Technische Universität Berlin

FAKULTÄT II
MATHEMATIK UND
NATURWISSENSCHAFTEN
Institut für Mathematik

Computing Delay Resistant Railway Timetables

by

Christian Liebchen
Michael Schachtebeck
Anita Schöbel   Sebastian Stiller
André Prigge

No. 2007/17

Computing Delay Resistant Railway Timetables*

Christian Liebchen Michael Schachtebeck
Anita Schöbel Sebastian Stiller André Prigge

March 31, 2007

Abstract

In the past, much research has been dedicated to compute optimum railway timetables. A typical objective was the minimization of passenger waiting times. But only the planned nominal waiting times were addressed, whereas delays, as they occur in daily operations, were neglected. Delays were 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 *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 was 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

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. Delay resistance may not be attained by inserting arbitrarily large time supplements. Rather, the art of delay resistant timetabling 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 ([KDV05, LS06]). 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. They 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, smarter reactions 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 *delay management problem* and includes to decide which connections between trains should be maintained and which not, see [Sch06b] and references therein.

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 behave, that has been optimized for a simplified delay management? The question is, whether those timetables keep what they promise, namely, that disturbances do not affect the quality of service to the passengers too much, if good delay management strategies are applied.

Although this is a straightforward question, it has to the best of our knowledge not been treated in the literature before. In our 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 had been proposed by Fischetti and Monaci [FM06]. 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 about the quality of the original timetable.

This case study also allows to evaluate the power of optimal delay management. In fact, we will also calculate the delay experienced by the passengers, if the same timetables and the same set of source-delays are managed by a simple delay management policy.

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. But it is optimal for each of these.

It turns out that the effects of delays are less worse 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. Finally, the delay resistant timetabling proves particularly effective against short delays.

Related Work. The only comparable work we are aware of is due to Engelhardt-Funke and Kolonko, see [EFK04]. Unlike ours their approach is able to integrate the delay management into the construction of the timetables by evolutionary algorithms. Our approach in the first step constructs *optimal* timetables with respect to the expectation over all scenarios for a carefully simplified objective function. In the second step it evaluates them with a limited number of scenarios and an *optimal* delay management policy, thus for the real objective function in these scenarios. In contrast, in [EFK04] for the heuristic construction of the timetable an evaluation is used, which is not based on an optimal delay management and which solely relies on a limited number of scenarios. Their construction of the timetable is not an optimization in the strong sense, and they never pay respect to an optimal delay management nor do they at any point take into account all scenarios. The carefully simplified 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. We give an exact interpretation of this objective function, that allows to choose suitable parameters. Note again, that for evaluation we use an exact objective, involving optimal delay management.

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 explain in detail how event-activity networks can be used to model both the periodic event scheduling problem as well as the delay management problem. Our approaches for finding delay resistant timetables, using them as input for defining delay management problem, 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 conclusions for further research.

2 Models

In this paper we deal with two mathematical optimization problems: In our first step we compute optimal periodic timetables with a 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 [Roc84]. 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

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

Observe that by introducing antiparallel arcs, formally each instance of FDP can be transformed such that $\ell_a = -\infty$, for all $a \in A'$. What remains are nothing but the optimality conditions for the distance labels in the Shortest Path Problem (SPP). In particular, for an instance $\mathcal{I}$ of FDP there exists a solution π, if and only if the transformed instance of SPP does not contain any directed circuit of negative length.
The nodes in the network are called *events* and correspond to arrivals or departures of trains at stations, i.e.

$$V = V_{arr} \cup V_{dep}.$$

The vector π is then called a *timetable*, the value π_i is the time instant at which the event i is scheduled. One may think of π_i as an absolute point in time of the day.
Of course, the times π_i must not be chosen arbitrarily. Rather, they must respect many operational and marketing requirements. These are encoded by the constraints (1) that are defined for the arcs of the network. Observe that the quantities $\pi_j - \pi_i$ are time durations, and we refer to the arcs as *activities*. We partition the set of activities into five subsets

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

where we explain the meaning of these subsets right after having presented how periodicity is added to FDP.

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.
We obtain subsets of non-periodic events i_k that take place at time $\pi_{i_k} = \pi_i + kT$ for $a_i < k < b_i, k \in \mathbb{Z}$. 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.
Consequently, if the goal is to find a periodic timetable, the mathematical optimization model features the variables π_i for the periodic time assigned to the equivalence classes rather than those for the non-periodic events i_k. It is

fruitful 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 values π_i are of course still time instants. But in contrast to the values π_{i_k}, they do no longer refer to an absolute point in time of the day. Rather, they refer to a point in time of one abstract copy of the period time. Section 3.3 describes in detail how the periodic events' times are rolled out to obtain the non-periodic event times.
The corresponding decision problem is called Periodic Event Scheduling Problem (PESP), see [SU89]. In addition to the input to an instance of FDP, a fixed constant period time T is specified. But in contrast to other approaches to timetabling ([DV95, KS88]), in any of our computations we will use T only in its binary 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 ([Odi96]). As in the non-periodic case, we also partition the set $\underline{A}$ of periodic constraints into five subsets

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

Observe that the PESP is the building block of many studies on periodic railway timetabling ([SS94, Nac98, Lin00, KP03, Lie06]). In particular, the first mathematically optimized timetable that went into daily operation has been computed based on the PESP ([Lie05a]).

On the one hand, periodic timetables are a specialization of general timetables in which each trip may be scheduled individually. Hence, in terms of well-quantifiable objectives no periodic timetable will ever be strictly better than the best general timetable. On the other hand, there have been identified many non-quantifiable criteria with respect to which periodicity adds an additional benefit ([Lie05b]). Indeed, most public transportation companies in Europe operate their networks subject to periodic timetables. This is why in our case study we will always compute periodic timetables as the regular service timetables. 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.

Now we explain the meaning of the five subsets of activities. Only the headway activities require different treatments in the non-periodic case (A_{head})

and in the periodic case ($\underline{A}_{head}$). We will leave out the $\underline{A}$ notation for a moment.

- *Driving activities.* The arcs in A_{drive} represent the driving of a train between consecutive stations, i.e. a driving activity connects a departure event of some train t with its next arrival event. The bounds ℓ_a and u_a represent the minimum and maximum allowed driving time.

- *Stopping activities.* The arcs in A_{stop} model the time period in which a train is stopping in a station to allow passengers to board and un-board. A stopping activity hence connects an arrival event with a departure event, if both events refer to the same train and the same station.

- *Transfer activities.* The arcs in $A_{transfer}$ indicate that a transfer for the passengers is possible from some train t_1 to another train t_2. Consequently, a transfer activity connects the arrival of train t_1 at a station to the departure of train t_2 at the same station. The bound ℓ_a refers to the minimum time passengers need when they change from t_1 to t_2. And u_a ensures a minimum level of quality because it prevents the transfer waiting time from getting arbitrarily large.
  Note that in the periodic formulation the optimizer retains the freedom to implicitly decide to which non-periodic event π_{j_k} a passenger arriving with event $\pi_{i'_k}$ will connect. The offset k_a is chosen to ensure that for given π_i, π_j the shortest connection time, i.e., the connection to the next train from A to B is counted.

- *Turnaround activities.* The arcs in A_{turn} are introduced, because each timetable does not only affect the passengers, but also has an immediate impact on the operating costs, i.e., on the number of trains that are required to operate the timetable. A turnaround activity thus measures the time duration from the arrival of a physical train in its terminus station until the departure of the same physical train, e.g., in the opposite direction. The lower bound ℓ_a ensures that a train will only be planned for its next trip, if that trip respects the (technical) minimum turnaround time, see [LM04] for further details.

- Finally, the *headway activities* A_{head} are needed to model the limited capacity of the track system. There are two types: The first type refers to trains departing from the same station into the same direction. By setting the lower bound ℓ_a as the minimum security distance, they guarantee that there is enough time between the two departures. In the periodic case, the upper bound u_a then is $T - \ell_a$, where we assume identical speeds of the trains, but different speeds can be modeled, too, see [LM04, Lie06]. The second type refers to single track segments between two stations. Here, a train has to wait until an oncoming train has arrived.

In the aperiodic case, both types of headway constraints have to be modeled as disjunctive constraints, i.e. only one of each pair of them has to be respected: Either i has to wait until i' is finished (plus some security time $\ell_{ii'}$) or vice versa. This is due to the fact that the order of the events i, i' is not given a priori.

The travel times and transfer activities will play the central role in our stochastic expansion of the PESP, see Section 3.2. To conclude, a great variety of features of railway timetabling is covered by event-activity models like FDP and PESP. Many more practical requirements can be modeled, see [LM04, Lie06] for the case of periodic timetabling. Hence, these models are the models of choice for the timetabling problems that we are going to deal with.

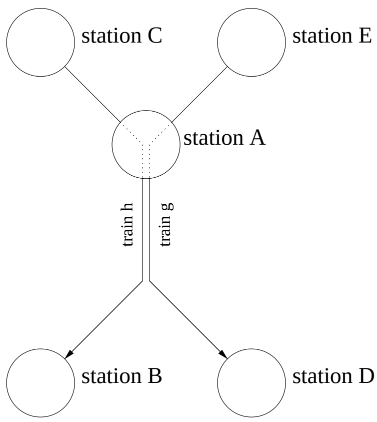

Figure 1: A part of a public transportation network with five stations and two trains. After their departure in A both trains have to use the same track.

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

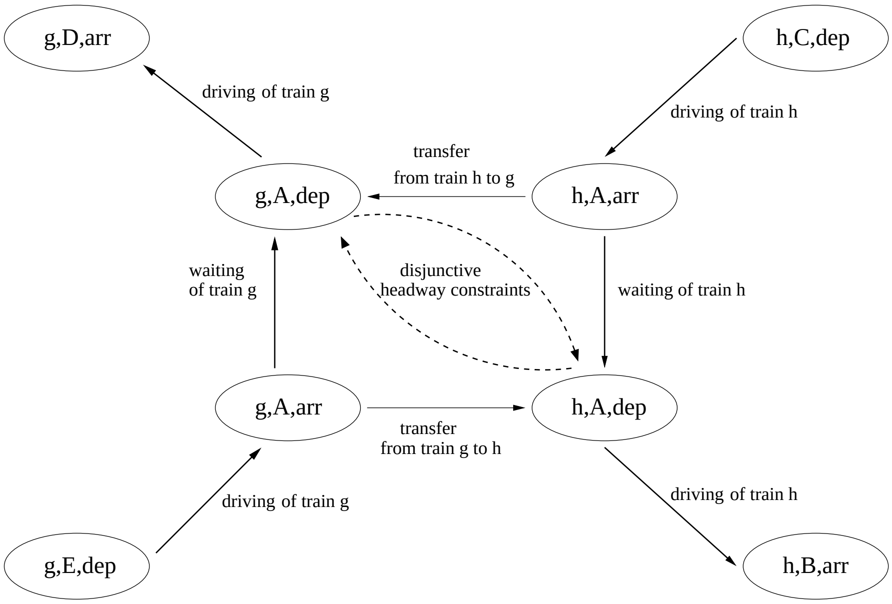

Figure 2: The event-activity network corresponding to Figure 1.

defining strict upper bounds $u_a - \ell_a$ on the slack, e.g. for the most important transfer activities.
There exists an alternative to ensure transfer quality in a network: minimizing the objective of total passenger transfer times in the entire network. To this end, we must be given the number of passengers w_a for each transfer activity $a \in A_{trans}$, or $a \in \underline{A}_{trans}$, respectively. Formally, we simply have to minimize the linear objective function of the weighted sum of slack. Therefore, in the periodic case the upper bound u_a is set to $u_a = \ell_a + T - 1$, whereby any (integer) transfer waiting time becomes feasible. Still shorter waiting times are better than longer ones in respect to the linear objective function.
Another motivation for a linear objective function is the fact that the timetable has an immediate effect on the number of trains that are required for its operation. In the case of periodic timetables, it has been shown that in the PESP model only a linear objective function admits an adequate integration of the corresponding real-world piecewise constant cost function ([LM04]). In the case of a linear objective function being added to an instance of PESP, we speak of an *optimization instance of PESP*.

2.2 The Source of Delay

As model of disturbances we assume the driving time of each train to vary according to a known probability distribution. Therefore, the lower bounds for arcs $a \in \underline{A}_{drive}$ or $\in A_{drive}$ corresponding to a single (periodic) train ride are random variables $\ell_a : R \to \mathbb{Q}^+$, where R is the set of all scenarios. To comply with standard notions such a disturbance from now on will be called a *source delay*. We distinguish between source delays, i.e., the seminal prolongation of a trains' driving time, and delays 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. Whereas in a different situation, e.g., a schedule including a large amount of slack time, a source delay may result in no delay at all.

As we are optimizing periodic timetables we also have to explain how source delays and resulting delays fit into the periodic framework. Both, the periodic and the non-periodic, singular perspective on disturbances is reasonable. A periodic source delay is, e.g., a construction site slowing down all train trips realizing a certain driving arc in every period. Whereas a singular source delay could be a jammed door delaying a single train trip realizing this driving arc only in a single period. As singular source delays make up for an important part of the practically relevant source delays, we cannot neglect them. But a single singular source delay could affect several periods, and each in a different way. Thus its incorporation into the stochastic PESP model promises to be difficult. Periodic source delays cause the question, whether the source delay of each period will be absorbed until the similar event of the next period takes place. If this is not the case periodic source delays have non-periodic consequences, too. Fortunately, this difference will be immaterial in our optimization approach.

Moreover, we only specify the boundary distribution on each driving arc instead of their joint distribution. In fact, for our optimization approach it will be sufficient to know these boundary distributions, as will become clear in Section 3.2.

In order to assess the delay resistance of a schedule the optimizer needs not only to know the distribution of the source delay but also the reaction towards source delay in a realization. This reaction is what is called *delay management policy*.

We will explain and formalize the problem of delay management in the following section.

An ideal optimization takes the expectation of some objective function which describes the behavior of the timetable and a corresponding optimal disposition timetable for each scenario, including singular and periodic source delays, weighted according to their joint distribution. Here one has to make some careful but effective simplifications to compute the timetables. This will be found in Section 3.2.

2.3 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. In case of such delays, the question is to decide if trains should wait for delayed feeder trains or if they better depart on time (*wait-depart decisions*). From these decisions one obtains a *disposition timetable*. Such a timetable 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 gets NP-hard even in very special cases, see [GJPS05, GGJ+04, CS07]. Neglecting only the headway activities, we obtain the (pure) delay management problem. An integer programming model based on a representation of the problem in an activity-on-arc-project network is given in [Sch01, Sch06b]. These publications include an investigation of the special structure of the activity networks used in delay management. The model will be described in detail in Section 3.4. The general integer programming model was further refined in [GHL06].
A first online-approach is provided in [GJPW07]. A bicriteria model for delay management in the context of max-plus-algebra has been presented in [RdVM98], a formulation as discrete time-cost tradeoff problem is given in [GSar]. How to react in case of delays has also been tackled by simulation and expert systems. We refer to [SM97, SM99, SBK01, SMBG01] for providing knowledge-based expert systems including 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 [BGJ+05]).

3 Integer programs

In the previous section we explained the pure theory, the ideal model. In this section we are looking for ways to solve it. First we explain the integer

programming approach to the deterministic PESP. After that we give a detailed description how we incorporate stochasticity and delay management into this approach. This incorporation process has to find a smart balance between accurate modeling and the mandatory feature not to increase the size of the PESP-IP substantially. This is mandatory in order 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 of these instances entails a concrete scenario of source delays. In the last part of this section we explain one more 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) are 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 ([LPW05]), 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 ([Nac98]). In the context of electrical engineering, one would call these new variables x the

(periodic) tension that is induced by some node potential π. It is convenient to think of x as $x_a = \pi_j - \pi_i + T \cdot k_a$. The resulting equivalent IP finally reads

$$\textbf{(PESP-IP-}x\textbf{-}z\textbf{)} \qquad \min f(x,z) = \sum_{a \in \underline{A}} \tilde{w}_a \cdot (x_a - \ell_a)$$

such that

$$\begin{aligned} x_a &\leq u_a \quad \text{for all } a \in \underline{A} && (8)\\ x_a &\geq \ell_a \quad \text{for all } a \in \underline{A} && (9)\\ \Gamma^{\mathsf{T}} x - T \cdot z &= 0 && (10)\\ x_a &\in \mathbb{Z} \quad \text{for all } a \in \underline{A} && (11)\\ z_C &\in \mathbb{Z} \quad \text{for all } C \in B, && (12) \end{aligned}$$

where the $|\underline{A}| \times (|\underline{A}| - |\underline{V}| + 1)$-matrix Γ is the arc-cycle incidence matrix of some integral cycle basis B of the directed graph $\underline{D} = (\underline{V}, \underline{A})$, see [LP02, LR07] for the relevant properties of integral cycle bases of graphs.
In fact one can take the most benefit out of PESP-IP-x-z by adding further valid inequalities. Such have been proposed by Odijk ([Odi96]) and Nachtigall ([Nac98]), and we use them throughout any of our PESP optimization runs.

3.2 Adding Delay-Resistance

Optimizing delay resistant timetables we simplify certain aspects of the underlying model. We will present the model and point out our simplifying assumptions. Some effects of the simplification can be quantified a priori, some cannot. Nevertheless, we expect all simplifications to be tolerable in the sense that the resulting plan performs well in practice. In order to verify this claim, we test our plan in a non-simplified environment a posteriori.
Of course, the non-simplified testing can only take into account a limited number of scenarios, whereas the simplified version is optimized with respect to all scenarios. Optimizing or even assessing the accurate model over all scenarios is far beyond the scope of computational possibilities.

3.2.1 Simplified Delay Management

The most important simplification concerns the delay management which we assume to underly the performance of a timetable in a scenario, i.e., which determines the delay related part of the objective function. As delay

management policies and delay propagation are notoriously hard problems in their own right, *for the optimization* of the periodic timetable some kind of simplification at this point is unavoidable.
We assume a strict no-wait policy. This means for the assumed disposition timetable every (non-periodic) 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. However, such a strict no-wait policy cannot be implemented in practice. This is due to the following reasons:

- The headway activities in the PESP 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. Among these activities the order is of course mandatory.

- One physical train usually takes more than one trip. But it cannot start the second trip before it has finished the first. This is modeled by the turnaround activities in PESP, together with a linear objective function, but also neglected in the strict no-wait policy.

Clearly, constraints expressing such conditions cannot be dropped in reality. At some arcs an all-wait policy and therefore a propagation of source delays is a matter of fact.
Hence, adopting the strict no-wait policy we underestimate the effects of source delays, as we exclude delay propagation. This unrealistic assumption yields that *every* departure event will in every scenario take place as scheduled. As a drawback, this implies that our optimization cannot respond to the *propagation* of delay.
More precisely: the strict no-wait policy $\mathcal{SN}$ which we assume for optimization is not practical. Compared to the practical no-wait policy $\mathcal{PN}$, i.e., a delay management that always decides to start as early as *possible* (though not ahead of schedule), the expected delays ($\mathbb{E}[D(\cdot)]$) of the 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 $\mathbb{E}[D(\mathcal{OM})] \leq \mathbb{E}[D(\mathcal{PN})]$. Whether the strict no-wait policy performs better 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.

3.2.2 Strict No-Wait Policy and Periodic Timetabling

At first glance the unrealistic assumption of a strict no-wait policy cancels all effects of stochastic driving times. Looking closer, this approach within periodic timetabling only neglects the technical aspects of the railway system, whereas it accounts correctly for the passengers' perspective. This will become clear by the following considerations.

Scenario Expansion. For explanatory reasons we introduce a local scenario expansion of the PESP graph that models correctly the strict no-wait policy.

Assume the distribution of a train's driving time, which is the random variable giving the lower bound ℓ_a of the corresponding arc a, to be finitely discretized in $h \in \mathbb{N}$ scenarios. To construct the expansion, substitute a driving arc a by h arcs a^r each starting at the same departure vertex but leading to its own copy j^r of the arrival vertex. For each train to which passengers will transfer, those arrival vertices are connected to the single departure vertex of the transfer train, see Figure 3. For each r the set of arcs incident to j^r are an entire copy of those incident to j in the original graph. In fact r stands for a scenario, and arcs incident to j^r shall model what happens to the transfer passengers arriving at j in scenario r. Each copy a^r of the driving arc gets a different, fixed value for its duration, the driving time in that scenario. Each transfer arc is weighted according to the probability that the driving time of its start vertex' unique incoming arc occurs. (The upper bounds on a driving arc a^r are set equal to the lower bound, i.e., the actual driving time.)

Interpretation of the Expansion. In this way, a pair of departure events of trains between which passengers transfer, is linked by a set of parallel paths of length 2. Each of the paths represents a scenario. The first arc has fixed length, namely the scenario's driving time on that track. The second arcs' length, i.e., the transfer times in the different scenarios, are set simultaneously by the optimization process deciding on the relative values of the departure events of the two trains.

The optimization will seek to keep these transfer times short. In a non-periodic setting this would be trivial, as feasibility would imply that the schedule is dictated by the longest driving time. In the periodic setting this is not the case. A path representing a scenario with low probability but long driving time might be quasi neglected in order to have 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,

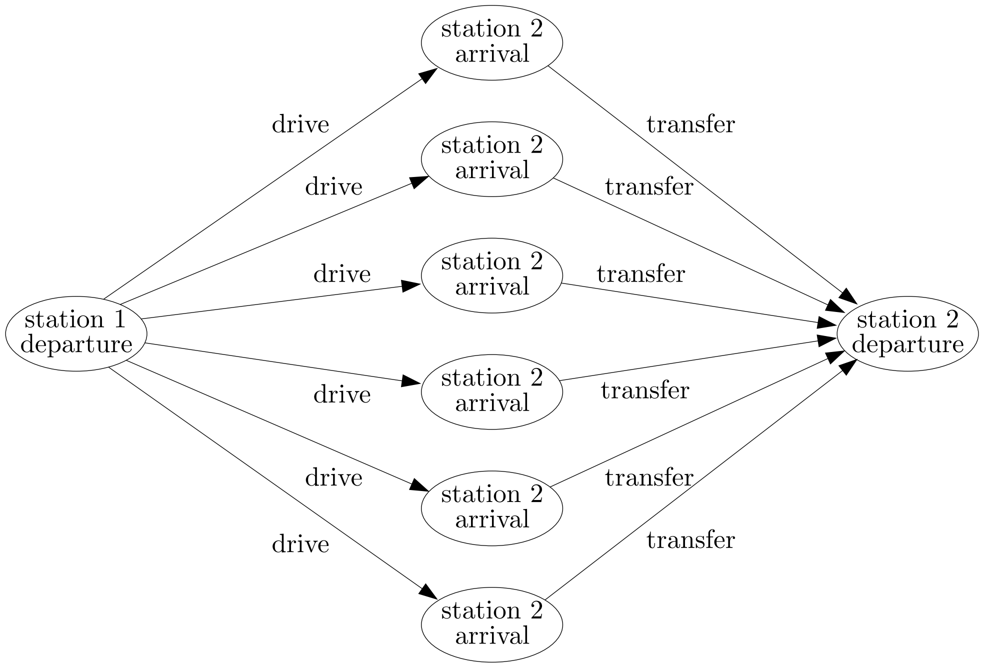

Figure 3: Expansion of a driving and waiting activity to six different scenarios.

that those passengers connect to the train of the next period. The waiting time for this is expressed correctly in the periodic model.
In other words: In periodic timetabling a strict no-wait policy is realistic for arcs expressing passengers' connections to other trains. Here the connections are not broken but only established with a later period, for which the waiting time can be expressed correctly by a PESP graph with only a constant bigger size than the deterministic problems' PESP graph.
The technique we exemplified for the transfer arcs can also be applied to other arcs that enter the objective function.

Stochastic Dependency. Nevertheless, the effects of uncertainty remain local by virtue of the strict no-wait policy. Therefore it suffices to know the boundary distributions for each arc's driving time. In this model it does not matter which joint distribution underlies the driving times of the trains.
Moreover, it is immaterial in this setting whether a source delay is periodic or singular. In both cases the measure is the probability for a driving time to occur weighted over all scenarios and—with uniform distribution—over all periods. As there is no delay propagation a singular source delay that occurs on average in 1 out of 3 realizations of a periodic trip, has the same effect,

as a periodic source delay that occurs in 1 out of 3 scenarios for all periods.

3.2.3 Objective Function

In the light of stochastic driving times two different objective functions appeal to us. Prima facie, one can be interested in minimizing the sum of the expected travel times of all passengers, where the travel time of a passenger i is the span between the moment she gets on the first train of her journey (t^i_{str}) until her last train arrives at her destination (t^i_{arr}).

$$E\left[\sum_i t^i_{\mathrm{arr}} - t^i_{\mathrm{str}}\right] = \sum_r \left[\sum_{a \in A_{drive}} w_a t^r_a + \sum_{a \in A_{wait}} w_a t^r_a + \sum_{a \in A_{transfer}} w_a t^r_a\right]$$

This measure nicely decomposes into the expected driving and stopping time of the trains weighted by their passenger load and the expected, weighted time passengers have to wait for a transfer. The latter is the matter of optimization. Invoking again the above discretization of the distributions, and the described scenario expansion of the graph, the expected transfer time of all passengers is counted correctly by summing the length of the copies of the transfer arcs weighted by the probability of the corresponding scenario. A more detailed description of this objective function, the graph expansion and its non-discretized version can be found in [LS06].
Though natural at first sight, this objective function neglects an important aspect of delay resistant timetabling and delay management. Delay is the deviation from what was planned. A train that arrives notoriously five minutes behind schedule causes greater discomfort to the customers, than a train that is scheduled to arrive five minutes later. In the first case, the company breaks a promise, whereas in the latter case, the passenger already accepted a less ambitious promise. Nominal travel time and expected delay, i.e., expected positive deviation from the schedule, must be weighed against each other. In general a minute of expected delay weighs heavier than a minute prolongation of nominal travel time. The objective described so far measures the expected travel time of the passengers, with no respect to how much it deviates from what passengers planned. The customers plans are generated by the published timetable. To capture this aspect the objective function must have some reference to the nominal schedule.
Before showing how this can be—and actually—is incorporated in our setting, we address the delay resistant optimization of arrival times, which we did not incorporate in this study, although it fits into our framework, too.

Arrival Events. In the graph expansion each vertex of an arrival event is substituted by a set of arrival vertices, one for each scenario. The departure events are not duplicated. This is because they play substantially different roles in the interpretation. An arrival vertex corresponds to the arrival event of a scenario. This is also true for the non-duplicated departure events, because in all scenarios every departure is on time. But the value of a departure vertex also corresponds to the time published in the timetable. Which time shall be published for an arrival event?
This matters in two respects: First, the published time is the time with respect to which the delay in a certain scenario is measured.
Second, the relative time difference between the published arrival time of a train and the published departure time of another train at the same station decides which transfer time between the two trains is offered. For every transfer a minimum transfer time is known. A passenger information system that chooses a transfer will send the passenger to the earliest train reachable after the minimum transfer time. Every transfer planned with at least this minimum transfer time, is promised. This implies, that by fixing the arrival time early enough for a quick transfer the railway company gives the service guarantee to provide for this transfer, for which it will be liable—at least in our objective function.
In practice planners sometimes use the arrival times to actively negate such a service guarantee. A tight connection is rendered nominally impossible by prolongating the nominal driving time in order to avoid liability in case the short transfer fails.
In principal, our approach models also the arrival times as being to the disposal of the optimizer, as it has also been done in [KDV05]. However, in this study we refrain from using this optimization potential.
More precisely, in accordance with Leaflet 451–1 of UIC (Union Internationale de Chemins de Fer), we fix the nominal driving time of each train to 107% of the technically minimal driving time, for performance reasons. In the disposition timetable that we construct in the delay management, in order to catch up their delays we allow trains to use 76.4% of these driving time supplements. 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 contrast, 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, which equals 76.4% of the added supplement of 7 min.
Thus, driving times are fixed for the planning. Their fluctuation in the scenarios must be compensated by planning buffered transfer times.

Incorporating Delay in the Objective Function. By fixing the nominal driving time of each train, we have a reference to measure when a passenger misses a guaranteed transfer. Assume, e.g., that 20% of the trains corresponding to a certain driving arc $a = (i, j)$ need more than 1.08 of the minimal driving time. Assume further we schedule the departure event h of a train, to which some passengers from a want to transfer, such that $\pi_h - \pi_i = \ell_{(j,h)} + 1.08t_{(i,j)}$, where $\ell_{(j,h)}$ is the minimum transfer time and $t_{(i,j)}$ denotes the minimum time of the driving activity corresponding to a. Then 20% of the passengers will experience a delay of the period time T: As the arrival event was published under the assumption that the train needs $1.07t_{(i,j)}$ the passengers expect to get their transfer—but 20% will not get it. The passengers, who miss the connection, plan to transfer to the train departing at time π_{h_k} but now have to wait for the train that departs at time $\pi_{h_{k+1}} = \pi_{h_k} + T$. Consequently, since trains shall never operate prior to schedule, these passengers have a definite delay of T, compared to their planned itinerary (or an integer multiple of T, in the case they even missed other transfers). Observe that counting a delay of T for such a missed transfer equals the sum of their extended transfer time *plus* the extended driving time of the feeder train. Hence, we only check for delays at transfers. The only error that we make this way is that we do not capture the delay that passengers face on the last trains of their itineraries.[1] Therefore, scheduling such that $\pi_h - \pi_i = \ell_{(j,h)} + 1.08t_{(i,j)}$ yields $0.2T$ minutes (20% of the passengers have to transfer to the train of the next period) of expected delay for the passengers who transfer from arrival j to departure h.
If we specify some *delay-weighting factor* s, such that one expected minute of delay is as important as s minutes of nominal prolongation of travel time, then we can incorporate the delay into the contribution of the transfer arc $b = (j, h)$ to the objective function. Finally, before entering the objective function, this contribution is multiplied by a weight, w_b. This is the same weight that is also used in a purely nominal objective function, to determine the importance of an arc b. This weight typically and in our study corresponds to the number of passengers using the transfer arc.

Convex and Piecewise-Linear Objective Functions. Let c be the resulting objective function and c_b its restriction to the component of the solution vector corresponding to the transfer arc b. The function $c_{b=(j,h)}$ depends on the distribution p_a for scenarios $r \in R$ and their random variable ℓ_a^r, i.e.,

[1] In general, of course we would like to include the delays on the final trains of itineraries as well. However, for this particular case study, unfortunately the data that would be necessary to incorporate this effect was not available to us.

the driving time of the driving arc $a = (i, j)$. (We denote the nominal driving time by t_a to stress the difference.) The cost c_b is independent of the other driving times' distributions.

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

The clumsy expression rounded down serves as an indicator, whether under timetable π in scenario r the passengers who transfer along arc b get their connection. If it becomes negative, they do not get it, and c_b increases by $w_b p_a(r) T$. When checking this, keep in mind that we only consider small disturbances $(\ell_a^r - t_a)$, thus the rounded expression has absolute value less than T. This cost function has a reference to the published timetable and its promises for transfer. Despite its intimidating aspect, it can be included into the PESP optimization under a mild assumption.

Such a function c_b need not be convex. However, the authors were told by practitioners that real-world distributions give almost convex functions in the sense that only a small error is incurred if one substitutes them by a convex function. Given that c_b is convex we perform a further approximation by piecewise-linearizing the function. In this study we decided to split our function into two intervals with negative slope and one interval with slope equal to w_b. The latter is the time interval when we expect all trains, that instantiate the driving arc of the feeder train in any scenario, have reached their destination.

Eventually, we have a piecewise-linear, convex objective function on each transfer arc. Such functions can easily be integrated to 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 [LS06]. We introduce an artificial fractional variable y_b. Additional linear constraints ensure that given a certain transfer time x_b, the value of the variable y_b will be at least equal to that of the desired, convex, piecewise-linear function at x_b. As we minimize the value of y_b, the convex, piecewise-linear function is minimized, too.

For clarity in presenting the stochastic effects in the objective function we present the contribution of the transfer arc b *before* the weighting with w_b. Figure 4 displays the two components of the contribution of a transfer arc b to the objective *without* multiplication by w_b. The x-axis gives the time planned for the transfer. The figure is translated such that it starts at the minimum

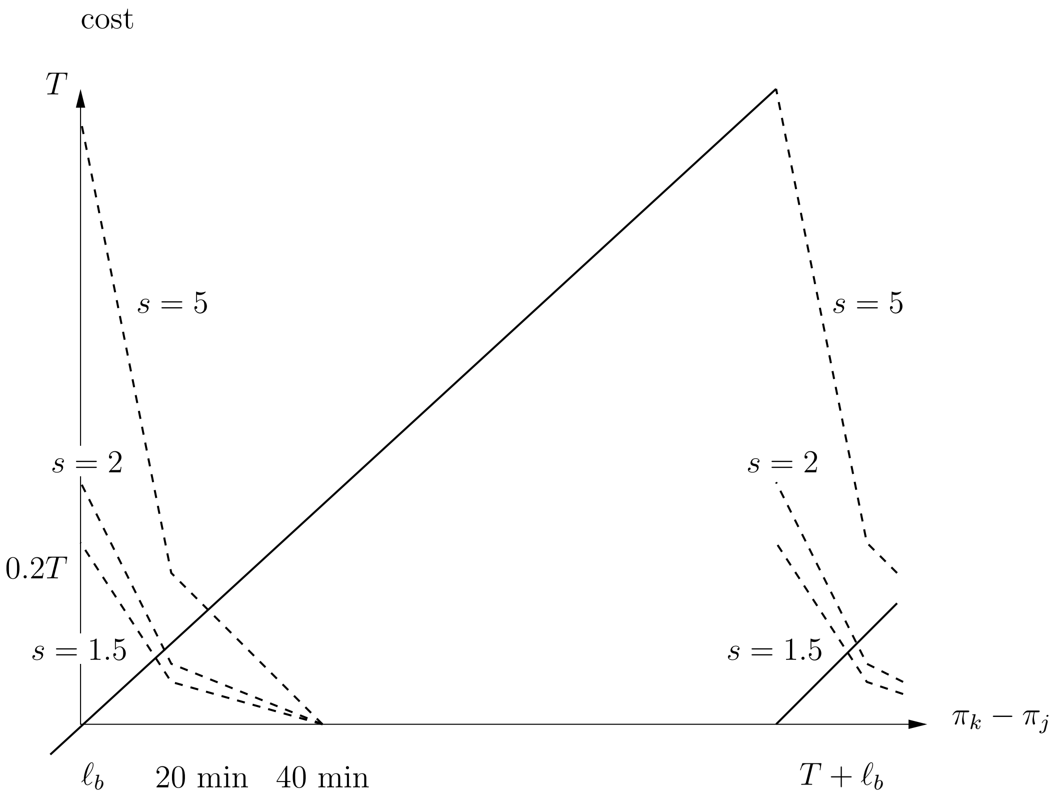

Figure 4: The nominal and some delay-cost functions

transfer time ℓ_b. The nominal cost function—the transfer times according to the published timetable—raises with slope equal to 1 through the whole period. The three dotted lines of the decreasing functions correspond to the three different delay-weighting factors s which we consider, 1.5, 2 and 5. Each one gives the fraction of all passengers that will in expectation miss their connection multiplied with s, if the planned transfer time is the x-value and the y-axis ranges from 0 to 1. Thus if you interpret the y-axis to range from 0 to T, as it is done for the graph of the nominal function, the three graphs give the delay-cost incurred for the three delay-weighting-factors $s \in \{1.5, 2, 5\}$. This is because every missed connection means one period, T minutes, of delay for the transferring passenger. Fix a factor s and a distribution, e.g., $s = 1.5$ and the distribution underlying the dotted lines in Figure 4, then the complete cost function before multiplying with arc weight w_b for an arc b is the sum of the nominal function and the function depicted by the lowest dotted line.

After one period all cost functions repeat themselves. Note that we have chosen the set of the delay-weighting factors s such that the non-continuous

jump of the summed cost function is negative for every of these factors, except for $s = 5$.
For each of the three weighting factors we consider three different distributions shown in Figure 7. The three delay penalty functions in Figure 4 are based on one of those, namely distribution C. We refrain from showing all ten objective functions, corresponding to three distributions, three delay-weighting factors s, plus the nominal function, in one complete unintelligible diagram.

3.2.4 Strict No-wait, Practical No-wait, and Light Robustness

To summarize, we make the following simplifications for calculating delay resistant timetables.

1. We assume a no-wait policy.
2. We even assume a strict no-wait policy.
3. We fix the scheduled driving times for our optimization.
4. We only count the delay of passengers that loose a connection, but not the delay of the last train of a passenger's trip.
5. We assume c_b to be convex.
6. We approximate c_b piecewise linear, i.e., we assume discrete distributions.

We commented on some of these issues earlier. Note that the normal, feasible no-wait policy in some railway networks is the only response that practitioners seem to have to the "repair game" [EGL05]:

> The management of the region south-west (of Deutsche Bahn AG) decided to apply a 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)

The strict no-wait policy leads to the following situation. The effect of a disturbance to the optimization problem is purely local. Yet, the solution that hedges against these effects optimally exploits the network structure. Therefore, our approach fits into the scheme of so-called Light Robustness.
Light Robustness [FM06] starts with calculating a conventional linear program that describes the deterministic optimization problem in question. Its

optimal objective value is stored as a reference value $c(x) = z$. Then a limited disturbance in data is added to the model in a worst case approach: A robust solution x' has to satisfy the linear program in all scenarios that may occur due to the limited change of data. Moreover, for a minimization problem the new solution must not exceed the old objective value by more than some $\alpha \geq 1$, i.e., $c(x') \leq \alpha z$. This double requirement may easily yield an infeasible instance. Therefore, Light Robustness allows the new solution x' to violate each of the robust counterparts of the original constraints. Denote the vector of violations by $\lambda(x')$. The optimization computes an x' that minimizes a usually non-linear function $f(\lambda(x'))$. Light Robustness minimizes the local violations.
Our solutions are lightly robust against the maximal driving time, which may reasonably occur in any scenario. A transfer that is not provided for the maximal driving time incurs a local punishment in our objective function, namely in the delay penalty. The quality guarantee α is what will be called in the result section the Price of Robustness.
We differ from standard Light Robustness in two respects. First, we incorporate the original objective into our new objective function. This means we allow for a trade-off between the nominal quality guarantee α, and the delay penalty, i.e., in terms of Light Robustness, the function f. Second, our model allows a precise understanding of the function f in terms of a weighting factor, distributions of driving times, and certain simpliying assumptions on delay propagation and delay management. We apply this simplification for being able to solve our model. The good news is that we can analyze our simplification.

3.3 Creating an Instance of the Delay Management Problem

Given a set $\underline{V} = \underline{V}_{arr} \cup \underline{V}_{dep}$ of periodic events and a set $\underline{A} = \underline{A}_{drive} \cup \underline{A}_{stop} \cup \underline{A}_{transfer} \cup \underline{A}_{turn} \cup \underline{A}_{head}$ of periodic activities together with lower and upper (periodic) bounds $\ell_a \leq u_a$ for all $a \in \underline{A}$, the methods of Section 3.2 construct timetables, which are the basis for solving the delay management problem. In particular, the input consists of

- the period $T(i)$ for each event $i \in \underline{V}$,
- the time $\pi_{first}(i)$ of the first occurrence, and
- the time $\pi_{last}(i)$ of the last occurrence for each periodic event $i \in \underline{V}$.

Note that for each activity $a = (i,j) \in \underline{A}_{drive} \cup \underline{A}_{stop} \cup \underline{A}_{turn} \cap \underline{A}_{transfer}$ we know that[2] $T(i) = T(j)$, and that

$$u_a - \ell_a < T(i). \tag{13}$$

Our goal is to expand the periodic network $\underline{D} = (\underline{V}, \underline{A})$ into a non-periodic network $D = (V, A)$. As input for the delay management problem, we need the scheduled time $\pi(i_k)$ for all $i_k \in V$, and we need the lower bounds ℓ_a for all non-periodic activities $a \in A$. Recall that in the case of delays, in the disposition timetable we may plan every driving time by 5% faster than it was fixed for the regular service timetable. Since in case of delays all activities are performed at the same speed or faster than initially planned, and since the original timetable π was feasible with respect to the upper bounds u_a, upper bounds need not be considered in the delay management problem.

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

In a non-periodic event-activity network without headway constraints, a directed edge $a = (i,j) \in A$ fixes the order of the two events i and j by requiring $\tilde{\pi}_j \geq \tilde{\pi}_i + \ell_a$. Such a network is cycle-free. In contrast, the non-periodic network $D = (V, A)$ we generate by driving the algorithm above contains directed cycles – for every headway edge $a = (i,j) \in A_{head}$, there exists another headway edge $a^{-1} = (j,i) \in A_{head}$. Analogously to a non-periodic

[2] In general, for $a = (i,j) \in \underline{A}_{transfer}$ one would have $T(i) \neq T(j)$, but the data that we use in our study are such that also for $\underline{A}_{transfer}$ we have $T(i) = T(j)$.

event-activity network without headway constraints, we can interpret these headway edges as follows: For each pair $(a = (i, j), a^{-1} = (j, i))$ of headway edges, either $\tilde{\pi}_i \geq \tilde{\pi}_j + \ell_{a^{-1}}$ or $\tilde{\pi}_j \geq \tilde{\pi}_i + \ell_a$ has to be satisfied – it is a part of solving (DM) to decide which of these two disjunctive constraints should be satisfied and which should be the one to be dropped. By solving (DM), we decide for each such pair (a, a^{-1}) if event i is scheduled before event j respecting the lower bound ℓ_a (we keep a and drop a^{-1}), or the other way round. When doing so, we have to keep exactly one headway edge from each such pair in such a way that the resulting network does not contain any directed cycle (see [Sch06a]).

3.4 Solving the Delay Management Problem

Using the notation of event-activity networks $D = (V, A)$, a *timetable* π is given by assigning a time π_i to each event $i \in V$. In the context of delay management, we are, however, interested in the disposition timetable, which will be called $\tilde{\pi}_i$, $i \in V$. We further use the numbers ℓ_a as the technical minimal necessary times for performing activity $a \in A$, and w_a as the number of passengers planning to use the connection $a \in A_{transfer}$. We also take T as the common period of all events.

Each of our delay scenarios is defined by a set of *source delays*, i.e., a set of events $V_{del} \subseteq V_{arr}$ such that $d_i > 0$ for all $i \in V_{del}$ and a set of activities $A_{del} \subseteq A$ such that $d_a > 0$. (For non-delayed events and activities we set $d_i = 0$, and $d_a = 0$, respectively.) Note that there is a basic difference between source-delayed events and source-delayed activities:

- A source delayed activity $a = (i, j)$ can arise due to an obstacle on the tracks or due to a detour within a station. It has to be added to a possible other delay of event i. I.e., if a train starts delayed at event i and there is an obstacle on the track, the additional delay has to be added to its departure delay.
- A source-delayed event i, however, need not be added to a possible other delay of event i. Source delayed events can be caused e.g. by a delayed train driver starting his shift in the station or by a track that is occupied due to repair work until a fixed point of time.

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

$$\tilde{\pi}_i = \text{actual time of event } i.$$

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, since no train is allowed to start earlier than planned. For all transfer activities $a \in A_{transfer}$ we introduce

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

and for $a = (i,j) \in A_{head}$ 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 [Sch06c, Sch06a]).

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

such that

$$\begin{aligned}
\tilde{\pi}_i &\geq \pi_i + d_i \quad \text{for all } i \in V_{del} && (14)\\
\tilde{\pi}_j - \tilde{\pi}_i &\geq \ell_a + d_a \quad \text{for all } a = (i,j) \in A_{stop} \cup A_{drive} \cup A_{turn} && (15)\\
M z_a + \tilde{\pi}_j - \tilde{\pi}_i &\geq \ell_a \quad \text{for all } a = (i,j) \in A_{transfer} && (16)\\
M g_a + \tilde{\pi}_j - \tilde{\pi}_i &\geq \ell_a \quad \text{for all } a = (i,j) \in A_{head} && (17)\\
g_a + g_{a^{-1}} &= 1 \quad \text{for all } a, a^{-1} \in A_{head} && (18)\\
\tilde{\pi}_i &\in \mathbb{N} \quad \text{for all } i \in V\\
z_a &\in \{0,1\} \quad \text{for all } a \in A_{transfer}
\end{aligned}$$

Note that w_i are positive weights that represent the importance of event i. The first constraint (14) makes sure that no train departs earlier as scheduled, while (15) ensures that the delay is carried over correctly from one event to the next. Recall that in (15) the lower bound is by 5% smaller than the driving times that were used for the computation of the regular service timetable. Constraints (14) and (15) additionally ensure that the source delays d_i and d_a, respectively, are taken into account. In particular, if event i takes place at some time point $\tilde{\pi}_i$, event j must be later than $\tilde{\pi}_i + \ell_a$ if $a = (i,j)$ is the activity linking i and j. If $z_a = 0$, constraint (16) ensures that the delay is carried over for each maintained connection. For $z_a = 1$, however, constraint (16) becomes redundant whenever M is large enough. In particular, we need that $M \geq \max_{i \in V} \tilde{\pi}_i - \pi_i$ is large enough. Since the computation time allowed it, we used $M = H + 2T$ and are hence sure

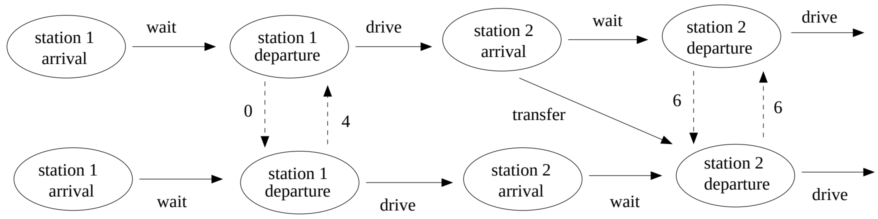

Figure 5: Graphical interpretation of the capacity constraints

that M is large enough. Constraints (14) to (16) 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 (17) and (18). These constraints can be equivalently reformulated to

$$\text{(17) and (18)} \iff \tilde{\pi}_j - \tilde{\pi}_i \geq \ell_a \text{ or } \tilde{\pi}_i - \tilde{\pi}_j \geq \ell(a^{-1}).$$

Hence, (17) and (18) require for all $(i, j) \in A_{head}$ 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. If one edge of each pair is chosen, the order of the events is fixed, and at the same time the security distances, indicated as bounds ℓ_a of edge $a = (i, j)$, are respected. Recall that the event-activity network without dotted edges is cycle-free. When fixing the order of the events, one has to choose one edge from each pair of dotted edges in such a way that the resulting network also 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 $a \in A$ 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 (for a definition of the never-meet property and its satisfaction in practice, see [Sch06c]). To this end, w_i has to be chosen as the number of passengers with final destination i.
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 [GSar, HdV01].

In our case study we minimize the objective function mentioned above, but in the evaluation we also calculate the following characteristics of the disposition plan found:

- objective value
- number and percentage of missed connections
- number and percentage of passengers who missed a connection
- number and percentage of delayed arrival events
- sum of all delays over all arrival events.

Solution approach. If the z_a variables and the g_a variables have been fixed, the remaining problem can be easily solved by the forward phase of the critical path method (CPM). To this end, we assume that the events are ordered according to the scheduled times π_i, i.e. in their "natural order". Then we set

$$\begin{aligned} A^{relevant} &:= A_{stop} \cup A_{drive} \cup A_{turn} \\ &\quad \cup \{a \in A_{transfer} : z_a = 0\} \cup \{a \in A_{head} : g_a = 0\} \end{aligned}$$

and obtain the optimal disposition timetable $\tilde{\pi}$ through

$$\begin{aligned} \tilde{\pi}_1 &:= \pi_1 + d_1 \\ \tilde{\pi}_i &:= \max\{\pi_i + d_i, \max_{a=(j,i)\in A} \tilde{\pi}_j + \ell_a\}, \quad i = 2, \ldots, |V|. \end{aligned}$$

Usually, the wait-depart decisions z_a and the order in which the trains use common capacities (represented by the variables g_a) are not known. In this case several heuristics may be used, see [SS07]. 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 since we limited our calculations to an observation period of some hours and since within the Harz region, there are only few conflicts with the never-meet property in most of our delay scenarios.

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. The northern

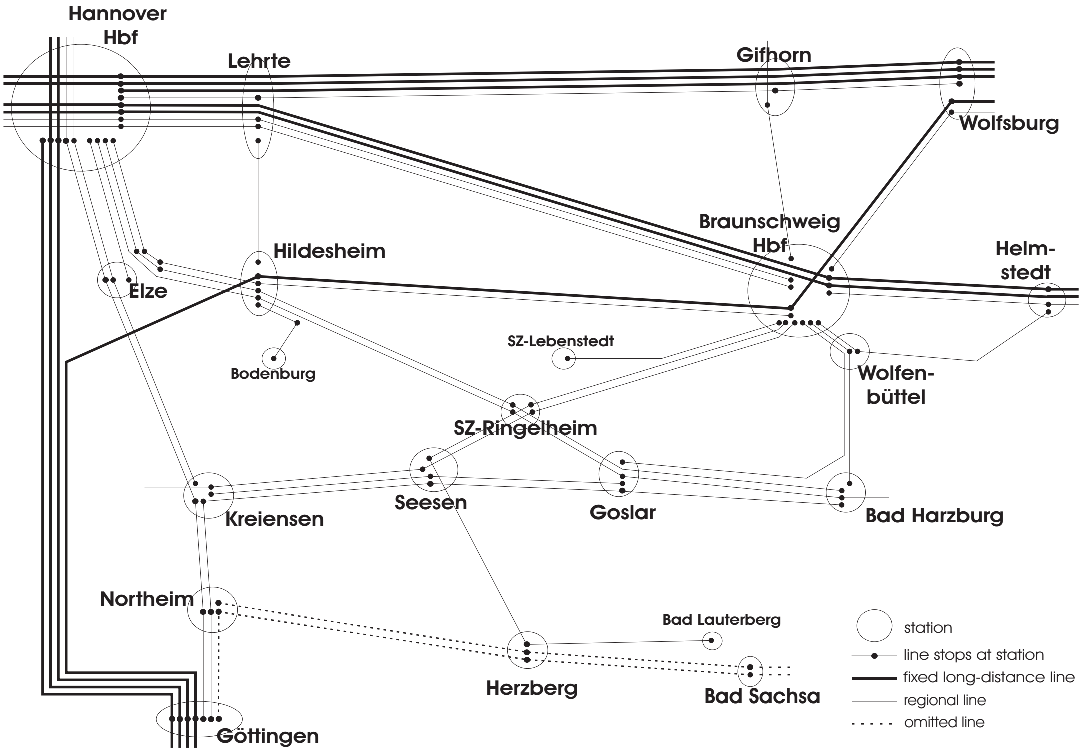

Figure 6: The railway lines in the Harz Region of the Lower Saxony part of the German Railway network

and western boundary of this region are the tracks Wolfsburg–Hannover and Hannover–Göttingen, respectively.
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 take in this first study on optimization and recovery of railway timetables 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 scenario that was kindly provided to us by Deutsche Bahn AG. This way, we are sure that the optimized timetables that we are about to compute for the regional service lines can be embedded into a realistic scenario for Germany as a whole.

In total, our scenario 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 which we compute will 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–Braunschweig–Hildesheim–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. The weights represent the number of passengers who use the transfers, and they were estimated by Deutsche Bahn AG using their so-called traffic model.
In addition, note that when computing the periodic regular service timetable we keep the driving times of the trains unchanged, for performance reasons. Yet, we allow for 26 stop activities their minimum dwell time to be extended by several 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 [LM04, Lie06] for any details.

In Table 1 we report the size of the resulting network $\underline{D} = (\underline{V}, \underline{A})$. Notice that for the purpose of periodic timetabling, many 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 [Lie06] for details. Unfortunately, in the contracted network the information becomes highly aggregated, and thus is no more suited to derive the non-periodic network for the delay management problem. This is why 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, too.
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 $a \in A$. As the only information we have on the number of passengers are the weights $\tilde{w}_a$ of the transfer activities $a \in A_{transfer}$ 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 $a \in A_{transfer}$ where $\overline{w}$ denotes the arithmetic mean of the weights $\tilde{w}_a$ of all transfer activities.

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
$\lvert\underline{A}_{trans}\rvert$	or	A_{trans}	182	–	≈ 500
$\lvert\underline{A}_{head}\rvert$	or	A_{head}	454	–	≈ 11000

For delay resistant timetabling we consider three distributions of source delays, described in Table 2 and shown in Figure 7. In that table P_{ontime} denotes the probability that a train needs for a certain driving arc 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. Thereby 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%.

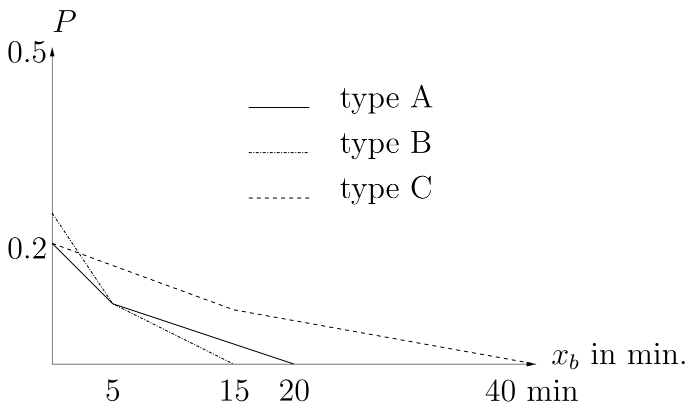

Figure 7: The three Boundary Distributions for the Probability to Miss a Connection Scheduled with tension x_b on transfer arc b.

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

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 then reduce the non-periodic network (V, A) as follows:

- generate a list $\mathscr{T}$ of all trains τ arriving at or departing from one of the five central stations within the observation period H
- for each non-periodic event $i \in V$
  - if i belongs to a train $\tau \in \mathscr{T}$: keep i
  - otherwise: delete i from V, $V = V \setminus \{i\}$
- for each non-periodic activity $a = (i, j) \in A$
  - if $i \in V$ and $j \in V$: keep a
  - otherwise: delete a from A, $A = A \setminus \{a\}$.

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 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 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.

Analyzing the results the following questions are of central interest. Is it possible to reduce the expected delay by means of delay resistant timetabling in a realistic setting? How much nominal travel time of the passengers has to be sacrificed to obtain a certain level of robustness? Is it possible to reduce the passengers' delay by means of delay management? Is the theoretical robustness of the delay resistant timetables echoed by their actual behavior under realistic delay management?

To answer these questions we simulate the nominally optimal and the delay resistant timetables under an optimal delay management as well as under a feasible no-wait policy.

It turned out that delay resistant timetabling reduces the expected delay substantially both under no-wait and under optimal delay management. This success is achieved at the expense of a clearly small increase in the nominal travel times. Second, for both nominal and delay resistant timetabling the improvement when applying optimal delay management instead of the no-wait policy is significant. Finally, some more detailed observations were achieved in the theoretical evaluation used in the optimization, which are justified by the practical evaluation based on the simulation.

The optimization of the timetables has been obtained by solving PESP-IP-x-z, with refined objective function, by CPLEX 10.1 on a 3GHz 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*

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

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'_{15C}	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 compiled these results. To compare the different delay resistant plans, we also show which delay penalties each plan incurs in the objective function of 15C, i.e., we give $C'_{15C}(ID)$ which is by constant factor equal to $C'_{2C}(ID)$ and $C'_{5C}(ID)$. The choice of these functions will be become clear by the closer analysis. As a first hint, observe that the plans 15C and 5C achieve maximum and minimum PoR among all delay resistant timetables. Furthermore, the table gives the CPU time in seconds until optimality was established. The computation time for the nominal timetable can be reduced to less than half an hour by a further acceleration technique, that is not ready yet to use for the delay resistant models. 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 random scenarios. The last column gives the 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. For the average of the simulated, feasible no-wait policy we also give in brackets the number of missed connections in normalize with respect to the corresponding number in the default plan with optimal delay management.

Qualitatively, the results of the theoretical evaluation by different objective functions are echoed by both the simulation under the no-wait and under an optimal delay management policy. Again a certain margin of imprecision has to be granted because a simulation takes into account a relatively small number of scenarios. Recall that the samples correspond to distribution A. The data supports that our carefully simplified objective function is a good direction for optimization to balance delay resistance and nominal travel time in timetabling, particularly so for the short source delays, that haunt everyday railway business.
The normalization in Table 3 disguises the improvements of the delay management. The ratio between absolute values for missed connections of the feasible no-wait policy and the optimal delay management is 2.18. This means, that even the most delay resistant plan 5C under no-wait delay management has 4.5% more missed connections than the nominally optimal plan under optimal delay management. In other words, the most resistant plan (5C) together with the feasible no-wait policy work almost as well as optimal delay management with the nominal plan.
We have chosen the normalization nevertheless, because it reveals that the qualitative assessment of a timetable by the simplified objective function comply with the results of both delay management approaches. Interestingly, the optimal delay management even in these relative terms is always better than the no-wait policy. Optimal delay management makes better use of the possibilities that delay resistant timetables offer.
Looking closer, it is most interesting how changes in the distribution or changes in the delay-weighting factor s influence the features of the resulting timetables.
The PoR seems a lot more susceptible to the weighting factor than to the underlying distribution. Moreover, distribution C, which is the one with many long source delays, for small weighting factors (1.5 and 2) has a smaller or 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 tailor made for A than for C. Second, the unexpected behavior vanishes, when the weighting factor is lifted to 5. Then the

optimizer pays respect to 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. To protect 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. The plan 15C does not even have the lowest delay penalty in his own setting. But it has the lowest nominal value among all delay resistant plans. Therefore, in sum it is the best plan for his setting. In contrast consider the plan 5C, which is also the best plan for his 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.
Also the RoD increases with increasing weighting factors—at least for distribution A and B. Under distribution C the RoD reacts only at high weighting factors. The main difference between the behavior of the PoR and that of the RoD lies in the susceptibility to distributions. The RoD of the B plans is clearly higher than that of A plan with the same weighting factor. For distribution C always the lowest RoD values are achieved. This again backs the interpretation that by means of timetabling one can do more against short source delays than against long source delays.

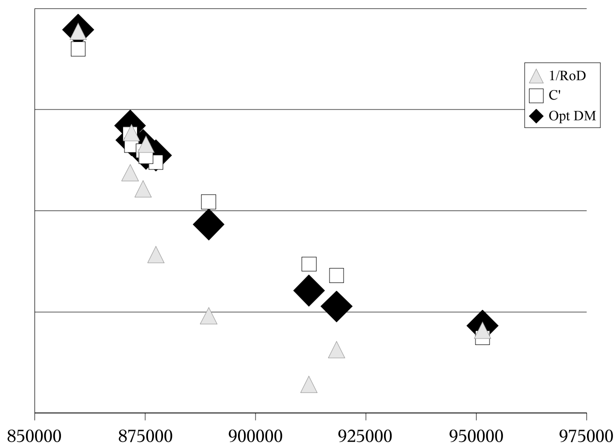

Figure 8: Simplified Objective Matches Optimal Delay Management

In fact, the delay penalty used in our simplified objective function provides a surprisingly adequate estimate of the actual delay penalty, namely the number of missed connections under optimal delay management. Note that Figure 8 is obtained from the original values of the simplified delay penalty

with respect to distribution C only by multiplication with a constant. The ten plans are ordered according to their nominal objective value. This result encouragingly justifies to use our simplified objective function for delay resistant timetabling.
For the quantitative results one should look at the simulation average. There it shows that 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, take timetable 2B. Compared to the optimum nominal timetable, it increases the nominal objective by only 3%. But this halves the number of passengers who miss a connection! Nevertheless, one might also want to take into account the different absolute levels of these two conflicting objective functions. Then, it turns out that the gain that we obtain with respect to the delay penalty outweighs the increase in nominal waiting time, as long as the delay-weighting factor is chosen to at least ≈ 1.5.
Note that the samples, the "reality" of our study if you like, were drawn according to distribution A. Nevertheless, it may well be that a different distribution—distribution B in this case—yields the more desirable approximation.

Recommended workflow. Here, it is time to recall what we were initially aiming at: Computing a periodic timetable that allows for concise actions in delay management during its operation. Hence, the key is to roughly approximate the involved delay management objective function. In other words, we are interested in identifying an add-on to the standard periodic timetabling objective function with two major properties: (i) being easy to evaluate; (ii) provide a cost value for each nominal timetable that is much similar to the actual costs of the timetable under small disturbances.
We recommend the following workflow. First, compute several distinct timetables, which should put different emphasize on nominal waiting time and the risk of missed connections, or on cost efficiency. Second, evaluate our simple delay penalty functions on any of these timetables, for different distributions for the delays. Third, simulate their behavior by confronting any of these timetables with a sample set of scenarios, and compute optimum disposition timetables. Fourth, calibrate the simplified objective function for each distribution via the delay-weighting factor s to best fit the cost observed in the simulation. Fifth, select the distribution and delay-weighting factor that fits best. Now, compute with this delay penalty function the delay resistant periodic timetable. For the last step, the delay-weighting factor s may be scaled according to the level of risk aversion the management targets at.

6 Conclusion

We successfully added delay resistancy to the computation of periodic railway timetables. 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, because in our slight extension of the "Light Robustness" approach we did not add any integer variables to the well-established integer programs for periodic timetabling. Nevertheless, further research should aim at bringing the computation times of the refined model even closer to the standard model. In particular, solving the delay management problem serves as an accurate measure for estimating the delay resistancy of a 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.

7 Acknowledgment

We thank Jens Dupont, Robert Firla, and Frank Geraets, Deutsche Bahn AG, for inspiring parts of this research and for providing us with the data necessary to perform our case study.

References

[BGJ+05] 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).

[CS07] C. Conte and A. Schöbel. Identifying dependencies among delays. Technical report, Institut für Numerische und Angewandte Mathematik, Universität Göttingen, 2007. To appear at IAROR 2007.

[DV95] Joachim R. Daduna and Stefan Voß. Practical experiences in schedule synchronization. In Joachim R. Daduna, Isabel Branco, and José M. Pinto Paixao, editors, *CASPT*, volume 430 of *Lecture Notes in Economics and Mathematical Systems*, pages 39–55. Springer, 1995.

[EFK04] Ophelia Engelhardt-Funke and Michael Kolonko. Analysing stability and investments in railway networks using advanced evolutionary algorithms. *International Transactions in Operational Research*, 11:381–394, 2004.

[EGL05] Jan Ehrhoff, Sven Grothklags, and Ulf Lorenz. Parallelism for perturbation management and robust plans. In José C. Cunha and Pedro D. Medeiros, editors, *Euro-Par*, volume 3648 of *Lecture Notes in Computer Science*, pages 1265–1274. Springer, 2005.

[FM06] M. Fischetti and M. Monaci. Robust optimization through branch-and-price. In *Proceedings of AIRO*, 2006.

[GGJ+04] 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.

[GHL06] 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.

[GJPS05] 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.

[GJPW07] 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.

[GSar] A. Ginkel and A. Schöbel. To wait or not to wait? the bicriteria delay management problem in public transportation. *Transportation Science*, to appear.

[HdV01] B. Heidergott and R. de Vries. Towards a control theory for transportation networks. *Discrete Event Dynamic Systems*, 11:371–398, 2001.

[KDV05] L.G. Kroon, R. Dekker, and M. Vromans. Cyclic railway timetabling: A stochastic optimization approach. In *Algorithmic Methods for Railway Opti-*

mization, 2005. To appear. Preprint available at `http://www.few.eur.nl/few/research/ecopt/publications`.

[KP03] Leo G. Kroon and Leon W.P. Peeters. A variable trip time model for cyclic railway timetabling. *Transportation Science*, 37:198–212, 2003.

[KS88] Wolf-Dieter Klemt and Wolfgang Stemme. Schedule synchronization for public transit networks. In Joachim R. Daduna and Anthony 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.

[Lie05a] Christian Liebchen. Der Berliner U-Bahn Fahrplan 2005 – Realisierung eines mathematisch optimierten Angebotskonzeptes. In *HEUREKA '05: Optimierung in Transport und Verkehr, Tagungsbericht*, number 002/81. FGSV Verlag, 2005. In German.

[Lie05b] Christian Liebchen. Fahrplanoptimierung im Personenverkehr—Muss es immer ITF sein? *Eisenbahntechnische Rundschau*, 54(11):689–702, 2005. In German.

[Lie06] C. Liebchen. *Periodic Timetable Optimization in Public Transport*. dissertation.de – Verlag im Internet, Berlin, 2006.

[Lin00] Thomas Lindner. *Train Schedule Optimization in Public Rail Transport*. Ph.D. thesis, Technische Universität Braunschweig, 2000.

[LM04] Christian Liebchen and Rolf H. Möhring. The modeling power of the periodic event scheduling problem: Railway timetables – and beyond. Preprint 020/2004, TU Berlin, Mathematical Institute, 2004. To appear in Springer LNCS Volume Algorithmic Methods for Railway Optimization. An extended abstract has been accepted for publication in Proceedings of the Ninth International Workshop on Computer-Aided Scheduling of Public Transport (CASPT).

[LP02] Christian Liebchen and Leon W.P. Peeters. On cyclic timetabling and cycles in graphs. Technical Report 761-2002, TU Berlin, Mathematical Institute, 2002.

[LPW05] Christian Liebchen, Mark Proksch, and Frank H. Wagner. Performance of algorithms for periodic timetable optimization. In Mark Hickman, editor, *Computer-Aided Transit Scheduling— Proceedings of the Ninth International Workshop on Computer-Aided Scheduling of Public Transport (CASPT)*, Lecture Notes in Economics and Mathematical Systems. Springer, 2005. To appear.

[LR07] Christian Liebchen and Romeo Rizzi. Classes of cycle bases. *Discrete Applied Mathematics*, 155(3):337–355, 2007.

[LS06] C. Liebchen and S. Stiller. Delay resistant timetabling. Technical Report 2006/24, Technische Universität Berlin, 2006. presented at CASPT'06.

[Nac98] Karl Nachtigall. *Periodic Network Optimization and Fixed Interval Timetables.* Habilitation thesis, Universität Hildesheim, 1998.

[Odi96] Michiel A. Odijk. A constraint generation algorithm for the construction of periodic railway timetables. *Transportation Research B*, 30(6):455–464, 1996.

[RdVM98] B. De Schutter R. de Vries and B. De Moor. On max-algebraic models for transportation networks. In *Proceedings of the International Workshop on Discrete Event Systems*, pages 457–462, Cagliari, Italy, 1998.

[Roc84] R. Tyrrell Rockafellar. *Network flows and monotropic optimization.* John Wiley & Sons, Inc., 1984.

[SBK01] 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.

[Sch01] A. Schöbel. A model for the delay management problem based on mixed-integer programming. *Electronic Notes in Theoretical Computer Science*, 50(1), 2001.

[Sch06a] A. Schöbel. Capacity constraints in delay management. In *CASPT 06*, June 2006.

[Sch06b] A. Schöbel. *Customer-oriented optimization in public transportation*, volume 3 of *Optimization and Its Applications*. Springer, New York, 2006.

[Sch06c] A. Schöbel. Integer programming approaches for solving the delay management problem. *Lecture Notes in Computer Science*, 2006. to appear.

[SM97] L. Suhl and T. Mellouli. Supporting planning and operation time control in transportation systems. In *Operations Research Proceedings 1996*, pages 374–379. Springer, 1997.

[SM99] L. Suhl and T. Mellouli. Requirements for, and design of, an operations control system for railways. In *Computer-Aided Transit Scheduling*. Springer, 1999.

[SMBG01] 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.

[SS94] Alexander Schrijver and Adri G. Steenbeek. Dienstregelingontwikkeling voor Railned. Rapport CADANS 1.0, Centrum voor Wiskunde en Informatica, December 1994. In Dutch.

[SS07] M. Schachtebeck and A. Schöbel. Algorithms for the delay management problem. Technical report, Georg-August Universität Göttingen, 2007. working paper.

[SU89] Paolo Serafini and Walter Ukovich. A mathematical model for periodic scheduling problems. *SIAM Journal on Discrete Mathematics*, 2(4):550–581, 1989.

[Wag04] Udo Wagner. Pünktliche Abfahrt verhindert Domino-Effekt auf dem Gleis. BAHNZEIT *(employee newsletter of Deutsche Bahn AG)*, 5:4, 2004. In German.

Reports from the group

"Combinatorial Optimization and Graph Algorithms"

of the Department of Mathematics, TU Berlin

2007/22 *Michael Elkin and Christian Liebchen and Romeo Rizzi:* New Length Bounds for Cycle Bases

2007/19 *Nadine Baumann and Sebastian Stiller:* The Price of Anarchy of a Network Creation Game with Exponential Payoff

2007/17 *Christian Liebchen and Michael Schachtebeck and Anita Schöbel and Sebastian Stiller and André Prigge:* Computing Delay Resistant Railway Timetables

2007/03 *Christian Liebchen and Gregor Wünsch and Ekkehard Köhler and Alexander Reich and Romeo Rizzi:* Benchmarks for Strictly Fundamental Cycle Bases

2006/32 *Romeo Rizzi and Christian Liebchen:* A New Bound on the Length of Minimum Cycle Bases

2006/24 *Christian Liebchen and Sebastian Stiller:* Delay Resistant Timetabling

2006/08 *Nicole Megow and Tjark Vredeveld:* Approximation Results for Preemptive Stochastic Online Scheduling

2006/07 *Ekkehard Köhler and Christian Liebchen and Romeo Rizzi and Gregor Wünsch:* Reducing the Optimality Gap of Strictly Fundamental Cycle Bases in Planar Grids

2006/05 *Georg Baier and Thomas Erlebach and Alexander Hall and Ekkehard Köhler and Heiko Schilling:* Length-Bounded Cuts and Flows

2005/30 *Ronald Koch and Martin Skutella and Ines Spenke :* Maximum k-Splittable Flows

2005/29 *Ronald Koch and Ines Spenke :* Complexity and Approximability of k-Splittable Flows

2005/28 *Stefan Heinz and Sven O. Krumke and Nicole Megow and Jörg Rambau and Andreas Tuchscherer and Tjark Vredeveld:* The Online Target Date Assignment Problem

2005/18 *Christian Liebchen and Romeo Rizzi:* Classes of Cycle Bases

2005/11 *Rolf H. Möhring and Heiko Schilling and Birk Schütz and Dorothea Wagner and Thomas Willhalm:* Partitioning Graphs to Speed Up Dijkstra's Algorithm.

2005/07 *Gabriele Di Stefano and Stefan Krause and Marco E. Lübbecke and Uwe T.Zimmermann:* On Minimum Monotone and Unimodal Partitions of Permutations

2005/06 *Christian Liebchen:* A Cut-based Heuristic to Produce Almost Feasible Periodic Railway Timetables

2005/03 *Nicole Megow, Marc Uetz, and Tjark Vredeveld:* Models and Algorithms for Stochastic Online Scheduling

2004/37 *Laura Heinrich-Litan and Marco E. Lübbecke:* Rectangle Covers Revisited Computationally

2004/35 *Alex Hall and Heiko Schilling:* Flows over Time: Towards a more Realistic and Computationally Tractable Model

2004/31 *Christian Liebchen and Romeo Rizzi:* A Greedy Approach to Compute a Minimum Cycle Bases of a Directed Graph

2004/27 *Ekkehard Köhler and Rolf H. Möhring and Gregor Wünsch:* Minimizing Total Delay in Fixed-Time Controlled Traffic Networks

2004/26 *Rolf H. Möhring and Ekkehard Köhler and Ewgenij Gawrilow and Björn Stenzel:* Conflict-free Real-time AGV Routing

2004/21 *Christian Liebchen and Mark Proksch and Frank H. Wagner:* Performance of Algorithms for Periodic Timetable Optimization

2004/20 *Christian Liebchen and Rolf H. Möhring:* The Modeling Power of the Periodic Event Scheduling Problem: Railway Timetables — and Beyond

2004/19 *Ronald Koch and Ines Spenke:* Complexity and Approximability of k-splittable flow problems

2004/18 *Nicole Megow, Marc Uetz, and Tjark Vredeveld:* Stochastic Online Scheduling on Parallel Machines

2004/09 *Marco E. Lübbecke and Uwe T. Zimmermann:* Shunting Minimal Rail Car Allocation

2004/08 *Marco E. Lübbecke and Jacques Desrosiers:* Selected Topics in Column Generation

2003/050 *Berit Johannes:* On the Complexity of Scheduling Unit-Time Jobs with OR-Precedence Constraints

2003/49 *Christian Liebchen and Rolf H. Möhring:* Information on MIPLIB's timetab-instances

2003/48 *Jacques Desrosiers and Marco E. Lübbecke:* A Primer in Column Generation

2003/47 *Thomas Erlebach, Vanessa Kääb, and Rolf H. Möhring:* Scheduling AND/OR-Networks on Identical Parallel Machines

2003/43 *Michael R. Bussieck, Thomas Lindner, and Marco E. Lübbecke:* A Fast Algorithm for Near Cost Optimal Line Plans

2003/42 *Marco E. Lübbecke:* Dual Variable Based Fathoming in Dynamic Programs for Column Generation

2003/37 *Sándor P. Fekete, Marco E. Lübbecke, and Henk Meijer:* Minimizing the Stabbing Number of Matchings, Trees, and Triangulations

2003/25 *Daniel Villeneuve, Jacques Desrosiers, Marco E. Lübbecke, and François Soumis:* On Compact Formulations for Integer Programs Solved by Column Generation

2003/24 *Alex Hall, Katharina Langkau, and Martin Skutella:* An FPTAS for Quickest Multicommodity Flows with Inflow-Dependent Transit Times

2003/23 *Sven O. Krumke, Nicole Megow, and Tjark Vredeveld:* How to Whack Moles

2003/22 *Nicole Megow and Andreas S. Schulz:* Scheduling to Minimize Average Completion Time Revisited: Deterministic On-Line Algorithms

2003/16 *Christian Liebchen:* Symmetry for Periodic Railway Timetables

2003/12 *Christian Liebchen:* Finding Short Integral Cycle Bases for Cyclic Timetabling

Reports may be requested from:

Sekretariat MA 6–1
Fakultät II – Institut für Mathematik
TU Berlin
Straße des 17. Juni 136
D-10623 Berlin – Germany

e-mail: klink@math.TU-Berlin.DE

Reports are also available in various formats from

`http://www.math.tu-berlin.de/coga/publications/techreports/`

and via anonymous ftp as

`ftp://ftp.math.tu-berlin.de/pub/Preprints/combi/Report-`*number*`-`*year*`.ps`