periodic event i.

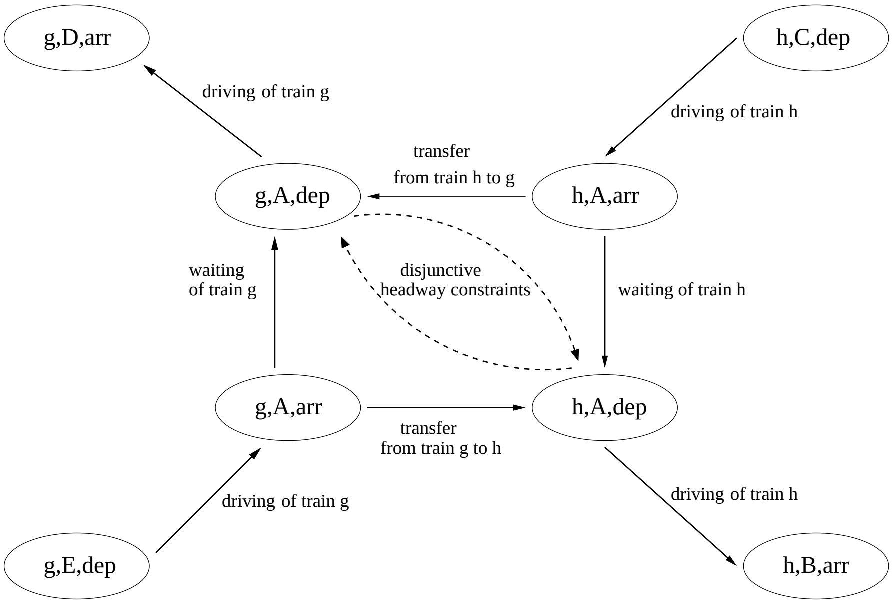

Figure 2: The event-activity network corresponding to Figure 1.

It is convenient to think of the equivalence classes $i = \{i_k\}_k$ as periodic events with π_i as the periodic time assigned to i. As in such a periodic system every action repeats after the period time T, one can assume w.l.o.g. that $\pi_i \in [0, T)$.

The decision problem that is widely used for periodic timetabling is the Periodic Event Scheduling Problem (PESP), see [33]. In addition to the input to an instance of FDP, a fixed constant period time T is specified. Notice that in other models for periodic timetabling, such as the ones in [3, 13], the number of variables contains this number T as a factor. In contrast, in any of our computations we will use T only in its standard logarithmic encoding. In T-PESP—or PESP for short—one looks for a vector $\pi \in [0, T)^{\underline{V}}$ such that

$$\forall a = (i, j) \in \underline{A}\, \exists k_a \in \mathbb{Z} : \ \ell_a \leq \pi_j - \pi_i + T \cdot k_a \leq u_a. \tag{2}$$

The decision problem whether a given network admits a feasible solution π is NP-complete because it generalizes Vertex Colorability ([24]). As in the non-periodic case, we also partition the set $\underline{A}$ of periodic constraints into five subsets

$$\underline{A} = \underline{A}_{\text{drive}} \cup \underline{A}_{\text{stop}} \cup \underline{A}_{\text{transfer}} \cup \underline{A}_{\text{turn}} \cup \underline{A}_{\text{head}}.$$

Observe that in the periodic case, the headway constraints $\underline{A}_{\text{head}}$ do not play any special role because we may simply define $u_a := T - \ell_a$, where ℓ_a is the minimum headway time.

The PESP is the building block of many studies on periodic railway timetabling ([28, 23, 22, 15, 16]). In particular, the first mathematically optimized timetable that has been used in daily operation has been computed based on the PESP ([17]).

Since most public transportation companies in Europe operate their networks subject to periodic timetables, we compute periodic timetables as the regular service timetables in our case study. In contrast, the goal of any disposition timetable is to react on the specific disturbances that occur during each individual day of operation by tailored decisions. Hence, disposition timetables always have to be non-periodic.
If the time duration that a timetable π defines for an activity a exceeds its lower bound ℓ_a, we speak of *slack*. Its amount is given by

$$(\pi_j - \pi_i) - \ell_a \geq 0 \qquad \text{or} \qquad (\pi_j - \pi_i + T \cdot k_a) - \ell_a \geq 0$$

in the non-periodic or in the periodic case, respectively. In the pure feasibility problems FDP and PESP, the quality of a timetable can be ensured by defining relatively strict upper bounds $u_a - \ell_a \ll T$ on the slack, e.g. for the most important transfer activities.
The most important goal for deterministic timetable optimization is to minimize the total passengers' transfer times in the network. To this end, we must be given the number of passengers w_a for each transfer activity $a \in A_{\text{transfer}}$ or $a \in \underline{A}_{\text{transfer}}$, respectively. Given this weight we have to minimize the linear objective function of the weighted sum of slack. A linear objective function can also be used to minimize other types of cost in a timetable, see ([18]) for details. If a linear objective function is added to an instance of PESP, we speak of an *optimization instance of PESP*.

2.2 The Source of Delay

We assume the driving time of each train to vary according to a known probability distribution. The lower bounds for arcs a belonging to $\underline{A}_{\text{drive}}$ or A_{drive}, corresponding to a single (periodic) train ride, are random variables $\ell_a : R \to \mathbb{Q}^+$ where R is the set of all scenarios.
In case this random variable ℓ_a exceeds a certain, 'usual' travel time for arc a, we speak of a *source delay*. We distinguish between source delays, i.e. the seminal prolongation of a trains' driving time, and delays[2] which result from source delays. A source delay may cause several delays at different stations, even for trains that have not themselves been subject to a source delay,

[2]Sometimes, such delays are referred to alternatively as *secondary delays*, or even *knock-on delays*.

e.g. through headway requirements. In a different situation, for example a schedule including a large amount of slack time, a source delay may result in no delayed event at all.
In periodic timetabling one has to consider both *periodic source delays*, e.g. a construction site slowing down all train trips affecting a certain driving activity in every period, and *aperiodic source delays*, e.g. a jammed door delaying a single train trip realizing this driving activity only in a single period.
In our model, periodic and aperiodic source delays can be incorporated by the same method, by redistributing the probability mass of an aperiodic source delay over all periods. Moreover, it suffices for our model to specify the boundary distributions for the (periodic and aperiodic) source delays on each driving arc instead of the joint distribution for all driving arcs, cf. Section 3.2.

2.3 The Delay Management problem

Delay management deals with (small) source delays of a railway system as they occur in the daily operational business of any public transportation company. The question is to decide if trains should wait for delayed feeder trains or if they better should depart on time (*wait-depart decisions*). From these decisions one obtains a *disposition timetable* which has to respect operational constraints, in particular the limited capacity of the track system. The difficulty of delay management comes with the following evident goal: make the disposition timetable as convenient as possible for the passengers.

If the transfer activities and the headway activities are neglected, the problem is easy and can be solved efficiently by the critical path method (CPM). If either the transfer activities or the headway activities are taken into account, the problem becomes NP-hard even in very simple networks and basic delay scenarios, see [8, 7, 2]. Neglecting only the headway activities, we obtain the (pure) delay management problem. An integer programming model is given in [29, 30]. These publications include an investigation of the special structure of the underlying event-activity networks. The model will be described in Section 3.4. The general integer programming model has been further refined in [11].
A first online-approach is provided in [9]. A bicriteria model for delay management in the context of max-plus-algebra has been presented in [25], a formulation as discrete time-cost tradeoff problem is given in [10]. How to react in case of delays has also been tackled by simulation and expert systems. We refer to [35, 36, 34, 37] for providing knowledge-based expert systems includ-

ing a simulation of wait-depart decisions with a *what-if* analysis. Real-world applications have been studied e.g. within the project *DisKon* supported by Deutsche Bahn (see [1]).

3 Integer programs

In this section we first explain the integer programming approach to the deterministic PESP. After that we give a detailed description how we incorporate stochasticity and a simplified delay management into this approach. This incorporation process has to find a balance between accurate modeling and the necessary feature not to increase the size of the PESP-IP in order to be able to construct plans on the real-world level.
The last two parts of this section deal with the techniques used when the periodic timetable is fixed. First we explain how the periodic timetable is rolled-out into a non-periodic plan based on which an instance of the delay management problem is created. Each such instance entails a concrete scenario of source delays. Finally we introduce one additional integer programming approach. This time it solves the delay management problem.

3.1 Computing Optimum Periodic Timetables

The most straightforward way to compute an optimum solution for an optimization instance of PESP is to solve the following mixed-integer linear program

$$\textbf{(PESP-IP-}\pi\textbf{)} \qquad \min f(\pi, p) = \sum_{a=(i,j)\in \underline{A}} \tilde{w}_a \cdot (\pi_j - \pi_i + T \cdot k_a - \ell_a)$$

such that

$$\begin{aligned}
\pi_j - \pi_i + T \cdot k_a &\leq u_a \quad \text{for all } a = (i,j) \in \underline{A} && (3)\\
\pi_j - \pi_i + T \cdot k_a &\geq \ell_a \quad \text{for all } a = (i,j) \in \underline{A} && (4)\\
\pi_i &\geq 0 \quad \text{for all } i \in \underline{V} && (5)\\
\pi_i &< T \quad \text{for all } i \in \underline{V} && (6)\\
k_a &\in \mathbb{Z} \quad \text{for all } a \in \underline{A}. && (7)
\end{aligned}$$

The constraints (3), (4), and (7) constitute a rephrasing of (2), and the constraints (5) and (6) scale the time vector π to the basic interval $[0, T)$. Observe that $(\pi, k) = (0, \frac{1}{T} \cdot \ell)$ is a trivial optimum solution of the LP relaxation of PESP-IP-π. Thus the linear relaxation of the PESP is of little use.

Although not pushing the LP optimum value beyond zero, an IP formulation which is equivalent to PESP-IP-π turns out to be much better suited for practical computations ([20]), e.g. using CPLEX. Instead of encoding the time information in a vector π which we define over the events, Nachtigall proposed to switch to time variables for the activities ([23]), $x_a = \pi_j - \pi_i + T \cdot k_a$, for $a = (i, j) \in \underline{A}$. In the context of electrical engineering, these new variables x are called the (periodic) tension induced by some node potential π. We refer to the resulting IP formulation as PESP-IP-x-z [16, 19].
One can improve PESP-IP-x-z by adding further valid inequalities. Such have been proposed by Odijk ([24]) and Nachtigall ([23]), and we use them throughout any of our PESP optimization runs.

3.2 Adding Delay-Resistance

The goal is to construct a periodic timetable minimizing a two-fold objective function. The first part is the nominal objective, i.e. the sum of weighted transfer times. The second part is the expected delay. The expected delay depends on the delay management and is an expected value with respect to the given joint distribution of the source delays. As this is too ambitious, we replace the second part by a simplified objective. A posteriori we compare the behavior of this simplified objective with the delay under a non-simplified, optimal delay management on a set of sampled scenarios. The simplifying assumptions for our periodic planning of the regular service are the following:

1. We assume a strict no-wait policy for delay management.
2. The published driving times are fixed parameters (not optimization variables).
3. We count the delay of passengers that miss a connection, but not the delay of the last train of a passenger's trip.
4. We approximate the resulting objective function by a convex, piecewise-linear function.

Strict No-Wait. We assume a strict no-wait policy: For the assumed disposition timetable every constraint can be broken in order to ensure that every departure event takes place as scheduled. Thus, under a strict no-wait policy source delays can only affect those train rides on which they occur. This approach is motivated by the (non-strict) no-wait policy for delay management, i.e. a delay management in which every train departs as early as *possible* (although not ahead of schedule). This feasible no-wait policy only breaks constraints A_{transfer}. Breaking a transfer is possible at a certain cost: The passengers can transfer to a later train, or receive a certain compensation. In contrast, the strict no-wait policy cannot be implemented in practice, because it also breaks the following practically indispensable constraints:

- The headway activities A_{head} model infrastructure requirements, e.g. that two trains using the same track must use it with a certain time difference.
- A train's trip consists of several driving and waiting arcs A_{drive} and A_{stop}, respectively. Precedence constraints among those activities are of course indispensable.
- One physical train usually covers more than one trip. However, it cannot start the second trip before it has finished the first one. This is modeled by the turnaround activities A_{turn}, together with a linear objective function.

Clearly, constraints expressing such conditions cannot be dropped in reality. At some arcs an all-wait policy and therefore a propagation of occasional delay is a matter of fact. Hence, adopting the strict no-wait policy, we underestimate the effects of source delays as we exclude delay propagation. More precisely, compare the strict no-wait policy $\mathcal{SN}$ to the (non-strict and thus practical) no-wait policy $\mathcal{PN}$. The expected passengers' delays ($\mathbb{E}[D(\cdot)]$) of these policies fulfill $\mathbb{E}[D(\mathcal{SN})] \leq \mathbb{E}[D(\mathcal{PN})]$. On the other hand, for an optimal delay management $\mathcal{OM}$ as used in the evaluation, we have $\mathbb{E}[D(\mathcal{OM})] \leq \mathbb{E}[D(\mathcal{PN})]$ because it is of course an option for the optimal policy $\mathcal{OM}$ to emulate the policy $\mathcal{PN}$. Whether the strict no-wait policy performs better than the optimal delay management because it unrealistically neglects necessary delay propagation or whether these effects are weaker than what an optimal delay management can win against a no-wait policy is not clear a priori.
Note that in some railway networks, the feasible no-wait policy is the only response that practitioners seem to have to the 'repair game' [4]:

> The management of the region south-west (of Deutsche Bahn AG) decided to apply a (non-strict) no-wait policy from March 24 on.
> Die Regionalleitung (Südwest) hat beschlossen, dass wir bereits ab 24. März die Wartezeitvorschrift auf ‚Keine Wartezeit' abändern.
> (Udo Wagner, Vorsitzender der Regionalleitung DB Regio Südwest, in BahnZeit Mai 2004, employee newsletter of Deutsche Bahn AG)

Fixed Driving Times. In principle, our approach allows for the departure times and the arrival times to be at the disposal of the optimizer as it has also been done in [14]. However, in this study we refrain from using this optimization potential in setting the arrival of a train ride independent from its departure. We assume fixed driving times for the regular service timetable. Thus the published arrival time of a trip will be determined by its published departure time.

For performance reasons, when constructing the delay resistant timetables, we fix the nominal driving time of each train (in accordance with Leaflet 451-1 of UIC (Union Internationale de Chemins de Fer)) to 107% of the technically minimal driving time. When we construct a disposition timetable with optimal delay management in the testing, we allow trains to use 76.4% of these driving time supplements in order to catch up with delay. For example, if under ideal conditions the technically minimum driving time is 100 min, in the regular service schedule we plan a driving time of 107 min. In the disposition timetable, a driving time of $0.95 \cdot 107 = 101.65$ min can be planned to catch up at most 5 min 21 sec of delay, i.e. 76.4% of the added supplement which was 7 min.

Explanatory Graph Expansion. Assuming a discrete scenario set, the delay resistant timetabling problem with strict no-wait policy can be expressed as a PESP graph with size only by a constant factor larger than the deterministic problems' PESP graph. This graph is attained by a local scenario expansion as shown in Figure 3.

In the deterministic model, a pair of departure events of train rides between which passengers transfer is linked by a path of length 2. The middle node represents the arrival of the first train at the station from which the second train departs. In the local scenario expansion, this pair of departure events is linked by a set of parallel paths of length 2. Each of the paths represents a scenario.

In each scenario the first arc has a different but fixed length, namely the scenario's driving time on that track. The optimization process determines the time difference between the *departure* events of the two trains. Thereby, the lengths of the second arcs, i.e. the transfer times in the different scenarios, are set simultaneously for all scenarios.

In the new objective function the total weight of the deterministic transfer arc in the deterministic objective function is distributed to the transfer arcs of the scenarios relative to probabilities of the scenarios.

The optimization will seek to keep the transfer times short. In an aperiodic model, feasibility would imply that the schedule is dictated by the longest

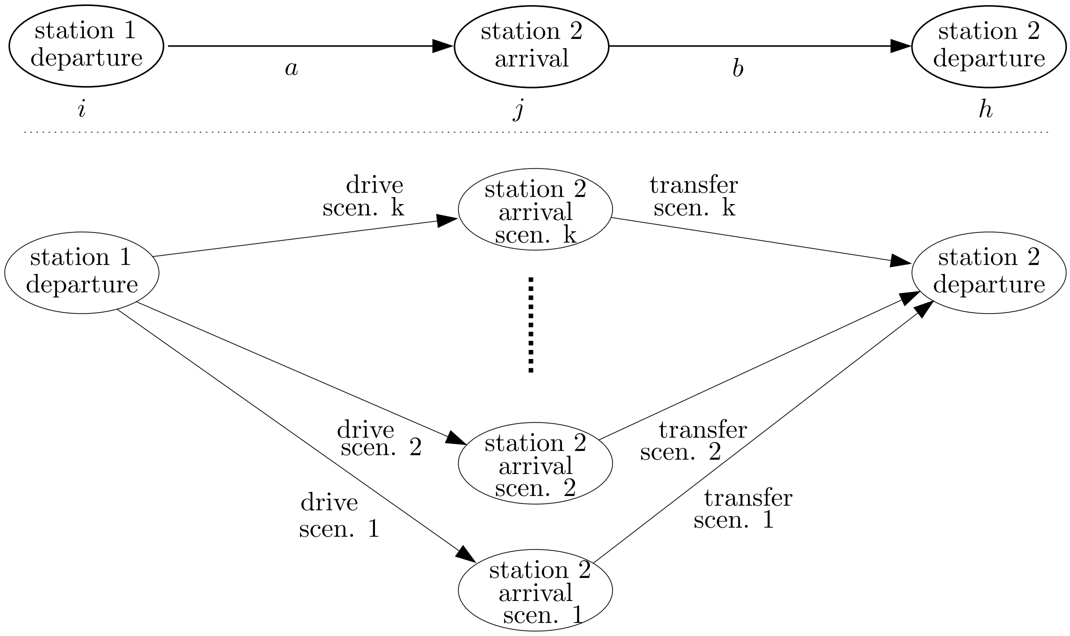

Figure 3: Expansion of a driving and waiting activity to k different scenarios.

driving time. In the PESP model, this is not the case. A path representing a scenario with low probability but long driving time might be quasi neglected in order to obtain a schedule that gives short connections for the likely scenarios. This negligence does not make the plan infeasible. The inequalities of transfer arcs of long but unlikely scenarios are nevertheless fulfilled modulo the period time T. In practice this means that those passengers connect to the train of the next period. The waiting time for this is expressed correctly by the PESP on the expanded network graph. The path of a neglected scenario incurs a high cost which is weighted with probability of the scenario in the objective function.

In periodic timetabling, surprisingly a strict no-wait policy turns out to be adequate for transfer arcs. Here, the connections are not broken but only established with a later period for which the expected waiting time can be expressed correctly by a PESP graph only a constant bigger in size than the deterministic problems' PESP graph. This technique which we exemplified for the transfer arcs can also be applied to other arcs expressing breakable constraints that enter the objective function.

Stochastic Dependency. Because of the strict no-wait policy, the effects of uncertainty remain local. Therefore it suffices to know the boundary dis-

tributions for each arc's driving time. Knowing the joint distribution of the driving times is not necessary. By the same argument, it is immaterial in this setting whether a source delay is periodic or aperiodic.
This holds for fixed passenger weights on transfer arcs which is the model we use here and for which test data is available to us. Alternatively, one can replace the fixed passenger weights by passenger paths. In the latter case, a passenger travels along the (aperiodic) realizations of a fixed path of (periodic) arcs. The passengers start their itinerary distributed equally over all periods. They transfer to the earliest realization of the connecting driving arc which is reachable. Thereby, a periodic driving arc does not have a passenger weight fixed for all periods but only a fixed average passenger weight over all periods. The realization of a driving arc in a period in which additional passengers from a delayed transfer use that driving activity can have more passengers than that driving arc has on average. This model is also covered by our cost model as long as the probability of missing any connection a is independent of that for missing any other connection b. This assumption seems quite natural in practice.

Punishing Delay. The weighted sum of transfer times in the expanded PESP graph is an exact measure for the expected travel time of all passengers. A more detailed description of this objective function, the graph expansion and its non-discretized version can be found in [21].
However, the expected travel time does not reflect how much a passenger's travel time deviates from what passengers planned. In general a minute of expected delay weighs heavier than a minute prolongation of nominal travel time. Therefore, we use a refined objective function which punishes delays, i.e. the deviation of the actual travel time from the travel time published in the timetable.
To this end we define a *delay-weighting factor* s and introduce a reference to the nominal schedule in the objective function. The contribution c_b of a transfer arc $b = (j, h)$ in the deterministic PESP graph to the objective for a timetable π is then defined as

$$\begin{aligned} c_b(\pi) \; &:= \; w_b \Bigg([\pi_h - \ell_b - (\pi_i + t_a)]_{\bmod T} \\ &\quad - s\, T \sum_r p_a(r) \left\lfloor \frac{[\pi_h - \ell_b - (\pi_i + t_a)]_{\bmod T} - (\ell_a^r - t_a)}{T} \right\rfloor \Bigg) . \end{aligned}$$

Here, we use t_a for the nominal driving time on arc $a = (i, j)$ leading to the transfer $b = (j, h)$. The minimum transfer time for b is given by the lower bound ℓ_b. Finally, in scenario r which occurs with (marginal) probability

$p_a(r)$, the minimum driving time is ℓ_a^r, i.e. the difference $(\ell_a^r - t_a)$ gives the prolongation of the driving time in scenario r. The rounded down term becomes negative when the prolongation in the scenario exceeds the planned buffer. A delay punishment of $w_b \cdot p_a(r) \cdot s \cdot T$ is then incurred in the objective function. Note that operating a periodic timetable, the total delay for a passenger who missed a connection always equals a full period time T, no matter how much of this extra time is spent in the delayed feeder train or waiting for the connection.
If the prolongation is less than the planned buffer, it causes no punishment. In practice the passengers experience a prolongated driving time and a reduced transfer time, but eventually reach their *planned* connection. Note that the above definition gives a correct model only for disturbances that are small with respect to T.

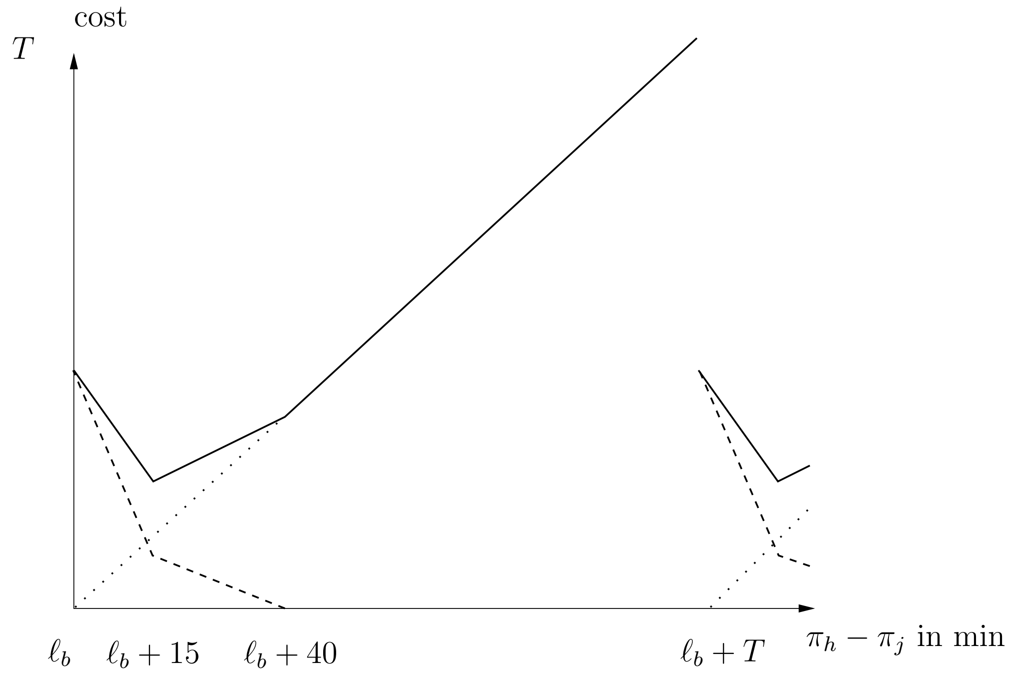

Figure 4: The nominal cost (dotted), the delay punishment (dashed), and the total cost function (solid) of a transfer arc b.

The function c_b need not be convex in the planned transfer time $\pi_h - \pi_j$ on arc b. However, according to practitioners, real-world distributions give almost convex functions. Given that c_b is convex, we perform a further approximation by piecewise-linearizing the function. Such objective functions can easily be integrated in the mixed integer program with a small increase in size compared to the original, deterministic PESP problem. In particular,

this can be done without introducing new integer variables. The technique is described in detail for a similar setting in [21].
Hence, we split our function c_b into two intervals with negative slope and one interval with slope equal to w_b. The latter is the time interval when in all scenarios all trains that instantiate the driving arc $a = (i, j)$ of the feeder train have reached the transfer station.
In the study we consider three different distributions for the driving times, see Figure 7 in Section 4. For the delay-weighting factor s we use three different values, $1.5, 2$, and 5. Optimizing with a delay weighting factor 2 means that one minute of expected delay causes the same cost as two minutes of prolongation in the planned travel time. Figure 4 shows the costs depending on the planned transfer time for an arc with unit passenger weight and minimum transfer time, distribution C (cf. Figure 7) and delay weighting factor 2.

Light Robustness. By the objective function we have chosen, our approach is an exact stochastic programming model for the delay due to missed connections under a strict no-wait policy—granted the convex, piecewise-linear approximation.
The strict no-wait policy provides for the crucial simplification. We will empirically show that this simplification still allows us to capture the behavior of an optimal delay management with sufficient accuracy.
The approach can also be understood as an extension to the concept of Light Robustness [6]. Light Robustness searches among all almost nominally optimal solutions for those which leave the most local slack in the constraints. The key problem of Light Robustness is to find a function of the slack variables of the constraint system that favors solutions with well distributed slack instead of those with high slack in a few constraints. Our objective defines such a function in a natural way, namely by counting the expected number of missed connections.

3.3 Creating an Instance of the Delay Management Problem

Using the methods of Sections 3.1 and 3.2, we obtain a periodic timetable π that assigns a periodic time π_i to each event $i \in \underline{V}$. For the delay management problem, we need a representation of the network with a granulation such that we can distinguish between two physical trains which operate on the same line. This means we have to *roll out* the periodic event-activity network. To this end, we additionally need

- the (aperiodic) times $\pi_{\text{first}}(i)$ and $\pi_{\text{last}}(i)$ of the first and last occurrence of each periodic event $i \in \underline{V}$, and
- lower and upper bounds $\ell_a \leq u_a$ for all $a \in \underline{A}$.

Note that for each activity $a = (i, j) \in \underline{A}_{\text{drive}} \cup \underline{A}_{\text{stop}} \cup \underline{A}_{\text{turn}} \cup \underline{A}_{\text{transfer}}$, we know that

$$u_a - \ell_a < T. \tag{8}$$

For the computation of $\pi_{\text{first}}(i)$ and $\pi_{\text{last}}(i)$, we need to fix a time interval $[t_{\min}, t_{\max}]$ with length H that describes our observation period. Then, for each $i \in \underline{V}$, we set

$$\begin{aligned} \pi_{\text{first}}(i) &= \min \{\pi_i + kT : \pi_i + kT \geq t_{\min}, k \in \mathbb{Z}\} \\ \pi_{\text{last}}(i) &= \max \{\pi_i + kT : \pi_i + kT \leq t_{\max}, k \in \mathbb{Z}\}. \end{aligned}$$

Our goal is to expand the periodic network $\underline{D} = (\underline{V}, \underline{A})$ into a non-periodic network $D = (V, A)$, cf. Section 2.1. As input for the delay management problem, we need the scheduled time $\pi(i_k)$ for all $i_k \in V$ and the lower bounds ℓ_a for all non-periodic activities $a \in A$. Since in the case of delays, all drive and turnaround activities a are performed faster (or at the same speed) than in the original timetable π, the upper bounds u_a used in the planning phase in the constraints $\pi_j - \pi_i \leq u_a$ are of no importance for these activities. For stop activities, however, we may voluntarily override certain upper bounds in order to ensure particularly sensitive transfers.

To obtain V, A, $\pi(i)$ for all $i \in V$ and ℓ_a for all $a \in A$, we proceed as follows:

- **for each** periodic event $i \in \underline{V}$ and for each k with $1 \leq k \leq 1 + \left\lfloor \frac{\pi_{\text{last}}(i) - \pi_{\text{first}}(i)}{T} \right\rfloor$, we create a new non-periodic event i_k with time $\pi(i_k) = \pi_{\text{first}}(i) + kT$
- **for each** periodic activity $a = (i, j) \in \underline{A}$
  - **if** $a = (i, j) \in \underline{A}_{\text{stop}} \cup \underline{A}_{\text{drive}} \cup \underline{A}_{\text{transfer}} \cup \underline{A}_{\text{turn}}$ **then** do:
    for each non-periodic event $i_s \in V(i)$ look for a non-periodic event $j_t \in V(j)$ satisfying $\pi(j_t) \in [\pi(i_s) + \ell_a, \pi(i_s) + u_a]$. If such an event j_t exists, it is unique due to (8) and we create a unique new non-periodic activity $a_{st} = (i_s, j_t)$ with $\ell_{a_{st}} = \ell_a$ (if $a \in \underline{A}_{\text{drive}}$, we set $\ell_{a_{st}} = 0.95 \cdot \ell_a$ to allow to catch up delays, see page 13).
  - **if** $a = (i, j) \in \underline{A}_{\text{head}}$ **then** do:
    for each non-periodic event $i_s \in V(i)$ and for each non-periodic event $i_t \in V(j)$ create two new non-periodic disjunctive activities $a_{st} = (i_s, i_t)$ and $a_{ts} = (i_t, i_s)$, define $\ell_{a_{st}} = \ell_a$ and $\ell_{a_{ts}} = T - u_a$.

The formulas for the lower bounds on the headway activities in the rolled-out network are based on the following: Consider a periodic headway activity $a = (i, j) \in \underline{A}_{\text{head}}$, linking two periodic events $i, j \in \underline{V}$. Let w.l.o.g. the periodic time π_i of event i be smaller than π_j. Rolling out these two events means that we first obtain an element of $V(i)$, then an element of $V(j)$, both in the first period. The latter is followed by the next element of $V(i)$ in the second period and so on, i.e. we obtain a sequence

$$\pi_i \leq \pi_j \leq \pi_i + T \leq \pi_j + T \leq \pi_i + 2T \leq \dots$$

of occurrences of events i and j. Let $i_s \in V(i)$ and $i_t \in V(j)$. If both events take place within the same period, we obtain $\pi(i_t) - \pi(i_s) = \pi_j - \pi_i \geq \ell_a$ to ensure the headway distance. If, however, $\pi(i_s) = \pi_i + T > \pi_j = \pi(i_t)$, we have $\pi(i_t) - \pi(i_s) = \pi_j - \pi_i - T$. Adding T to both sides yields $\pi(i_t) - \pi(i_s) + T = \pi_j - \pi_i \leq u_a$, i.e. $\pi(i_s) - \pi(i_t) \geq T - u_a$.

3.4 Solving the Delay Management Problem

In case of disturbances, the disposition timetable π as constructed in Section 3.3 has to be updated to a disposition timetable $\tilde{\pi}$. To evaluate the performance of the planned timetable under delays, we construct a set of delay scenarios as follows. Each scenario is defined by a set of *source delays*. These are modeled as a set of activities with a positive delay $d_a > 0$ indicating the additional length of the activity which can be periodic or non-periodic, see Section 2.2. For non-delayed activities, we set $d_a = 0$. The model also allows source delays on events which model for example a train driver appearing too late to his shift or a track that is occupied until a fixed point of time. However, we do not use such source delays on events in this case study.

Apart from the source delays d_a, we need the data generated in Section 3.3 as input for the delay management problem, i.e. the expanded network together with the lower bounds ℓ_a, the common period T of all events and the number of passengers w_a planning to use connection $a \in A_{\text{transfer}}$. Additionally, we include parameters w_i that represent the number of the passengers for which event $i \in V$ is their final destination.

Integer programming formulation. To model the delay management problem, we need the following three types of variables: First, for all events $i \in V$, we need

$$\tilde{\pi}_i = \text{actual time of aperiodic event } i \text{ in the disposition schedule.}$$

Note that the delay of event i is hence given by $\tilde{\pi}_i - \pi_i$ and that we have to require that $\tilde{\pi}_i \geq \pi_i$ holds: Otherwise, for departure events, a passenger who arrives on time may miss his train, and for arrival events, we cannot guarantee that the station already has sufficient platform capacity for the arriving train. For each transfer activity $a \in A_{\text{transfer}}$, we introduce

$$z_a = \begin{cases} 0 & \text{if transfer activity } a \text{ is maintained} \\ 1 & \text{otherwise,} \end{cases}$$

and for $a = (i, j) \in A_{\text{head}}$, we use

$$g_a = \begin{cases} 0 & \text{if } i \text{ starts before } j \\ 1 & \text{otherwise.} \end{cases}$$

The following is an integer programming formulation of the delay management problem (see [31, 27]).

$$(\mathbf{DM}) \qquad \min f(\tilde{\pi}, z) = \sum_{i \in V} w_i(\tilde{\pi}_i - \pi_i) + \sum_{a \in A_{\text{transfer}}} w_a T z_a$$

such that

$$\begin{aligned}
\tilde{\pi}_i &\geq \pi_i \quad \text{for all } i \in V & (9)\\
\tilde{\pi}_j - \tilde{\pi}_i &\geq \ell_a + d_a \quad \text{for all } a = (i, j) \in A_{\text{stop}} \cup A_{\text{drive}} \cup A_{\text{turn}} & (10)\\
M z_a + \tilde{\pi}_j - \tilde{\pi}_i &\geq \ell_a \quad \text{for all } a = (i, j) \in A_{\text{transfer}} & (11)\\
M g_a + \tilde{\pi}_j - \tilde{\pi}_i &\geq \ell_a \quad \text{for all } a = (i, j) \in A_{\text{head}} & (12)\\
g_a + g_{a^{-1}} &= 1 \quad \text{for all } a, a^{-1} \in A_{\text{head}} & (13)\\
\tilde{\pi}_i &\in \mathbb{N} \quad \text{for all } i \in V\\
z_a &\in \{0, 1\} \quad \text{for all } a \in A_{\text{transfer}}\\
g_a &\in \{0, 1\} \quad \text{for all } a \in A_{\text{head}}.
\end{aligned}$$

Note that w_i are positive weights that represent the importance of event i. The first constraint (9) makes sure that no train departs earlier than scheduled, while (10) ensures that the delay is carried over correctly from one event to the next. Constraints (10) additionally take the source delays d_a into account. In particular, if event i takes place at some time point $\tilde{\pi}_i$, event j must be later than $\tilde{\pi}_i + \ell_a$ where $a = (i, j)$ is the activity linking i and j. If $z_a = 0$, constraint (11) ensures that the delay is carried over for this connection, i.e. this connection is maintained. For $z_a = 1$, however, constraint (11) becomes redundant whenever M is large enough (for the size of M, see [27]). For our

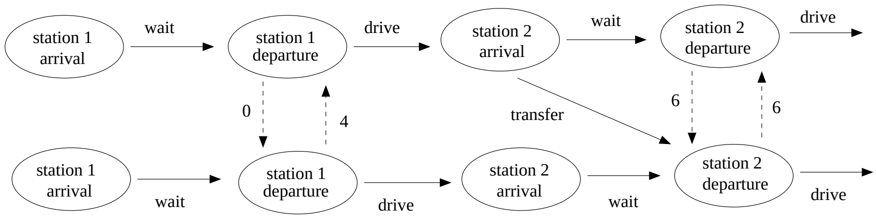

Figure 5: Graphical interpretation of the capacity constraints

computations, we used $M = H + 2T$ which was large enough. Constraints (9) to (11) describe the *pure* delay management problem. However, to get realistic results we need to consider also restrictions of the type A_{head} as done in (12) and (13). These constraints can be equivalently reformulated to

$$(12) \text{ and } (13) \iff \text{ either } \tilde{\pi}_j - \tilde{\pi}_i \geq \ell_a \text{ or } \tilde{\pi}_i - \tilde{\pi}_j \geq \ell_{a^{-1}}.$$

Hence, (12) and (13) require for all $(i, j) \in A_{\text{head}}$ that either $\tilde{\pi}_i \geq \ell_{a^{-1}} + \tilde{\pi}_j$ or $\tilde{\pi}_j \geq \ell_a + \tilde{\pi}_i$. Figure 5 shows the graphical interpretation of the headway constraints: While the solid activities are already fixed, the goal is to choose exactly one of each pair of dashed edges. This fixes the order of the events and at the same time ensures the safety distances, given as lower bounds ℓ_a on the edges $a = (i, j)$. Recall that a feasible disposition timetable exists if and only if the activities representing the precedence constraints are cycle-free. Hence, one has to choose one edge from each pair of dotted edges in such a way that the resulting network does not contain any directed cycle.

Objective function. The above formulation minimizes a combination of weighted dropped connections and weighted train delays. The weight of a dropped connection $a \in A_{\text{transfer}}$ is set to the time period T since this is the delay a passenger will suffer when missing a train. Although the formulation does not minimize the sum of additional delays over all passengers in general, it does so in a large class of delay management problems, namely whenever the *never-meet property* is satisfied. The never-meet property is fulfilled if in no feasible time-minimal solution of the delay management problem, the paths of two delayed customers will meet and if source delays only occur after non-delayed events (for a formal definition and its satisfaction in practice, see [32, 27]).
In our case, the objective used in (DM) is an approximation of the sum of all passengers' delays. It can also be seen as a weighted scalarization of the two objectives *minimize (weighted) number of dropped connections* and *minimize*

number of (weighted) train delays in minutes which are defined in bicriteria delay management problems [10, 12].

Solution approach. In our case study, we were fortunately able to solve the problem optimally by Xpress-Mosel 1.6.3 (2006b) within less than one hour of computation time. We are aware of the fact that these good results are achieved because we limited our calculations to an observation period of some hours and since within the region that we investigate, there are only few conflicts with the never-meet property in most of our delay scenarios. Otherwise, heuristics may be used, see [27].

4 The Data of the Case Study

We apply our approach to the Harz Region of the Lower Saxony (Niedersachsen) part of the German Railway network, see Figure 6.

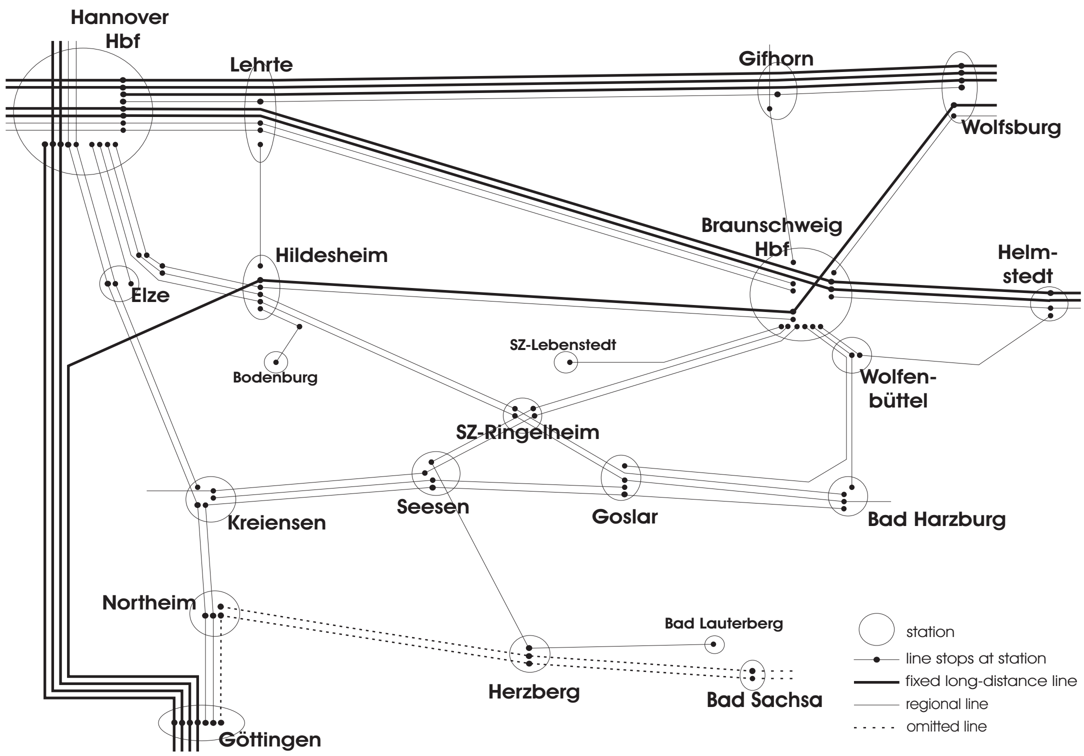

Figure 6: The railway lines in the Harz region of the Lower Saxony part of the German railway network.

In our case study, we consider the entire set of passenger railway lines that are operated within this region. Only this way we may expect to obtain significant results. Otherwise delay propagation which is caused by limited capacity of the infrastructure would not be an issue. The only relaxation that we introduce is that we assume all the lines to be operated with a period time of $T = 120$ minutes. In reality, there are a few lines which are operated hourly.
In the step of computing the periodic regular service timetable, we assume the timetables of the long-distance lines to be fixed to a case that was kindly provided to us by Deutsche Bahn AG. Thereby we ensure that the optimized timetables that we are about to compute for the regional service lines can be embedded into a realistic case for Germany as a whole.

In total, our case features 30 pairs of directed railway lines, including 9 pairs of long-distance lines with fixed timetables. For most of the tracks, the minimum headway that any timetable has to respect is 3 minutes. Furthermore, for more than 30 single track segments, we will ensure safe operations. Interestingly, the long-distance ICE line Berlin–Wolfsburg–Göttingen–Frankfurt–Basel/München has *three* of these single tracks along its route, and there are further regional service lines that operate on the very same single tracks. Among the timetables that respect all these infrastructural requirements, we mainly head for short transfer times—both nominal and during operations—along the 182 most important transfers within the region. More precisely, we have assigned weights $\tilde{w}$ to the corresponding transfer activities $\underline{A}_{\text{transfer}}$. The weights represent the number of passengers who use the transfers and were estimated by Deutsche Bahn AG using their traffic assignment model.

In addition, note that when computing the periodic regular service timetable we keep the driving times of the trains unchanged for performance reasons. In particular, these include the time supplements as they are suggested by the UIC, cf. Section 3.2. Yet, for 26 stop activities $a \in \underline{A}_{\text{stop}}$, we allow their minimum dwell time ℓ_a to be extended by a few minutes. This enables better synchronization at single tracks and at transfers. Moreover, we have to pursue two further important goals. First, where two lines with $T = 120$ minutes share the same tracks over a long distance, we require a balanced hourly service, e.g. between Braunschweig and Seesen. Second, in order to compute realistic timetables, we must not neglect operating costs, i.e. the number of trains that are required to operate the timetable, see [18, 16] for any details.

Graph data. In Table 1 we report the size of the resulting periodic network $\underline{D} = (\underline{V}, \underline{A})$. Note that for the purpose of periodic timetabling, many

Table 1: The sizes of the networks $\underline{D} = (\underline{V}, \underline{A})$, its contracted version, and the non-periodic network $D = (V, A)$

quantity			$\underline{D}$	contracted version of $\underline{D}$	D
$\lvert\underline{V}\rvert$	or	V	4721	65	≈ 15000
$\lvert\underline{A}\rvert$	or	A	5469	517	≈ 26500
$\lvert\underline{A}_{\text{transfer}}\rvert$	or	A_{transfer}	182	–	≈ 500
$\lvert\underline{A}_{\text{head}}\rvert$	or	A_{head}	454	–	≈ 11000

redundancies can be removed from this network. Hence, we also provide the corresponding values for the network that we obtain after having contracted many of the arcs, see [16] for details. Unfortunately, in the contracted network the information becomes highly aggregated and thus is no more suited to immediately derive the non-periodic network for the delay management problem. Therefore we roll out the non-periodic network $D = (V, A)$ as three copies of the uncontracted periodic network $\underline{D} = (\underline{V}, \underline{A})$ (as described in Section 3.3) and give its size in Table 1.

In the integer programming formulation (DM) of the delay management problem, we need to assign weights to all events $i \in V$ and to all activities $a \in A_{\text{transfer}}$. As the only information we have on the number of passengers are the weights $\tilde{w}_a$ of the transfer activities $a \in A_{\text{transfer}}$ derived from the periodic timetable, we set $w_i = 1$ for all $i \in V$. In order not to overestimate the importance of missed connections (compared to delayed arrival events), we set $w_a = \frac{\tilde{w}_a}{\overline{w}}$ for all $a \in A_{\text{transfer}}$ where $\overline{w}$ denotes the arithmetic mean of the weights $\tilde{w}_a$ of all transfer activities.

Delay Distributions. For delay resistant timetabling we consider three distributions of source delays, described in Table 2 and shown in Figure 7. Here, P_{ontime} denotes the probability that a train for a certain driving arc needs at most 107% of the corresponding minimal driving time. According to the UIC rule, 107% can be understood as the uniform supplement used so far. The time $t_{\max}$ is the maximum time in minutes a train exceeds this 107% driving time. Finally, z is some smaller time such that with probability $P(\leq z)$ the trains take at most z minutes longer than 107% of their minimal driving time. The table gives the values for type A, B, and C distributions. Together with the delay-weighting factors s this specifies the settings under which the ten timetables are optimized. DEF is the ID of the nominal or default plan that takes no delay into account.

In the figure the dashed line represents the type C distributions, the straight line type A. Type C is stochastically greater than A, and B is incomparable to both. It represents settings with many but small source delays. In type A and C we assume 80% of the trains to run on time (in the sense that their trip takes at most 107% of the minimal technical driving time). For type B this probability is lowered to 75%.

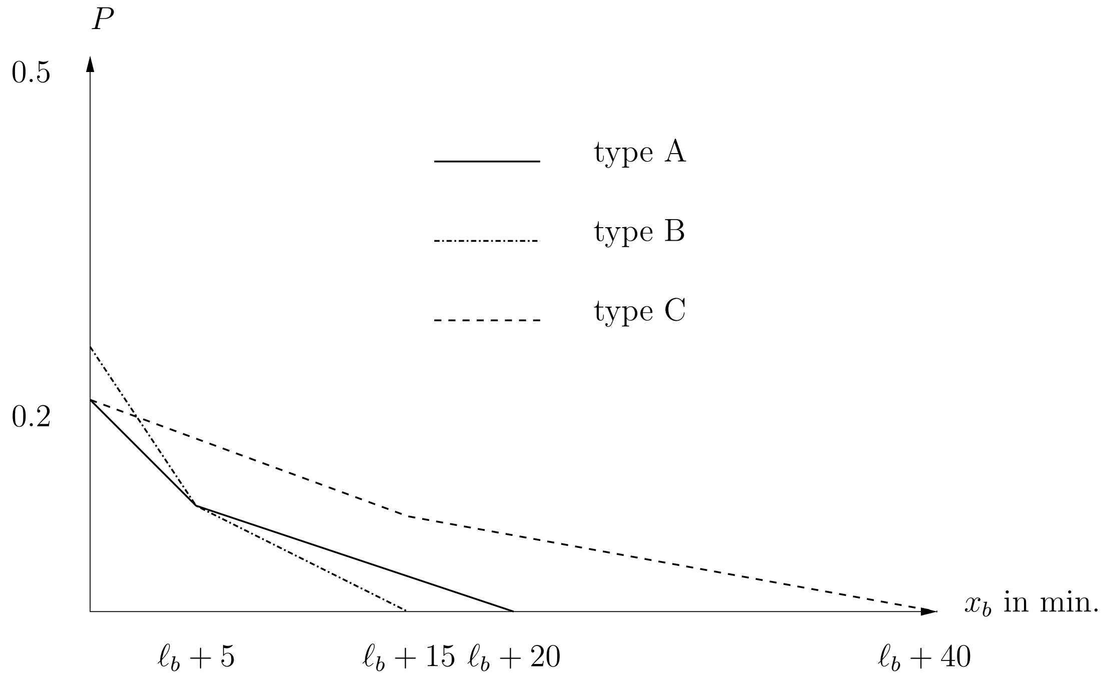

Figure 7: The three boundary distributions for the probability to miss a connection scheduled with tension x_b on transfer arc b.

To be able to solve the delay management problem optimally, we need to limit our calculations to an observation period H of some hours. Thus we consider all trains which arrive at or depart from one of five central stations within an interval of six hours. These five stations are chosen in such a way that *all* lines contain at least one of these stations – we cover the line paths with vertices. We then reduce the non-periodic network (V, A) that has been obtained by applying the method described in Section 3.3 as follows: First, we delete all events which do not belong to a trip arriving at or departing from one of the five central stations within the observation period. Then we delete all "dangling" activities, i.e. all activities which start or end with a deleted event.

The delays we use to compare different timetables are generated as follows: For each rolled-out period, we choose 24 different driving or stopping activities at random. Among these, 12 are delayed by a randomly chosen time between 60 and 300 seconds, while the other 12 activities are delayed by

Table 2: Distributions and delay weighting factors.

Id	P_{ontime}	z	$P(\leq z)$	$t_{\max}$	s
DEF	1	–	–	–	0
A1.5	0.8	5	0.9	20	1.5
B1.5	0.75	5	0.9	15	1.5
C1.5	0.8	15	0.95	40	1.5
A2	0.8	5	0.9	20	2
B2	0.75	5	0.9	15	2
C2	0.8	15	0.95	40	2
A5	0.8	5	0.9	20	5
B5	0.75	5	0.9	15	5
C5	0.8	15	0.95	40	5

between 360 and 1200 seconds (also randomly chosen). Hence we have a total of 72 source delayed activities during our 6-hours time slot, the sum of all delays lies between 15120 and 54000 seconds. This choice corresponds to an average scenario for a distribution fulfilling the boundary distributions specified as distribution type A.

5 Results

We constructed delay resistant periodic timetables for three different distributions and three different delay-weighting factors s for the Harz subnetwork of Deutsche Bahn AG. We also computed a nominally optimal timetable. Then for each of these ten plans we computed optimal disposition timetables under 68 random delay scenarios. The methods we applied in both steps proved to be capable of solving both the delay resistant timetabling and the delay management problem on the real-world level. The optimization of the timetables has been obtained by solving PESP-IP-x-z, with refined objective function, by CPLEX 10.1 on a 3 GHz PC.
Each timetable has been optimized with respect to a different objective function, resulting from their underlying distribution and the weighting factor. For a better comparison we also calculated the value that each timetable attains under the nine objective functions of the other plans. For a timetable ID let C_{ID} denote the function with respect to which ID has been optimized. Thus C_{DEF} is the nominal cost, and any other cost function C_{ID} can be written as $C_{\text{ID}} = C_{\text{DEF}} + C'_{\text{ID}}$ where C'_{ID} is the expected delay cost incurred by the distribution and the weighting factor of ID—in other words, the delay penalty.

For each plan ID we calculate the *Price of Robustness*

$$\mathrm{PoR}(\mathrm{ID}) := C_{\mathrm{DEF}}(\mathrm{ID})/C_{\mathrm{DEF}}(\mathrm{DEF})$$

which is the nominal cost of the plan ID in relation to the minimal nominal cost of any plan. This serves as a measure for how much nominal passenger traveling time is spent to achieve delay resistance in the sense of ID's delay penalty.
Like the PoR is a measure of how much is sacrificed, the *Ratio of Delay* measures the gain. It is defined by

$$\mathrm{RoD}(ID) = C'_{\mathrm{ID}}(\mathrm{DEF})/C'_{\mathrm{ID}}(\mathrm{ID}) \; .$$

It measures for a given setting how much delay penalty the nominal plan incurs in comparison to a plan optimized to that setting of distribution and weighting factor.

Table 3: Results and performance of the optimization for our regular timetables.

ID	PoR *100	RoD *100	Delay Penalty $C'_{1.5C}$	CPU Time in sec.	miss opt	miss no-w pol
DEF	100	100	100	4802	100	100 (218)
15A	101	132	85	4843	84	80 (174)
15B	102	162	80	9094	79	79 (171)
15C	101	121	83	7960	81	81 (176)
2A	102	137	82	9011	80	80 (174)
2B	103	195	73	6908	67	71 (156)
2C	102	124	81	18327	79	79 (171)
5A	107	220	60	44275	53	53 (115)
5B	106	253	62	83356	55	55 (121)
5C	111	205	49	53743	49	48 (105)

In Table 3 we show these results. To compare the different delay resistant plans, we also show which delay penalties each plan incurs in the objective function of 1.5C, i.e. we give $C'_{1.5C}(ID)$ which is up to a constant factor equal to $C'_{2C}(ID)$ and $C'_{5C}(ID)$. Observe that the plans 1.5C and 5C achieve minimum respectively maximum PoR among all delay resistant timetables. Furthermore, the table gives the CPU time in seconds until optimality was established. The last two columns give the number of passengers who missed their connection in the simulation, i.e. of those who experience a delay of two hours. These numbers are averaged over the 68[3] random scenarios. The

[3]The choice of the number of scenarios is completely arbitrary.

last column gives the equivalent value for the feasible no-wait policy, whereas the previous column refers to the optimal delay management. Except for the CPU time all columns are normalized such that the nominally optimal plan gets a value equal to 100. To compare no-wait policy and optimal delay management directly we give in brackets the number of missed connections under no-wait policy in percent of the number of missed connections of DEF under optimal delay management. For example, the plan optimized for distribution C and delay-weighting factor 5 causes 5% more missed connections under no-wait policy than the nominally optimal plan under optimal delay management.

Reducing Delays. For the quantitative results one should look at the average of missed connections in the simulation. In this value—at the price of a relatively small increase in the nominal objective value of the timetables (PoR)—a substantial reduction of the probability to miss a train can be achieved. For instance, consider timetable 5A. Compared to the optimum nominal timetable, it increases the nominal objective by only 7%. But this almost halves (53%) the number of passengers who miss a connection. Thus, our first goal, namely to substantially reduce delays without substantially increasing nominal traveling time, is achieved for real-world instances.

Quality of Simplified Objective. We use a simplified objective function in the optimization. From a priori arguments we expect it to estimate well the costs of optimal delay management. The tests support that the simplified objective function gives a good estimate of the optimal delay management.

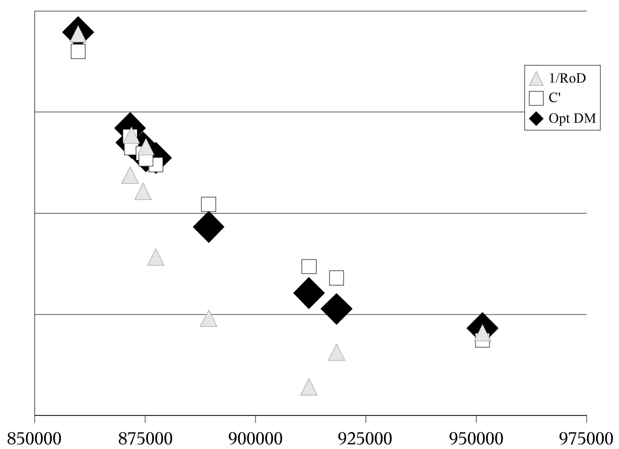

Figure 8: Simplified objective matches optimal delay management.

In fact, the delay penalty used in our simplified objective function provides a surprisingly adequate estimate of the actual delay penalty. This becomes apparent in Figure 8 which for the ten plans compares the simplified delay penalty $C'_{1.5C}$ (multiplied with a constant) on the one hand and the number of missed connections under optimal delay management on the other hand. The ten plans are ordered in the figure according to their nominal objective value. It is apparent how C' mimics the cost in optimal delay management. For comparison we display the inverse of the RoD, too. This result encouragingly justifies to use our simplified objective function for delay resistant timetabling.

Long Source Delays. Finally, a closer look at the results yields a general claim on delay resistant timetabling: Delay resistant timetabling is an appropriate measure to hedge against *small* source delays. To this end observe the following: Distribution C, which is the one with many long source delays, for small weighting factors (1.5 and 2) has a non-greater PoR than the other two distributions. This is surprising in two respects. First, the distribution A is stochastically smaller than C. Therefore contrary to the results one would expect the optimizer to invest less in delay resistance for a timetable tailored for A than for C. Second, the unexpected behavior vanishes when the weighting factor is lifted to 5. Then the optimizer takes into account the higher total expected source delay and invests a higher price for the robustness of plan 5C than for 5A or 5B.
This behavior gives rise to the following interpretation: Protecting the schedule against long source delays requires to intervene so strongly that it only pays if delay is weighted very high. This interpretation is supported by another detail. Plan 15C does not even have the lowest delay penalty in its own setting. Yet, it has the lowest nominal value among all delay resistant plans. Therefore, *in sum* it is the best plan for its setting. In contrast consider the plan 5C which is also the best plan for its setting, but for different reasons. It has the worst nominal value of all but the lowest delay penalty in its own (and, in fact, in every other) setting. To conclude, for a small weighting factor hedging against long source delays would be inappropriate. Only for high weighting factors it pays to give up nominal optimality to curtail the effect of long source delays.

6 Conclusion

In our study we successfully added delay resistance to the computation of periodic railway timetables. We estimated the delay resistance of a timetable by simulating delays and solving the corresponding delay management problem. The computational study that we effected on a part of the German railway network of Deutsche Bahn AG suggests that a significant decrease of passenger delays could be obtained at a relatively small price of robustness. This does not only apply to the nominal increase of passenger waiting times, but also to the computation times. Our concept can be interpreted as a slight extension of the "Light Robustness" approach but without adding integer variables to the well-established integer programs for periodic timetabling. Further research should aim at bringing the computation times of the refined model even closer to the standard model.

Going a step further, we aim at roughly approximating the complicated delay management objective function by adding a term that approximates the delay resistance to the standard periodic timetabling objective function. Two major properties of the resulting objective function are required: (i) being easy to evaluate; (ii) provide a cost value for each nominal timetable that is similar to the actual costs of the timetable under small disturbances. From the experience of this study we recommend the following workflow for achieving this goal:

First, compute several distinct timetables which balance differently the interests of nominal transfer time and the risk of missed connections. Second, evaluate our simple delay penalty functions on any of these timetables for different distributions of the delays and delay-weighting factors. Third, simulate their behavior by confronting any of these timetables with a sample set of scenarios and compute optimal disposition timetables. Fourth, calibrate the simplified objective function via the distribution and the delay-weighting factor s to best fit the cost observed in the simulation. Select the distribution and delay-weighting factor that fits best. Now, compute with these parameters in the simplified objective function the delay resistant periodic timetable.

Last but not least, let us mention that we are still seeking for a detailed feedback from practitioners, but due to the huge variety of data, this remains a challenge in its own right.

References

[1] N. Bissantz, S. Güttler, J. Jacobs, S. Kurby, T. Schaer, A. Schöbel, and S. Scholl. DisKon - Disposition und Konfliktlösungs-management für die beste Bahn. *Eisenbahntechnische Rundschau (ETR)*, 45(12):809–821, 2005. (in German).

[2] C. Conte and A. Schöbel. Identifying dependencies among delays. In *proceedings of IAROR 2007*, 2007. ISBN 978-90-78271-02-4.

[3] J. R. Daduna and S. Voß. Practical experiences in schedule synchronization. In J. R. Daduna, I. Branco, and J. M. P. Paixao, editors, *CASPT*, volume 430 of *Lecture Notes in Economics and Mathematical Systems*, pages 39–55. Springer, 1995.

[4] J. Ehrhoff, S. Grothklags, and U. Lorenz. Parallelism for perturbation management and robust plans. In J. C. Cunha and P. D. Medeiros, editors, *Euro-Par*, volume 3648 of *Lecture Notes in Computer Science*, pages 1265–1274. Springer, 2005.

[5] O. Engelhardt-Funke and M. Kolonko. Analysing stability and investments in railway networks using advanced evolutionary algorithms. *International Transactions in Operational Research*, 11:381–394, 2004.

[6] M. Fischetti and M. Monaci. Robust optimization through branch-and-price. In *Proceedings of AIRO*, 2006.

[7] M. Gatto, B. Glaus, R. Jacob, L. Peeters, and P. Widmayer. Railway delay management: Exploring its algorithmic complexity. In *Algorithm Theory - Proceedings SWAT 2004*, volume 3111 of *LNCS*, pages 199–211. Springer, 2004.

[8] M. Gatto, R. Jacob, L. Peeters, and A. Schöbel. The computational complexity of delay management. In D. Kratsch, editor, *Graph-Theoretic Concepts in Computer Science: 31st International Workshop (WG 2005)*, volume 3787 of *Lecture Notes in Computer Science*, 2005.

[9] M. Gatto, R. Jacob, L. Peeters, and P. Widmayer. On-line delay management on a single train line. In *Algorithmic Methods for Railway Optimization*, Lecture Notes in Computer Science. Springer, 2007. presented at ATMOS 2004, to appear.

[10] A. Ginkel and A. Schöbel. To wait or not to wait? The bicriteria delay management problem in public transportation. *Transportation Science*, 41(4):527–538, 2007.

[11] L. Giovanni, G. Heilporn, and M. Labbé. Optimization models for the delay management problem in public transportation. *European Journal of Operational Research*, 2006. to appear.

[12] B. Heidergott and R. de Vries. Towards a control theory for transportation networks. *Discrete Event Dynamic Systems*, 11:371–398, 2001.

[13] W.-D. Klemt and W. Stemme. Schedule synchronization for public transit networks. In J. R. Daduna and A. Wren, editors, *Computer-Aided Transit Scheduling— Proceedings of the Fourth International Workshop on Computer-Aided Scheduling of Public Transport*, volume 308 of *Lecture Notes in Economics and Mathematical Systems*, pages 327–335. Springer, 1988.

[14] L. Kroon, R. Dekker, and M. Vromans. Cyclic railway timetabling: A stochastic optimization approach. In *Algorithmic Methods for Railway Optimization*, 2005. To appear. Preprint available at `http://www.few.eur.nl/few/research/ecopt/publications`.

[15] L. G. Kroon and L. W. Peeters. A variable trip time model for cyclic railway timetabling. *Transportation Science*, 37:198–212, 2003.

[16] C. Liebchen. *Periodic Timetable Optimization in Public Transport*. dissertation.de – Verlag im Internet, Berlin, 2006.

[17] C. Liebchen. The first optimized railway timetable in practice. *Transportation Science*, 2008. accepted for publication.

[18] C. Liebchen and R. H. Möhring. The modeling power of the periodic event scheduling problem: Railway timetables - and beyond. In F. Geraets, L. G. Kroon, A. Schöbel, D. Wagner, and C. D. Zaroliagis, editors, *ATMOS*, volume 4359 of *Lecture Notes in Computer Science*, pages 3–40. Springer, 2004.

[19] C. Liebchen and L. W. Peeters. Integral cycle bases for cyclic timetabling. *Discrete Optimization*, 2008. accepted for publication; Preprint 761-2002 of TU Berlin, Mathematical Institute.

[20] C. Liebchen, M. Proksch, and F. H. Wagner. Performance of algorithms for periodic timetable optimization. In M. Hickman, P. Mirchandani, and S. Voß, editors, *Computer-Aided Transit Scheduling— Proceedings of the Ninth International Workshop on Computer-Aided Scheduling of Public Transport (CASPT)*, volume 600 of *Lecture Notes in Economics and Mathematical Systems*, pages 151–180. Springer, 2008.

[21] C. Liebchen and S. Stiller. Delay resistant timetabling. *Public Transport*, 2008. accepted for publication; Preprint 024-2006 of TU Berlin, Mathematical Institute.

[22] T. Lindner. *Train Schedule Optimization in Public Rail Transport*. Ph.D. thesis, Technische Universität Braunschweig, 2000.

[23] K. Nachtigall. *Periodic Network Optimization and Fixed Interval Timetables*. Habilitation thesis, Universität Hildesheim, 1998.

[24] M. A. Odijk. A constraint generation algorithm for the construction of periodic railway timetables. *Transportation Research B*, 30(6):455–464, 1996.

[25] B. D. S. R. de Vries and B. D. Moor. On max-algebraic models for transportation networks. In *Proceedings of the International Workshop on Discrete Event Systems*, pages 457–462, Cagliari, Italy, 1998.

[26] R. T. Rockafellar. *Network flows and monotropic optimization*. John Wiley & Sons, Inc., 1984.

[27] M. Schachtebeck and A. Schöbel. IP-based techniques for delay management with priority decisions. In *ATMOS 2008 - 8th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems*, Dagstuhl, Germany, 2008. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany.

[28] A. Schrijver and A. G. Steenbeek. Dienstregelingontwikkeling voor Railned. Rapport CADANS 1.0, Centrum voor Wiskunde en Informatica, December 1994. In Dutch.

[29] A. Schöbel. A model for the delay management problem based on mixed-integer programming. *Electronic Notes in Theoretical Computer Science*, 50(1), 2001.

[30] A. Schöbel. *Customer-oriented optimization in public transportation*, volume 3 of *Optimization and Its Applications*. Springer, New York, 2006.

[31] A. Schöbel. Capacity constraints in delay management. Technical report, ARRIVAL Report, 2007. TR-0017.

[32] A. Schöbel. Integer programming approaches for solving the delay management problem. In *Algorithmic Methods for Railway Optimization*, number 4359 in Lecture Notes in Computer Science, pages 145–170. Springer, 2007.

[33] P. Serafini and W. Ukovich. A mathematical model for periodic scheduling problems. *SIAM Journal on Discrete Mathematics*, 2(4):550–581, 1989.

[34] L. Suhl, C. Biederbick, and N. Kliewer. Design of customer-oriented dispatching support for railways. In S. Voß and J. Daduna, editors, *Computer-Aided Transit Scheduling*, volume 505 of *Lecture Notes in Economics and Mathematical systems*, pages 365–386. Springer, 2001.

[35] L. Suhl and T. Mellouli. Supporting planning and operation time control in transportation systems. In *Operations Research Proceedings 1996*, pages 374–379. Springer, 1997.

[36] L. Suhl and T. Mellouli. Requirements for, and design of, an operations control system for railways. In *Computer-Aided Transit Scheduling*. Springer, 1999.

[37] L. Suhl, T. Mellouli, C. Biederbick, and J. Goecke. Managing and preventing delays in railway traffic by simulation and optimization. In M. Pursula and Niittymäki, editors, *Mathematical methods on Optimization in Transportation Systems*, pages 3–16. Kluwer, 2001.

[38] U. Wagner. Pünktliche Abfahrt verhindert Domino-Effekt auf dem Gleis. BahnZeit *(employee newsletter of Deutsche Bahn AG)*, 5:4, 2004. In German.

Institut für Numerische und Angewandte Mathematik
Universität Göttingen
Lotzestr. 16-18
D - 37083 Göttingen

Telefon: 0551/394512
Telefax: 0551/393944

Email: trapp@math.uni-goettingen.de URL: http://www.num.math.uni-goettingen.de

Verzeichnis der erschienenen Preprints 2008:

2007-01	P. Serranho	A hybrid method for inverse scattering for sound-soft obstacles in 3D
2007-02	G. Matthies, G. Lube	On streamline-diffusion methods for inf-sup stable discretisations of the generalised Oseen Problem
2007-03	A. Schöbel	Capacity constraints in delay management
2007-04	J. Brimberg, H. Juel, A. Schöbel	Locating a minisum circle on the plane
2007-05	C. Conte, A. Schöbel	Identifying dependencies among delays
2007-06	D.S. Gilliam, T. Hohage, X. Ji, F. Ruymgaart	The Frechet-derivative of an analytic function of a bounded operator with some applications
2007-07	T. Hohage, M. Pricop	Nonlinear Tikhonov regularization in Hilbert scales for inverse boundary value problems with random noise
2007-08	C.J.S. Alves, R. Kress, A.L. Silvestre	Integral equations for an inverse boundary value problem for the two-dimensional Stokes equations
2007-09	A. Ginkel, A. Schöbel	To wait or not to wait? The bicriteria delay management problem in public transportation
2007-10	O. Ivanyshyn, R. Kress	Inverse scattering for planar cracks via nonlinear integral equations
2007-11	F. Delbary, K. Erhard, R. Kress, R. Potthast, J. Schulz	Inverse electromagnetic scattering in a two-layered medium with an application to mine detection
2007-12	M. Pieper	Vector hyperinterpolation on the sphere
2007-13	M. Pieper	Nonlinear integral equations for an inverse electromagnetic scattering problem
2007-14	I. Akduman, R. Kress, N. Tezel, F. Yaman	A second order Newton method for sound soft inverse obstacle scattering

2007-15	M. Schachtebeck, A. Schöbel	Algorithms for delay management with capacity constraints
2007-16	R. Kress, F. Yaman, A. Yapar, I. Akduman	Inverse scattering for an impedance cylinder buried in a dielectric cylinder
2007-17	T. Hohage, L. Nannen	Hardy space infinite elements for scattering and resonance problems
2007-18	De Marchi, Stefano, Schaback, R.	Stability of kernel-based interpolation
2007-19	Schaback, R.	Solving the Laplace Equation by Meshless Collocation Using Harmonic Kernels
2007.20	Schaback, R.	Adaptive numerical solution of MFS systems
2007.21	Schaback, R.	A Posteriori Error Estimates for Meshless Methods
2007.22	Lee, C.-F., Ling, L., Schaback, R.	On Convergent Numerical Algorithms for Unsymmetric Collocation
2007.23	Ling, L., Schaback, R.	Stable and Convergent Unsymmetric Meshless Collocation Methods
2007-24	C. Liebchen, M. Schachtebeck, A. Schöbel, S. Stiller, A. Prigge	Computing Delay Resistant Railway Timetables
2007-25	G. Lube, T. Knopp, G. Rapin, R. Gritzki, M. Rösler	Application of stabilized finite element methods to indoor air flow simulations
2007-26	X.Q. Zhang, T. Knopp, G. Lube	Calibration of model and discretization parameters for turbulent channel flow
2007-27	G. Lube, G. Rapin, J. Löwe	Local projection stabilization for inf-sup stable finite elements applied to the Oseen problem
2007-28	G. Lube, G. Rapin, J. Löwe	Local projection stabilization for incompressible flows: Equal-order vs. inf-sup stable interpolation