View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Erasmus University Digital Repository

Railway Timetabling from an Operations Research Perspective

Leo Kroon[1,3], Dennis Huisman[2,3], Gábor Maróti[1]*

[1] Rotterdam School of Management
Erasmus Center for Optimization in Public Transport (ECOPT)
Erasmus University Rotterdam, P.O. Box 1738
NL-3000 DR Rotterdam, The Netherlands

[2] Econometric Institute
Erasmus Center for Optimization in Public Transport (ECOPT)
Erasmus University Rotterdam, P.O. Box 1738
NL-3000 DR Rotterdam, The Netherlands

[3] Department of Logistics, NS Reizigers,
P.O. Box 2025, NL-3500 HA Utrecht, The Netherlands

E-mail: lkroon@rsm.nl, huisman@few.eur.nl, gmaroti@rsm.nl

Econometric Institute Report EI2007-22

Abstract

In this paper we describe Operations Research (OR) models and techniques that can be used for determining (cyclic) railway timetables. We discuss the two aspects of railway timetabling: (*i*) the determination of arrival and departure times of the trains at the stations and other relevant locations such as junctions and bridges, and (*ii*) the assignment of each train to an appropriate platform and corresponding inbound and outbound routes in every station. Moreover, we discuss robustness aspects of both subproblems.

1 Introduction

In this chapter we describe Operations Research (OR) models and techniques that can be used for determining (cyclic) railway timetables. A railway timetable consists of two elements: (*i*) the arrival and departure times

*This author is sponsored by the Future and Emerging Technologies Unit of EC (IST priority 6th FP), under contract no. FP6-021235-2 (ARRIVAL).

of the trains at the stations and other relevant locations such as junctions and bridges, and (*ii*) the assignment of each train to an appropriate platform and corresponding inbound and outbound routes in every station.

Preferably, the generation of the arrival and departure times and the selection of the routes through the stations are carried out simultaneously. However, when using a model based approach for generating a timetable, this would lead to models that are too large to be solved by the currently available optimization technology. Therefore, in a model based approach the two steps are commonly split: the timetable is computed first, and then the routings through the stations for this particular timetable are determined. To get an overall feasible timetable, several iterations of this process may be necessary. In this paper, we treat both aspects separately.

Of course, timetabling is not the only planning problem faced by a railway operator. One also has to deal with rolling stock and crew (see Figure 1). For a complete overview on the application of OR models in passenger railway transportation, we refer to Huisman et al. (2005).

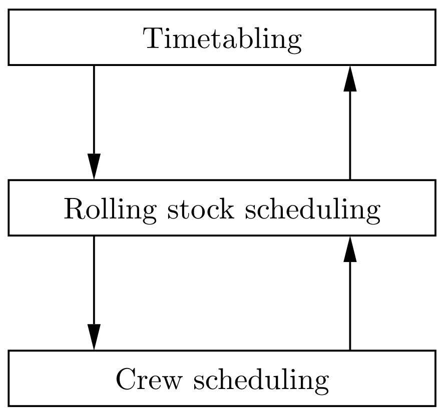

Figure 1: Planning problems

Planning problems within the railway world are traditionally solved based on the experience and the craftsmanship of the involved railway planners. However, in the nineties of the previous century, it was recognized within the railway world that the application of OR models and techniques for supporting the solution process of these problems may be beneficial along various dimensions. On one hand, the application of such models and techniques may lead to better solutions, and on the other hand it may lead to a reduction in the throughput time of the involved planning processes.

This transition was stimulated from two directions. First, at least in Europe, railway companies are more and more confronted with competition, due to recent changes in European legislation. Several railway companies have to operate on a commercial or semi-commercial basis, which puts more pressure on efficiency and operational excellence. The application of mathematical optimization models may help in realizing these objectives. Second, the scientific community has spent a lot of effort to model the railway planning problems in the most appropriate way. Together with new optimization techniques - often from the field of mathematical programming - and increased computing power of the available hardware, those planning problems can nowadays be solved in a reasonable amount of computing time. Twenty years ago this was not possible for many railway planning problems.

In this paper, we discuss the most important contributions in the field of model based timetabling, which form the basis for the advanced tools used by several railway operators nowadays. Moreover, we discuss an important research direction in the field of timetabling, namely timetables that are as *robust* as possible, i.e. timetables that have been constructed in such a way that small disturbances during the operations have only a limited impact.

The structure of this paper is as follows. Section 2 describes mathematical optimization models for generating cyclic railway timetables. The most commonly used model is explained in detail with an example. Thereafter, in Section 3, we discuss how the robustness of a cyclic timetable can be improved. Furthermore, results of an experiment in practice with such a robust timetable are reported. Section 4 considers the other aspect of timetabling, namely the routing of trains through railway stations. Robustness issues are discussed here as well. Finally, in Section 5 we conclude this paper with some perspectives on further developments.

2 Cyclic Timetabling

In many European countries, the passenger trains are operated according to a *cyclic* or a *periodic* timetable. This means that each line of passenger trains is operated in a cyclic pattern, e.g. the trains of the line run every 30, 60 or 120 minutes. Furthermore, if the passenger trains are operated according to a cyclic timetable, then it is usual that the timetable that is planned for the cargo trains is cyclic as well. However, in that case usually not all planned time slots for the cargo trains are actually used in the operations.

One advantage of a cyclic timetable is that passengers can easily keep in mind the departure time of their train at their station. Furthermore, it is relatively easy to set up a large number of transfer possibilities for passengers. However, a drawback is the fact that in a cyclic timetable it is difficult to offer a large number of direct connections, since a cyclic timetable is not quite flexible. Another drawback is the fact that a completely cyclic timetable may be rather inefficient: trains may have to be operated even at times with only a small number of passengers. Therefore, in practice there are usually exceptions to the completely cyclic timetable. For example, there may be some additional trains during rush hours, and frequencies may be reduced during late evening hours. Only a few European countries (e.g.

France and Spain) do not have any cyclicity or regularity in their timetables. In this paper, we do not discuss models and solution approaches for this kind of timetables. We refer the interested reader to Caprara et al. (2002).

In this paper, the infrastructure is assumed to be given, both *between* the stations and *within* the stations. Although often seen by the general public as part of the timetabling process, a line plan is assumed to be known a priori. We use the following definition of a *line*: a line is a direct railway connection between two end stations that is operated with a certain frequency and a certain train type, e.g. Intercity or stoptrain. As was indicated earlier, the timetabling step of determining arrival and departure times of the trains at the stations and the step of routing the trains trough the stations are considered separately in this paper. The first step is described in this section and in Section 3, and the latter step is described in Section 4.

2.1 A mathematical formulation

Most cyclic timetabling models are based on the Periodic Event Scheduling Problem (PESP), initially developed by Serafini and Ukovich (1989). The PESP model aims at cyclically scheduling a number of events $e = 1, \ldots, E$. The cycle time of a timetable is denoted by T. Thus, if an event takes place at time instant v, then a similar event takes place at all time instants $\{\ldots, v - 2T, v - T, v, v + T, v + 2T, \ldots\}$.

Note that the PESP model is a pure scheduling model and not a routing model. Thus, if there are several options for routing a train at the tracks *between* the stations, then it is assumed that a selection for one of these options has been made a priori. For example, if there are 4 parallel tracks between two stations (that is, 2 in each direction), then for each train a selection for one of the possible tracks has been made already.

In the case of a railway timetable, the events are the arrivals and departures of the trains at the stations and at the other relevant locations such as junctions: the decision variable v_e denotes the time at which event e is scheduled within each cycle. It is common that all event times are integer. Then v_e should have an integer value in the interval $[0, T - 1]$, and all computations are carried out modulo the cycle time T.

The event times are the start and end times of certain processes in the timetable. For example, trains have to run from one station to another, they have to dwell for a certain period of time in a station, there has to be a certain headway time between two consecutive trains crossing the same part of the infrastructure, two trains have to be split or combined, or they have a passenger or a rolling stock connection, etc. Thus there are processes related to single trains and processes related to pairs of trains.

The set of processes is denoted by P. If a process starts with event e and ends with event e', then this process is denoted by (e, e'). PESP assumes that for each process (e, e') a lower bound $L_{e,e'}$ and an upper bound $U_{e,e'}$ for the corresponding process time are known. For example, the running time of a train between two stations can be determined based on the details of the infrastructure between the stations (including the safety system) and the running time characteristics of the involved rolling stock. The parameters $L_{e,e'}$ and $U_{e,e'}$ preferably satisfy $0 \leq L_{e,e'} \leq U_{e,e'} \leq T - 1$. Note that a process with a process time exceeding the cycle time in principle does not fit within a cyclic timetable. Now the event times v_e and $v_{e'}$ related to the process (e, e') are connected by the following constraint.

$$(v_{e'} - v_e) \bmod T \in [L_{e,e'}, U_{e,e'}] \quad \text{for all } (e, e') \in P \tag{1}$$

Here the modulo operator mod T indicates that the timetable is cyclic with cycle time T. For example, time instant 55 is identical to time instant 175 in a cyclic timetable with cycle time $T = 60$. Moreover, if the departure and the arrival time of a train on a certain trip are 55 and 12, respectively, then the running time of the train on this trip equals $(12-55) \bmod 60 = 17$.

Since this modulo operator is difficult to handle in optimization models due to its strongly non-linear character, the cyclic character of the timetable is usually modeled by introducing binary decision variables $Q_{e,e'}$ that indicate whether the time interval between the events e and e' crosses the end of the cycle. That is, Constraint (1) is replaced by the following constraint.

$$L_{e,e'} \leq v_{e'} - v_e + T \times Q_{e,e'} \leq U_{e,e'} \quad \text{for all } (e, e') \in P \tag{2}$$

If the time interval between the events e and e' crosses the end of the cycle, then $Q_{e,e'} = 1$. If not, then $Q_{e,e'} = 0$. Loosely speaking, the decision variables $Q_{e,e'}$ determine the orders of certain pairs of events within each cycle. In the example above, we get that the running time of the trip that starts at 55 and ends at 12 equals $12 - 55 + 60 = 17$. A number of examples of constraints of type (2) is given in Section 2.2.

2.2 Example

In this section, we illustrate the different aspects of timetabling with a simple example. Moreover, we explain how these aspects can be modeled in the PESP formulation. At the end of this section, we give some results.

The example considers the triangle Amersfoort (Amf) - Deventer (Dv) - Zwolle (Zl) in the Netherlands, as depicted in Figure 2. The other relevant stations are Apeldoorn (Apd), Harderwijk (Hd) and Olst (Ost).

Consider a line plan with a minimum frequency of one train per half an hour. In Table 2.2, the details of the line plan and the minimum running times are reported. Note that in practice the minimum running times may depend on the applied rolling stock type.

Here "IC" stands for Intercity and "stop" stands for stoptrain. The columns in the table should be read as follows: the IC Amf - Dv dwells in

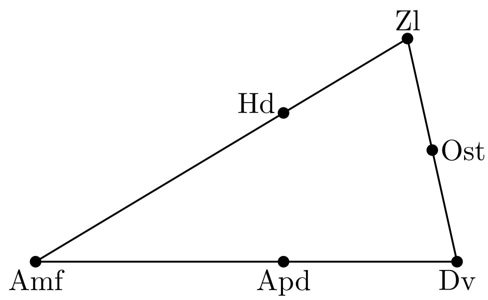

Figure 2: The triangle Amf - Dv - Zl

type	route	relevant stops	min. running time	freq.	train
IC	Amf - Dv	Apd	24 + 10	1	1a,1b
IC	Amf - Zl	-	35	2	2a,2b,3a,3b
IC	Zl - Dv	Ost	14 + 6	1	4a,4b
stop	Amf - Zl	Hd	28 + 26	1	5a,5b

Table 1: Line plan

Apd with minimum running times of 24 minutes (Amf - Apd) and 10 minutes (Apd - Dv), respectively. It runs once every half an hour. In the remainder of this section, we denote this train with number 1a in the direction of Dv and 1b in the opposite direction.

Stations that are not mentioned in Table 1 are not relevant. That is why for the stoptrain Amf - Zl only the stop in Hd is listed in the table. The dwell times for the other stops are included in the minimum running times. The IC Amf - Zl does not stop in between Amf and Zl. However, to construct the timetable one should know the time it passes Hd. The minimum running times from Amf and Zl to Hd are 15 and 20 minutes, respectively.

In the following, we give a number of examples of Constraints (2) for this instance of PESP. Because the timetable is repeated every 30 minutes, we take the cycle time T equal to 30 minutes in the model. Furthermore, we define arr_s^t and dep_s^t as the arrival and departure event of train t at station s. For the sake of simplicity, we remove the v from the notation and we denote the arrival and departure time of train t at station s as A_s^t and D_s^t.

The first type of constraints, which need to be satisfied, are the so-called running time constraints. Analogous to the general definition, dep_{Amf}^{1a} is the departure event related to the departure of train 1 (IC Amf-Dv) in Amf, and arr_{Apd}^{1a} is the arrival event of the same train at the subsequent station Apd. To ensure the required running time of at least 24 minutes on this trip, we get the following constraint.

$$24 \leq A_{\text{Apd}}^{1a} - D_{\text{Amf}}^{1a} + 30 \times Q_{dep_{\text{Amf}}^{1a}, arr_{\text{Apd}}^{1a}} \leq 29 \tag{3}$$

The upper bound of 29 has been included in this constraint since otherwise the constraint would be void. The next type of constraints that has to be taken into account, are the so-called headway constraints. This means that between two consecutive departures and arrivals on the same route, there should be a minimum headway. For instance, consider the route Amf - Zl, where the minimum time between the departures of an IC and a stoptrain in Amf is at least 3 minutes. This can be formulated as follows (Note that this constraint holds for both ICs, but we describe it only for train 2a).

$$3 \leq D^{5a}_{\text{Amf}} - D^{2a}_{\text{Amf}} + 30 \times Q_{dep^{2a}_{\text{Amf}}, dep^{5a}_{\text{Amf}}} \leq 14 \tag{4}$$

The difference in the departure times should be less than 14 minutes, because the stoptrain cannot be overtaken by the IC between Amf and Hd. In other words, to guarantee a time difference of at least 3 minutes in Hd, the difference in running time should be added to obtain the right difference in departure time in Amf (30 - (3 + 28 - 15) = 30 - 16 = 14). Similarly, other headway constraints can be formulated as well. This also holds for other types of headway constraints, like single track and crossing constraints at stations. An example of the first type is the single track between Dv and Ost. The difference in the departure times of the IC Dv-Zl and the opposite IC Zl-Dv should be at least 2 minutes on both sides of the single track section. The constraint reads as follows.

$$2 \leq D^{4b}_{\text{Dv}} - A^{4a}_{\text{Dv}} + 30 \times Q_{dep^{4b}_{\text{Dv}}, arr^{4a}_{\text{Dv}}} \leq 16 \tag{5}$$

The upper bound follows from the fact that the departure of train 4b from Dv cannot be later than the cycle time minus two times the running time on the single track line (6 minutes) plus the minimum headway time in Ost after the arrival of train 4a in Dv.

The next type of constraints that we consider involves the dwell times. Suppose that for every stop the minimum dwell time is 1 minute. For the stop in Apd, this can be modeled as follows.

$$1 \leq D^{1a}_{\text{Apd}} - A^{1a}_{\text{Apd}} + 30 \times Q_{dep^{1a}_{\text{Apd}}, arr^{1a}_{\text{Apd}}} \leq 29 \tag{6}$$

Finally, if a passenger connection between two trains (e.g. on a trip from Apd to Zl, a transfer time between 2 and 5 minutes is required in Dv) has to be scheduled, then this can be guaranteed as follows.

$$2 \leq D^{4b}_{\text{Dv}} - A^{1a}_{\text{Dv}} + 30 \times Q_{dep^{4b}_{\text{Dv}}, arr^{1a}_{\text{Dv}}} \leq 5 \tag{7}$$

Note that in this section only a subset of the relevant constraints is described. A complete list of all relevant constraints is provided in the appendix of this paper.

2.3 Objectives

Any feasible solution to the PESP system of inequalities provides a timetable satisfying all constraints. This is often an achievement in itself, but one likes to get a *good* timetable. Here, *good* can have many meanings. We will give a couple of examples of appropriate objectives, which may be conflicting. For instance, passengers prefer running times that are as short as possible (both on a single train and short transfer times from one train to another). This is often conflicting with a robust timetable, which is the topic of Section 3. The operator prefers short running times as well, since these running times affect the required amounts of rolling stock and crew. Furthermore, they like a timetable where short turn-around times for rolling stock and crew are possible at the end stations of a line. This can conflict with good transfer possibilities for the passengers.

2.4 Solving PESP

The decision variables $Q_{e,e'}$ that determine the orders of certain pairs of events in each cycle make PESP quite hard to solve by standard branch-and-bound methods: due to the relatively large coefficient T in the Constraints (2), the Linear Programming relaxations of models based on this formulation are quite weak. This is a drawback in a branch-and-bound procedure.

Therefore Schrijver and Steenbeek (1994) developed a constraint propagation algorithm for solving PESP. Their algorithm, called CADANS, has been implemented in the DONS system. This system has become an indispensable tool in the Dutch long term railway timetabling process in the Netherlands, see Hooghiemstra et al. (1999). Schrijver and Steenbeek (1994) also developed local optimization techniques to improve a feasible solution for fixed values of the variables $Q_{e,e'}$ in (2). Instances with up to 250 trains (all trains running in one hour of the Dutch timetable) can be solved usually within reasonable computing times.

Nachtigall and Voget (1996) use PESP to generate cyclic timetables with minimal passenger waiting times. Odijk (1996) uses PESP at a strategic level to determine the capacity of the infrastructure around railway stations. Kroon and Peeters (2003) describe a PESP model with variable running times. This results in a higher probability of obtaining a feasible solution.

In order to cope with the weak Linear Programming relaxation of models based on Constraints (2), Nachtigall (1999), Lindner (2000), and Peeters (2003) also describe a formulation of PESP based on cycle bases. This formulation does not use the event times as the decision variables, but the process times. Moreover, this formulation uses the fact that the sum of the process times along each directed cycle of the so-called *constraint graph* is a multiple of the cycle time. This constraint graph contains a node for each event time e, and an arc from node e to node e' for each process (e, e').

The cycle base formulation is somewhat easier to solve than the standard PESP formulation based on Constraints (2), because of the lower number of integer variables and the somewhat better Linear Programming relaxation. Several classes of cutting planes are described in the mentioned papers, which intend to further tighten the Linear Programming relaxation. Liebchen (2006) gives a complete overview of these results.

2.5 Solution to the example

We conclude this section with the "optimized" timetable of the example described in Section 2.2. To obtain this timetable, the PESP model has been implemented and solved with the commercial modeling tool and solver GAMS. As objective function we have chosen to minimize the total travel time of the passengers, thereby assuming that each Origin/Destination pair has the same number of passengers. Note that in the computations the arrival and departure times in Amf were fixed a priori.

Tables 2 and 3 report the resulting timetable. In the intermediate stations Apd, Ost and Hd, we only report the departure times. Recall that all event times are in the interval [0,29] due to the cycle time of 30 minutes.

train	2a	3a	5a		2b	3b	5b	
Amf	.04	.10	.13		.20	.23	.13	
Hd	.21	.25	.12	↓	.04	.07	.15	↑
Zl	.11	.15	.08		.12	.15	.18	

Table 2: "Optimized" timetable for Amf - Zl vv.

train	1a		1b	
Amf	.11		.23	
Apd	.06	↓	.29	↑
Dv	.16		.18	
train	4a		4b	
Dv	.18		.15	
Ost	.25	↓	.09	↑
Zl	.09		.24	

Table 3: "Optimized" timetable for Amf - Dv and Dv - Zl vv.

3 Robust timetabling

As was mentioned earlier, one of the objectives when creating a timetable is to create a timetable that is as *robust* as possible. That means that the timetable can deal as well as possible with relatively small external disturbances in the real-time operations. Thus robustness of a timetable may lead to a high punctuality in the real-time operations.

Robustness of a timetable has one or more of the following effects: (*i*) external disturbances can be absorbed to some extent so that they do not lead to delays, (*ii*) there are few secondary delays from one train to another, and (*iii*) delays disappear quickly, possibly with *small* measures of traffic control only. Both (*i*) and (*iii*) are a consequence of appropriately placed time supplements in the timetable, and (*ii*) is a consequence of appropriately placed buffer times between consecutive trains at certain locations.

Note that, with small traffic control measures only, a timetable can only be robust against *small* disturbances. If only small traffic control measures are allowed, then it is impossible to create a timetable that is robust against *large* disturbances, such as a track breakdown. In such cases, *large* measures of traffic control (e.g. re-routing or canceling trains) are inevitable.

This section describes a Stochastic Optimization Model that can be used to improve the robustness of a given cyclic timetable by modifying the time supplements and the buffer times in this timetable. As a consequence, the departure and arrival times of the trains are modified slightly. The model allocates the time supplements and the buffer times to those spots where they are as effective as possible for the robustness of the timetable.

The model requires an initial cyclic timetable as input and, based on that, it constructs an improved one with a higher robustness. In this section, we assume that this initial timetable has a cycle time T of one hour. Therefore, we will talk about hours instead of cycles.

The model contains a timetabling part for *determining* the improved timetable, and a simulation part for *evaluating* the improved timetable. In order to simulate the timetable, R realizations of the trains in the timetable are operated subject to *a priori* selected primary stochastic disturbances. Here each realization covers H consecutive hours of the timetable. Each realization can be considered as a single day.

Whereas the PESP model described in Section 2 includes the problem of determining the cyclic order of the trains on each part of the infrastructure, it is assumed in this section that these orders are given in the initial timetable and cannot be modified by the model. This is not a fundamental assumption, but it helps to keep the computation times within certain limits.

Moreover, it is assumed here that also in the realizations of the timetable the order of the trains on each part of the infrastructure is identical to that in the improved timetable (and thus to that in the initial timetable). Finally, all passenger connections between trains that are defined in the timetable are maintained in the realizations. In other words, the simulation part of the model does not include traffic control measures. However, since robustness deals with small disturbances only, traffic control measures are relatively unimportant, as was explained earlier.

3.1 Notation

As was described in Section 2, a timetable consists of a number of processes, such as running from one station to another or dwelling at a station. A process that starts with event e and ends with event e' is denoted by (e, e').

In the given cyclic timetable, the planned event time of event e is described by the parameter V_e. This parameter denotes the offset of the event time in each hour of the given cyclic timetable. Thus, in the given timetable, process (e, e') starts and ends at time instants V_e and $V_{e'}$, respectively. A process (e, e') that completely falls *within the hour* has $V_e < V_{e'}$, and a process (e, e') that *crosses the end of the hour* has $V_e > V_{e'}$.

To describe such situations, we introduce for each process (e, e') a binary *parameter* $K_{e,e'}$ that records whether or not the corresponding process crosses the end of the hour in the initial timetable. In other words, $K_{e,e'} = 1$ if and only if $V_{e'} < V_e$. The model is not allowed to modify the given cyclic orders of the events. Note that the role of the parameters $K_{e,e'}$ is similar to that of the decision variables $Q_{e,e'}$ described in Section 2.

In the improved timetable, the planned event time of event e is denoted by the decision variable v_e. The decision variable $s_{e,e'}$ denotes the planned time supplement for the process time of process (e, e'). In order to make sure that the end of the hour does not lead to undesirable restrictions for the planned event times, the planned event times are not restricted to the time interval $[0, T-1]$: they may take any (integer) value.

As was mentioned before, the timetable is evaluated during its generation by operating R realizations of the timetable subject to a priori selected independent stochastic disturbances. Each realization covers H consecutive hours of the cyclic timetable. Hour $h+1$ of realization r takes place after hour h of realization r. The stochastic disturbance of process (e, e') in hour h of realization r is denoted by the parameter $\delta_{e,e',r,h}$ for all $(e, e') \in P$, $r = 1, \ldots, R$, and $h = 1, \ldots, H$. The realized event time of event e in hour h of realization r is denoted by the variable $\tilde{v}_{e,r,h}$. The realized event times of each realization are assumed to occur on a linear time axis. Thus they are not restricted to the time interval $[0, T-1]$.

Mainly the delays of the events corresponding to arrivals of trains are evaluated, but also other delays can be taken into account. *Arrival* events are, by definition, events whose delays are measured. The set of arrival events is denoted by E_a. The delay of arrival event e in hour h of realization r is denoted by the decision variable $\Delta_{e,r,h}$. The average weighted delay of all trains is denoted by Δ. Certain events, such as a departure of a train, should not start before their corresponding planned event times. Such events are called *departure* events. The set of departure events is denoted by E_d.

3.2 Timetabling part of the model

As was explained in Section 2, most of the constraints to be satisfied in a cyclic timetabling model can be expressed in terms of the planned event times and the planned process times. For each process (e, e') that may include a variable amount of time supplement $s_{e,e'}$, we get the following.

$$M_{e,e'} + s_{e,e'} = v_{e'} - v_e + T \times K_{e,e'} \text{ for all } (e, e') \in P \tag{8}$$

The left-hand side of this equation describes the planned process time of a process (e, e') as the sum of the technically minimum process time $M_{e,e'}$ and the variable time supplement $s_{e,e'}$. The right-hand side describes it as the time difference between the planned completion time and the planned begin time of process (e, e'), thereby taking into account a possible crossing of the end of the hour by the term $T \times K_{e,e'}$.

For certain processes also an upper bound $U_{e,e'}$ on the planned process time may be specified. This results in the following constraints.

$$M_{e,e'} \leq v_{e'} - v_e + T \times K_{e,e'} \leq U_{e,e'} \text{ for all } (e,e') \in P \tag{9}$$

Other relevant constraints specify that, at each part of the infrastructure, the time difference between the last and the first planned event time within each hour should be less than the cycle time T. To that end, let e be the *first* planned event in an hour on a certain part of the infrastructure, and let e' be the *last* planned event in an hour on the *same* part of the infrastructure. Then the following constraint must be satisfied.

$$0 \leq v_{e'} - v_e \leq T - 1 \tag{10}$$

Constraints (10) are important since the event times are not restricted to the time interval $[0, T-1]$, as was explained earlier. These constraints guarantee that, after the optimization, all planned event times of the obtained timetable can be transferred back into the time interval $[0, T-1]$.

Next, in order to allocate a certain total amount of time supplement to the process times, Q subsets $B_1, \ldots, B_Q$ of processes are selected. Each subset B_q of processes is connected with a certain given amount of time supplement S_q to be allocated to the processes in the set B_q. Then the following constraints are to be satisfied.

$$\sum_{(e,e') \in B_q} s_{e,e'} \leq S_q \text{ for all } q = 1, \ldots, Q \tag{11}$$

For example, such a constraint may indicate that a certain total amount of running time supplement is to be allocated to the consecutive running times along the line of a single train. However, a certain amount of time supplement may also be allocated to a *number* of lines together.

Note that for modeling the processes that involve pairs of trains, the assumption that the orders of the events should remain unchanged is essential. Indeed, if the orders of the events would not be known a priori, then additional binary variables would be required to model these. However, given the orders of the events, all constraints can be described as in (8) – (10). Finally, non-negativity constraints have to be imposed on the variables $s_{e,e'}$, and if the timetable has to be expressed in integer minutes, then integrality constraints have to be imposed on the planned event times.

3.3 Simulation part of the model

Recall that the simulation part of the model does not include traffic control decisions. The H hours of each realization are operated one after another. A process (e,e') with $K_{e,e'} = 0$ has $V_e < V_{e'}$. Thus it is planned within a single hour. Therefore, we assume that in the realizations a process (e,e')

with $K_{e,e'} = 0$ ends in the same hour as the hour it started in. However, a process (e, e') with $K_{e,e'} = 1$ has $V_{e'} < V_e$. Obviously, in the realizations it is impossible that a process ends earlier than it started. Thus in the realizations it is assumed that a process (e, e') with $K_{e,e'} = 1$ ends in a later hour than the hour it started in. Thus, a process (e, e') with $K_{e,e'} = 1$ that starts in hour h of realization r at time instant $\tilde{v}_{e,r,h}$ ends in hour $h + 1$ of realization r at time instant $\tilde{v}_{e',r,h+1}$.

This implies that the following constraints link the event times of the processes to the technically minimum process times and the disturbances.

$$m_{e,e'} + \delta_{e,e',r,h} \leq \tilde{v}_{e',r,h+K_{e,e'}} - \tilde{v}_{e,r,h} \quad \text{for all } (e, e') \in P;\ r = 1, \ldots, R; \quad h = 1, \ldots, H \tag{12}$$

As a consequence, a delayed train in hour h of realization r may influence the trains in hour $h + 1$ of realization r. Note that the *realized* process times do not have an upper bound, in contrast with the *planned* process times that may have one. Indeed, the realized process times should have the freedom to be extended basically indefinitely, depending on the sizes of the disturbances in (12). Note further that (12) is an inequality and not an equality. Indeed, (12) only deals with process times that are enlarged due to *primary* disturbances. However, trains may also pick up *secondary* delays from interactions with other trains.

Departure events should not occur too early, and a delay corresponds to a late arrival event. This results in the following constraints.

$$v_e + h \times T \leq \tilde{v}_{e,r,h} \quad \text{for all } e \in E_d; \quad r = 1, \ldots, R;\ h = 1, \ldots, H \tag{13}$$

$$\tilde{v}_{e,r,h} - (v_e + h \times T) \leq \Delta_{e,r,h} \quad \text{for all } e \in E_a; \quad r = 1, \ldots, R;\ h = 1, \ldots, H \tag{14}$$

Here we use the cyclic character of the timetable, since the planned event time of event e in hour h of realization r equals $v_e + h \times T$.

Import delays of trains that enter the studied area can be modeled by adding a disturbance term to the left-hand side of (13). Finally, all delay variables $\Delta_{e,r,h}$ are non-negative. Indeed, positive delays of trains should not be compensated by negative delays of other trains.

As was indicated earlier, the objective is to minimize the weighted total (or average) delay of the trains. Thus the objective is to

$$\text{minimize } \Delta = \sum_{e \in E_a} \sum_{r=1}^{R} \sum_{h=1}^{H} w_e \Delta_{e,r,h} \tag{15}$$

Here the weights w_e indicate the weights of the different delays. Delays are weighted, since a delay of one train at a certain location may be more harmful to the passengers than a delay of another train at another location. Note that, besides average delays of trains, also several other aspects of the timetable may be incorporated in the objective function.

3.4 Results and timetable experiment in practice

Computational results obtained with the Stochastic Optimization Model can be found in Vromans (2005), Kroon et al. (2007a,b). These computational results show that the punctuality of a railway system can often be improved by minor modifications in the timetable.

In particular, with the same total amount of time supplements, the optimized timetables showed less knock-on delays in the real-time operations, and once delays of trains did occur, these were absorbed more quickly. In these papers it is also shown that the guideline of U.I.C. (2000) to allocate an amount of running time supplement to each trip in the timetable that is a fixed percentage of the technically minimum running time does not lead to a timetable that is maximally robust against disturbances.

Moreover, during the weeks 22 to 29 of 2006 (May 28 until July 23), a timetable generated by the Stochastic Optimization Model was tested in practice on the so-called "Zaanlijn" in the Netherlands. The "Zaanlijn" is part of the "Kop van Noord-Holland": it is the north-south connection between Den Helder and Amsterdam that is operated by Netherlands Railways. Note that a modified timetable was also operated on the "Zaanlijn" during the weeks 30 and 31 of 2006, but this period was not representative due to several other changes in the timetable at the same time.

The "Zaanlijn" has been notorious for its relatively low punctuality for several years. That is, the punctuality of the "Zaanlijn" was always significantly lower than the overall punctuality over all of the Netherlands. For this reason, it was decided by the top management of Netherlands Railways to carry out a number of experiments by temporarily operating slightly different timetables. To that end, the Stochastic Optimization Model was applied to the 2006 timetable of the "Zaanlijn". In the optimization, the total amount of running time supplement in the timetable remained the same, and the dwell times were unchanged. As a consequence, the individual travel times of the passengers changed only marginally.

Table 4 shows the average punctuality of the trains on the "Zaanlijn" as well as the average overall punctuality in the Netherlands during the weeks 22 to 29 of 2006. This table shows that, during the timetable experiment, the punctuality figures of the "Zaanlijn" are quite comparable with those of the overall punctuality. Note that weeks 24, 27 and 29 had a relatively large number of major disruptions of the railway infrastructure, which had a negative effect on the punctuality. As a comparison, the average punctuality of the "Zaanlijn" over the first 13 weeks of 2006 was 79.4%, and the average overall punctuality in the Netherlands over this period was 86.5%. Thus the

Table 4: Punctuality of the "Zaanlijn" and the overall punctuality

Week	22	23	24	25	
"Zaanlijn"	89.7%	87.5%	80.5%	89.5%	
Overall	88.6%	87.9%	82.6%	88.8%	
Week	26	27	28	29	Average
"Zaanlijn"	85.3%	74.8%	90.2%	85.8%	85.4%
Overall	86.5%	75.4%	87.3%	81.2%	84.8%

figures in Table 4 indicate that the application of the Stochastic Optimization Model had a positive effect on the punctuality of the "Zaanlijn".

In fact, the effects of the improved timetable were quite similar to what was expected. That is, at moments that there are only small primary disturbances, the improved timetable is better able to deal with these small disturbances. At moments that there are large disturbances, the improved timetable does not give a clear advantage over the original timetable.

4 Routing Trains through Railway Stations

Routing trains through railway stations is an integral part of railway timetabling, in particular in dense railway systems. In other words, as long as a detailed routing of the trains through the railway stations has not been determined, one does not have a timetable yet.

Railway stations turn out to be the main source of delays in a dense railway system. Therefore, focusing on a robust routing of trains through railway stations is highly relevant for improving the punctuality of a railway system. Note that robustness here again means insensitivity of the railway processes to relatively small disturbances in the real-time operations.

A significant number of OR publications have been devoted to the train routing problem in the past decades. Recent developments include Zwaneveld et al. (1996, 2001) who develop a model and solution approaches for finding a feasible routing for a given timetable. Their model is based on the concept of conflict graphs. This model, called STATIONS, was implemented in the timetabling tool DONS, see Hooghiemstra et al. (1999). A similar approach is used by Billionet (2003). In Section 4.4 of this paper this model is extended to take into account robustness issues as well. Fuchsberger (2007) describes an alternative multicommodity-flow type of model for solving the same problem. Caimi et al. (2005) present a heuristic algorithm for finding delay tolerant routings based on an iterative fixed point method.

4.1 The routing problem

The route of a train through a station is represented by the list of all infrastructure elements passed by the train. In particular, it includes the platform tracks assigned to the train. Once one knows the acceleration and deceleration capabilities as well as the planned arrival and departure times of a train, one can determine the exact time intervals when the infrastructure elements are occupied by the train according to the plan.

One major goal of routing trains through stations is to ensure that no infrastructure element is occupied by two different trains within a certain headway time. Robustness of the routing can be captured by reducing the number of occasions when the time difference of two trains on an infrastructure element is just higher than the minimum headway time.

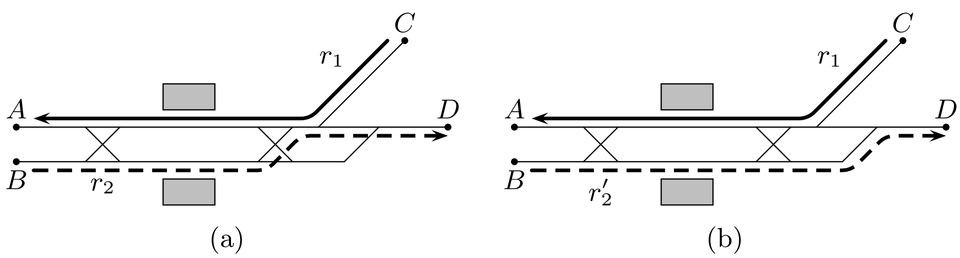

Figure 3: Crossing and non-crossing routes

Figure 3a indicates the lay-out of a station and shows the routes of two trains: train t from C to A and train t' from B to D. Note that routes r and r' share a piece of the infrastructure. Therefore, they cannot be used both if trains t and t' pass through the station at the same time. If, however, train t passes, say, more than 3 minutes earlier than train t' does, then the routing is feasible. However, in that case a small delay of train t will hinder train t'. Figure 3b indicates an alternative routing for trains t and t'. This routing is considered to be more robust than the previous one because routes r and r' are disjoint.

4.2 Platform assignment issues

The goal of routing trains through stations is, besides specifying the approach routes from and to the platforms, to assign the trains to the platforms themselves. The platform assignment has to satisfy a number of market requirements. For example, trains of the same line are to be assigned to the same platform, while trains departing towards the same destination are preferred to leave from the same platform. Then passengers can easily remember the departure platforms of their trains.

Certain pairs of trains have so-called *cross-platform connections*. These trains have short stops with overlapping time intervals, and are to be assigned to neighboring platforms. Then passengers can easily change from one of these trains to the other in that case.

The platforms of large stations may have different preference values when being assigned to trains. For example, trains with many expected passengers are preferred to arrive at platforms that can accommodate long enough trains and that are located closer to the main station facilities.

4.3 Finding a feasible routing

Zwaneveld et al. (1996, 2001) describe a model for routing trains through a railway station, given the arrival and departure times of the trains that were determined in an earlier stage. The basic idea is as follows. First the set of all possible routes is identified for each train. Next, one pre-computes which train-route pairs are conflicting, since they lead to a violation of the minimum headway time at one of the infrastructure elements. Then the model makes sure that none of these conflicting train-route pairs appear together in a solution. Note that Zwaneveld et al. (1996, 2001) distinguish inbound routes from outbound routes, but this distinction is neglected here.

Let T be the set of trains. For each train t, let R_t denote the set of all routes that can be assigned to train t. For train t and route $r \in R_t$, the parameter $p_{t,r}$ is a penalty for this combination. These penalties may indicate e.g. the train's preference for the platforms. It may also reflect how much of the station's capacity is blocked by route r: real-time operations can be easily disrupted if r has a conflict with many other possible routes.

The set $\mathcal{C}$ describes the possible conflicts between trains. The elements of $\mathcal{C}$ are 4-tuples (t, r, t', r') with $t, t' \in T$, $r \in R_t$, $r' \in R_{t'}$ such that choosing routes r and r' for trains t and t' violates the minimum headway time of one of the infrastructure elements. Also, the market requirements discussed in Section 4.2 are easily expressed by additional elements of $\mathcal{C}$. For example, the 4-tuple (t, r, t', r') is an element of $\mathcal{C}$ if trains t and t' should have a cross-platform connection, but routes r and r' lead to non-neighboring platforms.

Binary decision variables $X_{t,r}$ then describe whether route r is selected for train r. With these notations, the model reads as follows.

$$\begin{aligned}
\text{minimize} \quad & \sum_{t \in T} \sum_{r \in R_t} p_{t,r} X_{t,r} && && (16)\\
\text{subject to} \quad & \sum_{r \in R_t} X_{t,r} = 1 && \text{for all } t \in T && (17)\\
& X_{t,r} + X_{t',r'} \leq 1 && \text{for all } (t, r, t', r') \in \mathcal{C} && (18)\\
& X_{t,r} \in \{0, 1\} && \text{for all } t \in T, r \in R_t && (19)
\end{aligned}$$

Note that this model is similar to a Node Packing Model. Zwaneveld et al. (1996, 2001) solve the model by first looking for dominated routing possibilities. For example, if routes r and r' have the same start and end point, but route r' is a detour in comparison with route r, then all routing possibilities involving route r' are dominated by the routing possibilities involving route r. Moreover, the authors aggregate constraints (18) into genuine clique-constraints, which usually provide a stronger Linear Programming relaxation. The aggregated model is solved then by CPLEX, usually within a couple of minutes, even for larger sized stations.

4.4 Robust routing

The model described in Section 4.3 does not deal with any robustness aspects. Therefore, the model is extended here by maximizing the time between crossing movements of trains. In this way, the probability of knock-on delays is minimized. This fits with the robustness idea that consecutive utilizations of a single resource by different users should be spread in time.

Let $q_{t,r,t',r'}$ be a non-negative penalty value for each pair of trains t, t' and for all routes $r \in R_t$ and $r' \in R_{t'}$. These penalty values express the impact of the assignment of routes r and r' to trains t and t', respectively, on the robustness of the timetable. The penalty is zero if the simultaneous choice of routes r and r' for trains t and t', respectively, does not jeopardize the robustness. This happens if the routes are disjoint or if the time difference between trains t and t' is sufficiently large.

Then the robustness aspects are incorporated in the model (16) – (19) by replacing (16) by the following quadratic objective function.

$$\text{minimize} \quad \sum_{t\in T}\sum_{r\in R_t} p_{t,r}X_{t,r} + \sum_{t\in T}\sum_{r\in R_t}\sum_{t'\in T}\sum_{r'\in R_{t'}} q_{t,r,t',r'}X_{t,r}X_{t',r'} \tag{20}$$

In this extended model, all penalties $q_{t,r,t',r'}$ for selected pairs (t,r) and (t',r') are accumulated in the objective function. Indeed, $X_{t,r}X_{t',r'} = 1$ if and only if $X_{t,r} = 1$ and $X_{t',r'} = 1$. The optimal solution is a feasible routing which minimizes the sum of the involved penalties, thereby also taking into account the robustness aspect.

4.4.1 Linearization of the robust routing model

The quadratic integer programming model described in the previous section is too difficult to treat computationally. For example, the non-convex objective function rules out efficient convex optimization methods. However, with an appropriate linearization of the model, one may use sophisticated mixed integer linear programming algorithms.

An easy linearization method is to introduce non-negative continuous decision variables $Y_{t,r,t',r'}$, to replace the product $X_{t,r}X_{t',r'}$ by $Y_{t,r,t',r'}$ in the objective function (20), and to impose the following additional constraints.

$$X_{t,r} + X_{t',r'} \leq 1 + Y_{t,r,t',r'} \tag{21}$$

This is indeed a sound method, since in any optimal solution the value $Y_{t,r,t',r'}$ equals the product of $X_{t,r}$ and $X_{t',r'}$, as one easily checks. A drawback of this method is, however, that it leads to a weak Linear Programming relaxation. Therefore, other linearization methods will have to be used for solving practical instances. However, a discussion of other linearization methods falls out of the scope of this paper.

4.4.2 Aggregated routes

It turns out that the number of possible routes through a reasonably big station is quite large, limiting the possibility of applying the routing model for real-life problems. One way to deal with this is to work with a restricted set of allowed routes. This way is followed by Zwaneveld et al. (1996, 2001).

However, here we suggest another possibility. Instead of using the fully detailed routes through the station, we use the notion of aggregated routes, as indicated in Figure 4. The bold lines in Figure 4a indicate the detailed routes of two trains. An aggregated route arises by keeping the platform and the destination (A, B or C in the figure) of a route fixed, and by neglecting all intermediate infrastructure elements. The aggregated routes are represented by the bold arrows in Figure 4b.

Conflicts between aggregated routes are defined as follows. Let the platforms and the destinations be indexed from top to bottom. Then two aggregated routes are conflicting if the route from a lower indexed platform departs to a higher indexed destination, i.e. if the arrows in Figure 4b cross each other.

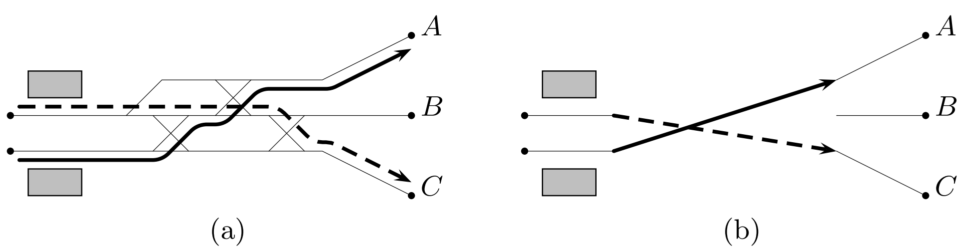

Figure 4: Detailed and aggregated routes

Subsequently, we solve the robust routing problem with these aggregated routes. The solution gives an indication of the detailed routing, which can be found afterwards by existing "non-robust" routing algorithms by fixing the platform assignment and looking for just a feasible routing.

Whenever the aggregated routes cross one another, the corresponding routes in any feasible detailed routing cross one another, too. However, it is not difficult to find examples where the opposite implication does not hold. Therefore, the robust routing model with the aggregated routes yields a lower bound on the penalty of any feasible detailed routing.

Having split the robust detailed routing problem into two parts (namely into robust aggregated routing and feasible detailed routing), there is a risk that a robust aggregated routing does not give rise to any feasible detailed routing. In such cases, it is vital to define more complex aggregated routes by taking key infrastructure elements (such as fly-overs, long parallel tracks without switches, etc.) into account. This enhances the probability of indeed finding a feasible detailed routing.

4.4.3 Computational experience and conclusions

A linearized formulation of the robust routing model (17) – (20) was implemented for the major Dutch railway node Utrecht Central Station in the Netherlands. Since the cycle time of the periodic timetable of Netherlands Railways is one hour, it suffices to compute the routes for a single hour only. In order to reduce the computation times, the aggregated routes that were described in Section 4.4.2 were used.

Utrecht Central Station is a busy station with about 55 trains passing every hour, bound to 5 destinations. The passenger trains can arrive at 14 platforms; freight trains may use three additional dedicated tracks to pass through the station. The aggregated routes regard the complex system of fly-overs at the northern side of the station.

The mixed integer program was solved by CPLEX 9.0 to optimality within an hour on a standard PC. The Linear Programming bounds of the first few nodes in the Branch-and-Bound tree are very weak. Therefore it is advantageous to instruct CPLEX to start the solution process with intensive branching, and to look for feasible solutions later.

The test implementation computed the aggregated routes only. A further manual step indicated that the model captured the key elements of the infrastructure conflicts in an adequate way, since the aggregated routes could be extended to a feasible detailed routing through the station.

The results showed that a manually planned routing usually can be improved slightly. But more importantly, the model allows a direct and quick evaluation of the effect of certain market requirements, such as cross-platform connections, on the robustness of the routing. Cross-platform connections have a negative effect on the robustness of the routing. Indeed, whereas the robustness objective tries to separate trains from each other in space and time, cross-platform connections do the opposite.

The model can also show the usefulness of minor changes of the arrival and departure times. In fact, shifting the arrival time of a single cargo train at Utrecht Central Station by one minute allows to decrease the optimal objective value by about 50%. In further research, detailed simulations of the processes inside the stations are to be carried out to investigate how well the objective function (20) reflects the robustness of the obtained routings.

5 Final remarks

An advantage of the application of optimization models is that it may lead to better solutions in a shorter throughput time. Better solutions may be the result of the fact that the application of optimization models enables one to study several solutions instead of just a single one. For example, one may compare the most cost-efficient solution with the solution that provides a maximum service to the passengers. In this way, one may explicitly study the trade-off between costs and service. Moreover, in several practical cases, the application of optimization models lead to improvements in several criteria at the same time. The latter results from the fact that manual planning methods are usually heuristic approaches.

Optimization models play an increasing role in the planning processes of railway systems. For example, the timetable of the Berlin metro that has been in operation since December 2005 was created completely with the help of optimization techniques based on the PESP model that was described in Section 2. Further details on this case can be found in Liebchen (2006).

Similarly, within Netherlands Railways the models described in this paper have developed into indispensable tools in the railway planning process. As was mentioned earlier, these models have been implemented in the automatic timetabling system DONS (=Designer Of Network Schedules). This system has been used extensively in the development of the Dutch railway timetable that has been in operation since December 2006. Also the simulation model SIMONE (SImulation MOdel for NEtworks) was used in this development process, see Middelkoop and Bouwman (2000). Whereas in earlier years new timetables were usually created in an incremental way by modifying an existing timetable, the current Dutch timetable was generated completely from scratch with the help of the described models. Also the available models for rolling stock circulation and crew scheduling played an indispensable role in this planning process. More details on these models can be found in Fioole et al. (2006) and Abbink et al. (2005).

One of the main objectives in the development of the current Dutch timetable was to improve its robustness, which should translate into increased punctuality figures in the operations. This was aimed at by allocating time supplements and buffer times in a different way, and by reducing the number of crossing train movements in station areas as much as possible. Although at the moment of writing (May 2007) the time that the current timetable has been in operation is still too short for definitive conclusions, a certain upward trend in the punctuality figures can be discovered.

A major challenge for further research is to find a better integration between the models developed for solving the timetabling step described in Sections 2 and 3 and the models developed for routing trains through railway stations described in Section 4. A better integration between these models will allow one to take into account more details of the routes through the stations already in the timetabling step. This may be beneficial for the quality of the finally obtained timetable, in particular for its robustness.

Moreover, until now, the *planning* process of a railway system has been the traditional application area for optimization models. Here one may apply large optimization models that can be solved to near optimality without caring too much about the computation time. However, decision support for *re-scheduling* in case of a disruption in the real-time operations may be even more relevant for increasing the quality of a railway system.

In particular, after a disruption of the railway system has occurred, decision support is needed on which trains to cancel or to re-route in order to uphold as much as possible of the service for the passengers, or on how to adjust the rolling stock circulations and the crew schedules. Disruption management is a re-active way of dealing with disruptions. It requires a fundamentally different focus of the involved optimization models: now a short computation time is essential, and near optimality of the obtained solutions is less relevant, but not at any price. This different focus requires still a lot of research, which poses an enormous challenge for the research community.

Acknowledgements The authors like to thank Twan Dollevoet, Willem van Jaarsveld and Evelien van der Hurk for calculating the timetable as presented in the example of Section 2.

References

E. Abbink, M. Fischetti, L. Kroon, G. Timmer, and M. Vromans. Reinventing crew scheduling at Netherlands Railways. *Interfaces*, 35(5):393–401, 2005.

A. Billionet. Using integer programming to solve the train-platforming problem. *Transportation Science*, 37:213–222, 2003.

G. Caimi, D. Burkolter, and T. Herrmann. Finding delay-tolerant train routings through stations. In H. Fleuren, editor, *Operations Research Proceedings 2004*, pages 136–143. Springer, 2005.

A. Caprara, M. Fischetti, and P. Toth. Modeling and solving the train timetabling problem. *Operations Research*, 50:851–861, 2002.

P. Fioole, L. Kroon, G. Maróti, and A. Schrijver. A rolling stock circulation model for combining and splitting of passenger trains. *European Journal of Operational Research*, 174(2):1281–1297, 2006.

M. Fuchsberger. Solving the train scheduling problem in a main station area via a resource constrained space-time integer multicommodity flow. Master's thesis, Institute for Operations Research, ETH Zürich, Switzerland, 2007.

J. Hooghiemstra, L. Kroon, M. Odijk, M. Salomon, and P. Zwaneveld. Decision Support Systems support the search for win-win solutions in railway network design. *Interfaces*, 29(2):15–32, 1999.

D. Huisman, L. Kroon, R. Lentink, and M. Vromans. Operations Research in passenger railway transportation. *Statistica Neerlandica*, 59:467–497, 2005.

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

L. Kroon, R. Dekker, G. Maróti, M. Retel Helmrich, and M. Vromans. Stochastic improvement of cyclic railway timetables. ERIM Research Report ERS-2006-067-LIS, Erasmus University Rotterdam, The Netherlands, 2007a.

L. Kroon, R. Dekker, and M. Vromans. Cyclic railway timetabling: a Stochastic Optimization approach. In F. Geraets, L. Kroon, A. Schoebel, D. Wagner, and C. Zaroliagis, editors, *Algorithmic Methods for Railway Optimization, Lecture Notes in Computer Science*, Berlin, 2007b. Springer.

C. Liebchen. *Periodic Timetable Optimization in Public Transport*. PhD thesis, TU Berlin, Germany, 2006.

T. Lindner. *Train schedule optimization in public rail transport*. PhD thesis, Technical University Braunschweig, Braunschweig, 2000.

D. Middelkoop and M. Bouwman. Train network simulator for support of network wide planning of infrastructure and timetables. In J. Allan, C. Brebbia, R. Hill, and G. Sciutto, editors, *Computers in Railways VII*, pages 267–276, Ashurst, UK, 2000. WIT Press.

K. Nachtigall. Periodic network optimization and fixed interval timetables. Habilitation thesis, Deutsches Zentrum für Luft- und Raumfahrt, Braunschweig, 1999.

K. Nachtigall and S. Voget. A genetic algorithm approach to periodic railway synchronization. *Computers and Operations Research*, 23:453–463, 1996.

M. Odijk. A constraint generation algorithm for the construction of periodic railway timetables. *Transportation Research*, 30:455–464, 1996.

L. Peeters. *Cyclic Railway Timetable Optimization*. PhD thesis, Erasmus University Rotterdam, Rotterdam School of Management, The Netherlands, 2003.

A. Schrijver and A. Steenbeek. Dienstregelingontwikkeling voor Railned (Timetable construction for Railned). Technical report, Center for Mathematics and Computer Science, Amsterdam, 1994. In Dutch.

P. Serafini and W. Ukovich. A mathematical model for Periodic Event Scheduling Problems. *SIAM Journal of Discrete Mathematics*, 2:550–581, 1989.

U.I.C. Timetable recovery margins to guarantee timekeeping - recovery margins. Leaflet 451-1 of U.I.C., Paris, France, 2000.

M. Vromans. *Reliability of Railway Systems*. PhD thesis, Erasmus University Rotterdam, Rotterdam School of Management, The Netherlands, 2005.

P. Zwaneveld, L. Kroon, H. Romeijn, M. Salomon, S. Dauzère-Pérès, C. Van Hoesel, and H. Ambergen. Routing trains through railway stations: Model formulation and algorithms. *Transportation Science*, 30(3):181–194, 1996.

P. Zwaneveld, L. Kroon, and S. van Hoesel. Routing trains through a railway staion based on a node packing model. *European Journal of Operational Research*, 128: 14–33, 2001.