Public Transp (2009) 1: 135–154
DOI 10.1007/s12469-009-0010-0

ORIGINAL PAPER

Capacity constraints in delay management

Anita Schöbel

Published online: 19 June 2009

Abstract We consider (small) disturbances of a railway system. In case of such delays, one has to decide if connecting trains should wait for delayed feeder trains or if they should depart on time, i.e. which connections should be maintained and which can be dropped. Finding such wait-depart decisions (minimizing e.g. the average delay of the passengers) is called the *delay management problem*. In the literature, the limited capacity of the tracks (meaning that no two trains can use the same piece of track at the same time) has so far been neglected in the delay management problem. In this paper we present models and first results integrating these important constraints. We develop algorithmic approaches that have been tested at a real-world example provided by *Deutsche Bahn AG*.

1 Introduction

Dealing with delayed vehicles is an important issue in the daily operational business of any public transportation company. If delays occur, the timetable has to be updated to a so-called *disposition timetable*. The goal is to find a disposition timetable which is convenient for the passengers, but on the other hand respects all operational constraints. Operational constraints in rail transportation mainly deal with the limited capacity of the track system. These capacity constraints basically ensure that there is no conflict between two trains, which are about to use the same piece of infrastructure. To this end, a fixed block system is used in many European countries: The track

This work was partially supported by the Klaproth-Stiftung and by the Future and Emerging Technologies Unit of EC (IST priority—6th FP), under contract no. FP6-021235-2 (project ARRIVAL).

A. Schöbel (✉)
Institute for Numerical and Applied Mathematics, Georg-August University Göttingen, Lotzestrasse 16-18, 37083 Göttingen, Germany
e-mail: schoebel@math.uni-goettingen.de

system is divided into blocks, and it is never allowed that two trains use the same block at the same time. Sometimes, also a rolling block system is used.

If the capacity of the track system is neglected, the problem from the passengers' point of view reduces to the following *(pure) delay management problem*: For each possible connection one has to decide if a connecting vehicle should wait for a delayed feeder train or if it is better to depart on time. The problem has first been introduced in Schöbel (2001) and further dealt with in Schöbel (2006, 2007), De Giovanni et al. (2008). As shown in Gatto et al. (2005) it is NP-hard, complexity issues are also treated in Gatto et al. (2004).

On the other hand, if the wait-depart decisions are neglected, the problem of finding a disposition timetable is called *railway re-scheduling problem*. There is an extensive amount of publications about re-scheduling in railway systems. For an older survey, see Cordeau (1998), a recent state-of-the art paper about routing and re-routing is Törnquist (2005). The special case of routing trains through railway stations has been considered e.g. in Zwaneveld (1996), Zwaneveld et al. (1996), Billionnet (2003). Re-Scheduling between stations has e.g. been considered in Brucker et al. (2002), and the complete railway system is treated e.g. in Adenso-Díaz et al. (1999), Wegele and Schnieder (2005). Recently a new approach has been presented in Velasquez et al. (2005), where constraint branching is used to find new routes through a railway station in case of delays. In the practice of many railway companies, priority rules are used for re-scheduling, see, e.g. Pachl (2000), Jacobs (2004).

In the current paper our goal is to provide a new model, which allows to include the capacity constraints in the formulation of the delay management problem. Some rough ideas and a simulated annealing approach in this field are handled in Norio et al. (2005); to the best of our knowledge no other integrated approaches are known so far.

When respecting capacity constraints in delay management, the first difficulty arising are the different levels of detail in delay management and re-scheduling. While a macroscopic scale is sufficient in delay management (nodes represent stations) a microscopic level is necessary to model the capacity constraints (blocks, platforms, and in the worst case all signaling points have to be considered as nodes). The application of the problem is obvious: Our research has been stimulated by a real-world application within the project *DisKon* of Deutsche Bahn (see Bissantz et al. 2005).

We start by presenting a formulation for the delay management problem (without capacity constraints) in Sect. 2, before we briefly present different possibilities to take capacity constraints into account in Sect. 3. In Sect. 4 we show how the microscopic capacity constraints can be lifted into the macroscopic model and develop properties of the resulting integrated model. Solution approaches and first numerical results are shown in Sect. 5.

2 Delay management without capacity constraints

Let a public transportation network $PTN = (V, E)$ with a set of stations V and a set of direct links between stations in E be given together with a set of trains $\mathcal{F}$. A *connection* in such a network is a triple (i, j, v) with $i, j \in \mathcal{F}$, $v \in V$, and such that passengers can transfer from vehicle i to vehicle j at station v.

The (pure) delay management problem, first introduced in Schöbel (2001), decides which connections should be maintained in case of delays and which other connections can be dropped. The goal is to minimize the inconvenience for the passengers, given by the delay they have when reaching their final destinations. To calculate such delays, it is not only necessary to know the wait-depart decisions, but it is also important to keep track of the new disposition timetable.

For an elegant formulation, we use the concept of event-activity networks (see e.g. Nachtigall 1998 for its usage in timetabling). Let us call an arrival of a vehicle g at a station v an *arrival event* (g, v, arr), and a departure of some vehicle g at some station v a *departure event* (g, v, dep). The event activity network is a graph $\mathcal{N} = (\mathcal{E}, \mathcal{A})$ with

- node set $\mathcal{E} = \mathcal{E}_{arr} \cup \mathcal{E}_{dep}$ with
  - $\mathcal{E}_{arr} = \{(g, v, arr) :$ train g arrives at station $v \in V\}$ as the set of *arrival events* and
  - $\mathcal{E}_{dep} = \{(g, v, dep) :$ train g departs from station $v \in V\}$ as the set of *departure events*.
- and directed edges $\mathcal{A} \subseteq \mathcal{E} \times \mathcal{E}$ consisting of *waiting, driving and changing activities* further specified below:

$$\begin{aligned}
\mathcal{A}_{wait} &= \{((g, v, arr), (g, v, dep)) \in \mathcal{E}_{arr} \times \mathcal{E}_{dep}\}\\
\mathcal{A}_{drive} &= \{((g, v, dep), (g, u, arr)) \in \mathcal{E}_{dep} \times \mathcal{E}_{arr} : e = (v, u) \in E\},\\
\mathcal{A}_{change} &= \{((g, v, arr), (h, v, dep)) \in \mathcal{E}_{arr} \times \mathcal{E}_{dep} : \text{ a connection}\\
&\qquad \text{from vehicle } g \text{ into } h \text{ at station } v \text{ is required}\}.
\end{aligned}$$

To simplify notation, let $\bar{i} \in \mathcal{F}$ denote the train corresponding to event $i \in \mathcal{E}$.

We remark that $\mathcal{N}$ is a special case of a time-expanded network and hence acyclic. Further note that $(i, j) \in \mathcal{A}$ means that event i has to be performed before event j can take place. A small event-activity network is depicted in Fig. 1.

Using the notation of event-activity networks, a *timetable* Π is given by assigning a time Π_i to each event $i \in \mathcal{E}$. In the context of delay management, we are, however, interested in the disposition timetable, which will be called x_i, $i \in \mathcal{E}$. We further define L_a as the technical minimal necessary time for performing activity a, w_i as the number of passengers getting off at event $i \in \mathcal{E}_{arr}$ ($w_i = 0$ for $i \in E_{dep}$) and w_a as the number of passengers planning to use the connection $a \in \mathcal{A}_{change}$. We also assume that we have a fixed time period T in which the timetable is repeated, and that all vehicles are on time in the next period.

Let us finally assume that all *source delays* are known, i.e., we have a set of events $\mathcal{E}_{del} \subseteq \mathcal{E}_{arr}$ such that $d_i > 0$ for all $i \in \mathcal{E}_{del}$. For non-delayed events we set $d_i = 0$. We need the following two types of variables:

For all changing activities $a \in \mathcal{A}_{change}$ we introduce

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

and for all events $i \in \mathcal{E}$ we need

$$x_i = \text{actual time of event } i.$$

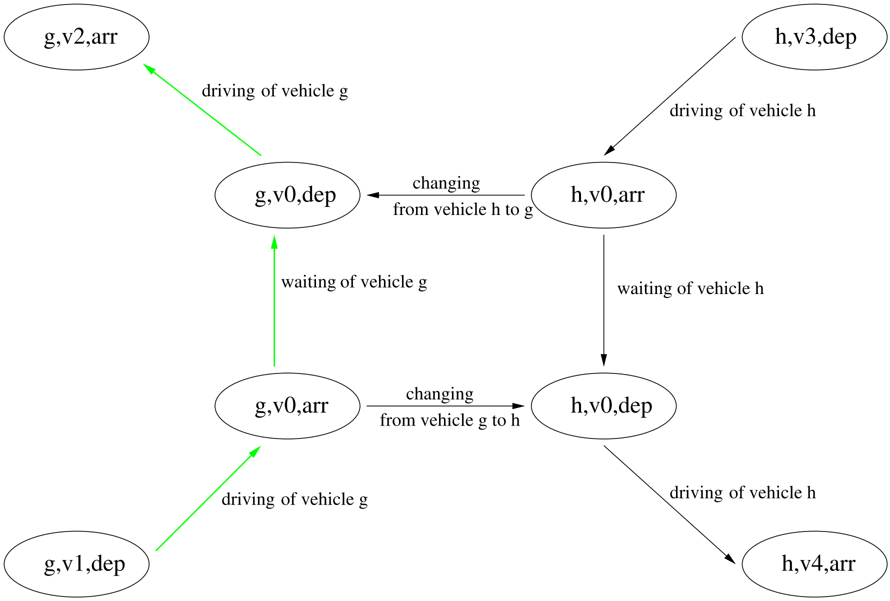

Fig. 1 An event-activity network

Note that the delay of event i is hence given by $x_i - \Pi_i$ and that we have to require that $x_i \geq \Pi_i$ holds, since no train is allowed to start earlier as planned. The following is an integer programming formulation of the (pure) delay management problem.

$$\text{(DM)} \quad \min f(x,z) = \sum_{i \in \mathcal{E}} w_i (x_i - \Pi_i) + \sum_{a \in \mathcal{A}_{change}} w_a T z_a$$

such that

$$x_i \geq \Pi_i + d_i \quad \text{for all } i \in \mathcal{E} \tag{1}$$

$$x_j - x_i \geq L_a \quad \text{for all } a = (i,j) \in \mathcal{A}_{wait} \cup \mathcal{A}_{drive} \tag{2}$$

$$M z_a + x_j - x_i \geq L_a \quad \text{for all } a = (i,j) \in \mathcal{A}_{change} \tag{3}$$

$$x_i \in \mathbb{N} \quad \text{for all } i \in \mathcal{E}$$

$$z_a \in \{0,1\} \quad \text{for all } a \in \mathcal{A}_{change}$$

The first constraint (1) makes sure that no train departs earlier than scheduled, while (2) ensures that the delay is carried over correctly from one event to the next. In particular, if event i takes place at some time point x_i, event j must be later than $x_i + L_a$ if $a = (i,j)$ is the activity linking i and j. If $z_a = 0$, constraint (3) is the same as (2) and hence ensures that the delay is carried over for each maintained connection. For $z_a = 1$, however, constraint (3) becomes redundant whenever M is large enough. Note that $M \geq \max_{i \in \mathcal{E}} d_i$ suffices.

The above formulation minimizes a combination of (weighted) dropped connections and (weighted) train delays. The weight of a (dropped) connection $a \in \mathcal{A}$ 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 (see Schöbel 2007). It is remarkable that the never-meet property is *almost* correct in practice, i.e. in many practical cases it is "almost" satisfied.

Formulation (DM) 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 (Ginkel and Schöbel 2007; Heidergott and de Vries 2001).

If the z_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

$$x_1 := \Pi_1 + d_1,$$
$$x_i := \max\left\{\Pi_i + d_i, \max_{a=(j,i)\in\mathcal{A}} x_j + L_a\right\}, \quad i = 2, \dots, n. \tag{4}$$

Usually, the wait-depart decisions z_a are not known. Then (DM) can be solved by a branch & bound approach, in which we branch along the changing activities in their natural order. In each step we fix a variable z_a to *wait* or *not wait*. Changing activities with non-fixed variables are called *open*. If all z_a are fixed, one can use (4) to calculate a solution. If there are still open changing activities in the actual branch & bound node we determine an upper bound by a heuristic, and a lower bound by solving the corresponding LP-relaxation.

Branch & Bound for (DM)

Input: (DM)
Step 1: Order the changing activities $a = (i, j)$ according to Π_j.
Set LIST:= $\{(DM)\}$, $f^* := \infty$
Step 2: If LIST= $\emptyset$ **stop**. Optimal solution (x^*, z^*). Otherwise select and delete a problem $(P) \in$LIST.
Step 3: Calculate an upper bound f_P^{upper} for (P) by fixing all open changing activities $z_a = 1$ and using (4) to calculate the x-variables.
Let x_P^{upper}, z_P^{upper} be the solution.

3.1: If $f_P^{upper} < f^*$ let $f^* = f_P^{upper}$ and $x^* = x_P^{upper}$, $z^* = z_P^{upper}$
3.2: Goto Step 4.

Step 4: Solve the LP-relaxation of (P). Denote the optimal solution by
x_P^{lower}, z_P^{lower} and its objective function value by f_P^{lower}.

4.1: If $f_P^{lower} > f^*$ goto Step 2.

4.2: `If` z_P^{lower} `is boolean do`
- `If` $f_P^{lower} < f^*$ `let` $f^* = f_P^{upper}$ `and` $x^* = x_P^{lower}$, $z^* = z_P^{lower}$
- `Goto Step 2`

Step 5: `Take the first open changing activity` a `of (P). Add two new problems to LIST, namely` (P_a^{wait}) `and` (P_a^{depart})`, where in the first one` $z_a := 0$ `and in the later one` $z_a := 1$ `is fixed. Goto Step 2.`

According to the selection rule in Step 2 one can obtain depth-first or width-first branch & bound trees. Their different behaviors is currently studied within the project *DisKon*, see Jobmann and Schöbel (2007).

3 Integrated approaches

To have realizable disposition plans, it is necessary to take the limited capacity of the track system into account. The crucial constraint basically is that no two trains can use the same piece of the infrastructure at the same time. To this end, the track system is divided into *blocks* (usually between two signaling points) and we have to make sure that no two trains will occupy the same block at the same time. Note that there are significant differences in the lengths of the blocks: It may differ between some 10 meters and several kilometers.

The following three different concepts allow to take capacity constraints into account:

1. Iterative approach
2. Microscopic approach
3. Macroscopic approach

The first approach (see Fig. 2) is used in the project DisKon (Bissantz et al. 2005). In the macroscopic step, several solutions of the delay management problem are calculated which are re-scheduled in a microscopic step. The solution with best performance in both steps is taken. The process can be repeated, if there is enough time. The approach is a heuristic; it needs not find a global optimum.

In a microscopic approach the capacity constraints are modeled explicitly. This leads to huge integer programs, which include disjunctive constraints. A promising possibility is to model the problem within a set packing approach and use constraint branching (see Ryan and Foster 1981) for its solution, see Velasquez et al. (2005).

In this paper we present a macroscopic approach which allows to treat the most important capacity constraints. To make sure that the solution obtained is applicable, a microscopic step needs to be added (as in the first approach) but the changes which are necessary to obtain a feasible solution are expected to be rather small.

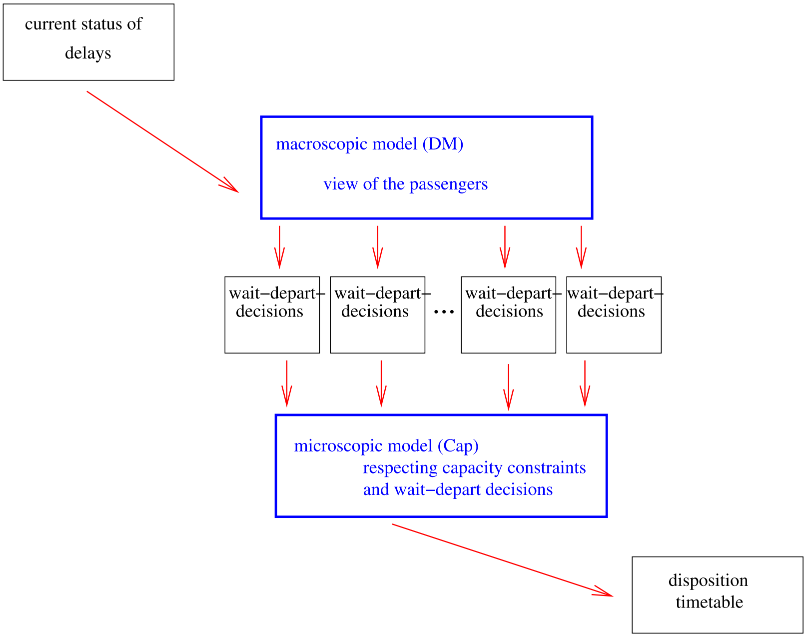

Fig. 2 The iterative approach used in the project DisKon, see Bissantz et al. (2005)

4 Capacity constraints in the macroscopic model

To formulate the capacity constraints in the macroscopic model we neglect the blocks and look at the edges between two stations. For each edge we determine the minimal time between two departures that prevents any block conflict (*headway*). If all trains have the same speed, the headway is given as

$$H_e := \max\{\text{driving time } b : b \text{ is block on edge } e\}$$

which is independent of the specific trains leaving. If trains have different speeds, the headway is not only dependent on the edge, but also on the two trains, or on the two events i and j, respectively.

Definition 1 By H_{ij} we denote the minimal time which has to be respected if event j follows event i. H_{ij} is called the *headway* between events i and j.

If $i = (\bar{i}, v, dep)$ and $j = (\bar{j}, v, dep)$ the headway is the time that train $\bar{j}$ corresponding to event j has to wait after the departure of train $\bar{i}$ corresponding to event i. The same can be done for oncoming traffic using the same single-track line. In this case the headway H_{ij} is the time, train $\bar{i}$ needs to pass the single track until the next station (or crossing point) and that has to be respected before the oncoming train $\bar{j}$ may leave from the opposite side. In both cases, we obtain disjunctive ordering constraints similar to the approach suggested by Szpigel (1973) for one long single track

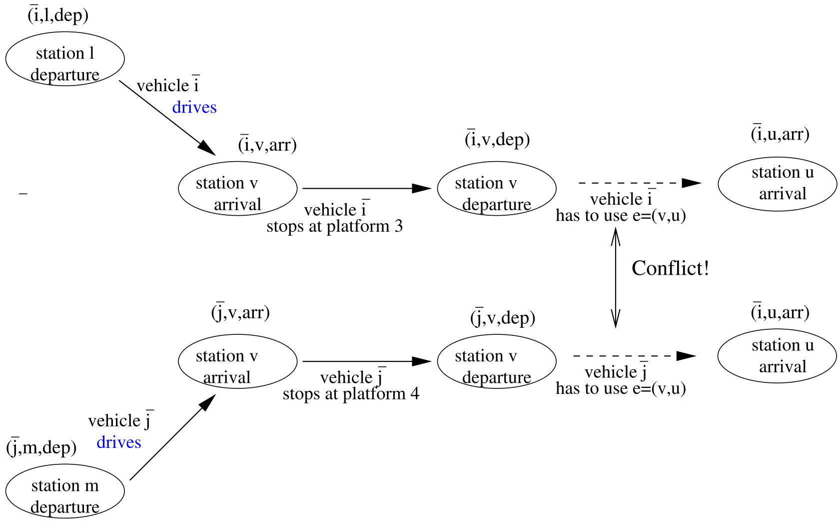

Fig. 3 The events $(\bar{i}, v, dep)$ and $(\bar{j}, v, dep)$ belong to the same set $\mathcal{E}(e)$ for the physical edge $e = (v, u)$

line. The set of feasible solutions (w.r.t. the headways between events i and j) is hence given as the set of all x_i, x_j such that either $x_i \geq H_{ji} + x_j$ or $x_j \geq H_{ij} + x_i$.

We now have to specify the set of events which compete for the same piece of track, since between these events we have to establish headway constraints. To this end, note that each departure event i corresponds to one unique driving activity $a = (i, j) \in \mathcal{A}_{drive}$. The physical edge $e \in E$ of the public transportation network PTN which is used by the driving activity a following event i is denoted by $e(i)$. I.e. $e(i)$ is the edge given by $(v, u) \in E$ if $i = (\bar{i}, v, dep)$ and $j = (\bar{i}, u, arr)$. For a departure event i, this means that $e(i)$ denotes the physical edge belonging to the unique driving activity following the departure event i. For any edge $e \in E$ we define

$$\mathcal{E}(e) = \{i \in \mathcal{E}_{dep} : e(i) = e\}$$

as the set of all events which are scheduled on the same infrastructure e in their next activity and are hence competing for it. (We can easily replace edges by blocks, if required.)

The situation is illustrated in Fig. 3, where the two trains $\bar{i}$ and $\bar{j}$ have to use the same track $e = (v, u)$ after their departures in station v. In this case we obtain

$$\mathcal{E}(e) = \{(\bar{i}, v, dep), (\bar{j}, v, dep)\}.$$

We are now in the position of formulating the headway constraints. Namely, for each edge e we require for all $i, j \in \mathcal{E}(e)$ that either $x_i \geq H_{ji} + x_j$ or $x_j \geq H_{ij} + x_i$. Since

$$\left| x_j - x_i + \frac{H_{ji} - H_{ij}}{2} \right| \geq \frac{H_{ji} + H_{ij}}{2} \Longleftrightarrow x_j - x_i \geq H_{ij} \text{ or } x_i - x_j \geq H_{ji}$$

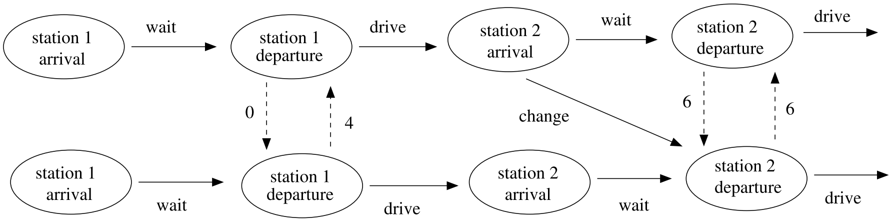

Fig. 4 Graphical interpretation of the capacity constraints

we can equivalently require that

$$\left| x_j - x_i + \frac{H_{ji} - H_{ij}}{2} \right| \geq \frac{H_{ji} + H_{ij}}{2} \quad \text{for all } i, j \in \mathcal{E}(e). \tag{5}$$

Consequently, the capacitated delay management model is the following.

$$\text{(Cap-DM)} \quad \min f(x,z) = \sum_{i \in \mathcal{E}} w_i (x_i - \Pi_i) + \sum_{a \in \mathcal{A}_{change}} w_a T z_a$$

such that (1), (2), (3), (5) are satisfied,

$x_i \in \mathbb{N}$ for all $i \in \mathcal{E}$, $z_a \in \{0,1\}$ for all $a \in \mathcal{A}_{change}$.

For the interpretation of the capacity constraints recall that a directed edge $a = (i, j)$ in the event-activity network fixes the order of the two events i and j by requiring $x_j > x_i + L_a$, where L_a is the given minimal duration of activity a. The capacity constraints can be interpreted analogously: For each pair of events i, j either $x_i \geq x_j + H_{ji}$ or $x_j \geq x_i + H_{ij}$ has to be satisfied—but it is not clear in advance which of the two disjunctive constraints should be satisfied and which not. Figure 4 shows the graphical interpretation of this fact: While the black 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 headway constraints, indicated as weights H_{ij} of edge (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. If $|\mathcal{E}(e)| = n$ there are $n!$ cycle-free solutions, each corresponding to one fixed order of these n events.

We remark that the problem of finding a feasible solution can also be seen as an edge orientation problem, where a graph G with a set of directed and a set of undirected edges is given, and one has to orient the undirected edges in such a way that no directed cycles occur.

We further remark that for fixed variables z_a our problem (Cap-DM) becomes a machine scheduling problem. More precisely, interpreting tracks as machines and trains as jobs yields a job-shop scheduling problem, but with additional precedence constraints between different jobs (corresponding to variables z_a which have been fixed to 0). This variant is called (Cap) and will be further analyzed in Sect. 5.

We now discuss the relation between (DM) and (Cap-DM). Let x^*, z^* be an optimal solution of (DM) and x^C, z^C be an optimal solution of (Cap-DM), with objective values y^* and y^C, respectively. Since (DM) is a relaxation of (Cap-DM) we conclude $y^* \le y^C$. We now identify cases in which an optimal solution of (DM) also solves (Cap-DM), i.e. cases, in which the headway constraints can be neglected. To this end, fix some edge $e \in E$ and consider the set $\mathcal{E}(e)$ of all events occurring along e.

Lemma 1 (Inverse triangle inequality for headways) *Let $i, j, k \in \mathcal{E}(e)$. For the corresponding headways we have*

$$H_{ik} \le H_{ij} + H_{jk}.$$

Proof As before, let $\bar{i}$ denote the train corresponding to event i. Recall that the headway H_{ij} is the smallest time such that the following holds: If $x_j > x_i + H_{ij}$ the train $\bar{j}$ of event j can follow the train $\bar{i}$ of event i and there will be no conflict between the trains.

We now consider H_{ik}, i.e. the minimal time, train $\bar{k}$ has to wait until it can follow train $\bar{i}$. We know that train $\bar{j}$ can follow train $\bar{i}$ after H_{ij} minutes and that train $\bar{k}$ can follow train $\bar{j}$ after H_{jk} minutes. Conflicts can only disappear if we remove train $\bar{j}$, so train $\bar{k}$ can follow train $\bar{i}$ after $H_{ij} + H_{jk}$ minutes, yielding that $H_{ik} \le H_{ij} + H_{jk}$. □

The next result shows that the capacity constraints are transitive.

Theorem 1 *Let $x_1 < x_2 < x_3$. If*

$$\left| x_2 - x_1 + \frac{H_{21} - H_{12}}{2} \right| \ge \frac{H_{21} + H_{12}}{2},$$

$$\left| x_3 - x_2 + \frac{H_{32} - H_{23}}{2} \right| \ge \frac{H_{32} + H_{23}}{2}$$

then also $|x_3 - x_1 + \frac{H_{31} - H_{13}}{2}| \ge \frac{H_{31} + H_{13}}{2}$ is satisfied.

Proof $x_i < x_j$ and $-(x_j - x_i - \frac{H_{ji} - H_{ij}}{2}) \ge \frac{H_{ji} + H_{ij}}{2}$ cannot be satisfied at the same time. Hence, we rewrite the assumptions to

$$x_2 - x_1 \ge H_{12},$$

$$x_3 - x_2 \ge H_{23}.$$

Adding these constraints and using Lemma 1 yields $x_3 - x_1 \ge H_{12} + H_{23} \ge H_{13}$, from which we conclude the required result. □

The result shows that we only have to check the headways of events that follow each other in the disposition timetable to make sure that all headway constraints are taken into account. We now introduce the *headway slack* with respect to events i and j.

Definition 2 Let $H_{ij} > 0$. The *headway slack* with respect to events i and j is defined as

$$S^{ij} := \Pi_j - \Pi_i - H_{ij}.$$

Then the following holds.

Lemma 2 *If* $x_i - \Pi_i - (x_j - \Pi_j) \le S^{ij}$, *then* (5) *is satisfied for events* i, j.

Proof

$$x_i - x_j - \Pi_i + \Pi_j \le S^{ij} \iff x_i - x_j \le -H_{ij}$$
$$\iff x_j - x_i \ge H_{ij},$$

hence, $|x_j - x_i + \frac{H_{ji}-H_{ij}}{2}| \ge \frac{H_{ji}+H_{ij}}{2}$. □

Since the delay of an event i is $x_i - \Pi_i$ the lemma can be interpreted as follows. The headway between two events i and j is taken into account whenever the delay of event j minus the delay of event i is not larger than the headway slack S^{ij}. Note that this interpretation of S^{ij} as *headway slack* is only justified, if $\Pi_i < \Pi_j$, i.e. in the case that event i is scheduled before event j. But

$$S^{ji} = \underbrace{\Pi_i - \Pi_j}_{\le 0} - \underbrace{H_{ji}}_{\ge 0} < 0$$

in this case, and an interpretation of S^{ji} as headway slack does not make much sense. However, $-S^{ji}$ gives the minimum delay event i will have if the order of events i and j is reversed.

We can use the headway slacks for a quick check, if the headway constraints (5) need be considered in (Cap-DM): Namely, in the case that the maximum possible delay is smaller than the minimum headway slack the capacity constraints need not be considered explicitly. The same holds, if the delays for consecutively scheduled events increase from one event to the next. This is formalized in the following corollary.

Corollary 1 *Let* $\mathcal{E}(e)$ *be the set of all departure events with corresponding activities starting at edge* e.

- *Define* $S^e := \min_{i,j\in\mathcal{E}(e)} S^{i,j}$. *Let* (x, z) *be a feasible solution of* (Cap-DM) *with delays* $x_i - \Pi_i \le S^e$ *for all* $i \in \mathcal{E}(e)$. *Then* (5) *is satisfied for all* $i, j \in \mathcal{E}(e)$.
- *Now assume that* $\mathcal{E}(e)$ *is ordered w.r.t. the disposition timetable* x_i. *If* $x_{i+1} - \Pi_{i+1} \ge x_i - \Pi_i$ *for all* $i \in \mathcal{E}(e)$ *then* (5) *is satisfied for all* $i, j \in \mathcal{E}(e)$.

Proof Part 1 directly follows from Lemma 2. From Theorem 1 we know that only events that follow each other in the disposition timetable have to be considered. Together with Lemma 2 this shows part 2. □

Finally, simple calculations show the following observations.

Lemma 3 *For all events* i, j *we have*

1. $H_{ji} + H_{ij} + S^{ji} + S^{ij} = 0$, *and*
2. $-S^{ji} \ge S^{ij}$.

5 Solution approach and numerical results

The idea of our solution approaches is to replace each of the disjunctive headway constraints

$$\left| x_j - x_i + \frac{H_{ji} - H_{ij}}{2} \right| \geq \frac{H_{ji} + H_{ij}}{2}$$

by one simple *precedence constraint*, namely either by the constraint

$$x_j - x_i \geq H_{ij}, \tag{6}$$

or by the constraint

$$x_i - x_j \geq H_{ji}. \tag{7}$$

To precisely state our algorithms, we need the following two simplified versions of (Cap-DM). For the first, we assume that all wait-depart decisions are known, and define

$$\mathcal{A}^{fix} = \mathcal{A}_{wait} \cup \mathcal{A}_{drive} \cup \{a \in \mathcal{A}_{change} : z_a = 0\}.$$

In this case, (Cap-DM) reduces to the following *re-scheduling* problem.

$$\text{(Cap)} \quad \min f(x) = \sum_{i \in \mathcal{E}} w_i (x_i - \Pi_i)$$

such that

$$x_i \geq \Pi_i + d_i \quad \text{for all } i \in \mathcal{E} \tag{8}$$

$$x_j - x_i \geq L_a \quad \text{for all } a = (i, j) \in \mathcal{A}^{fix} \tag{9}$$

$$\left| x_j - x_i + \frac{H_{ji} - H_{ij}}{2} \right| \geq \frac{H_{ji} + H_{ij}}{2} \quad \text{for all } i, j \in \mathcal{E}(e) \tag{10}$$

$$x_i \in \mathbb{N} \quad \text{for all } i \in \mathcal{E}$$

Note that this formulation includes the constraints (1), (5) that we already had in (Cap-DM), but neglects the binary variables z_a.

Further neglecting the capacity constraints (5) we get

$$\text{(Basic)} \quad \min f(x) = \sum_{i \in \mathcal{E}} w_i (x_i - \Pi_i)$$

such that

$$x_i \geq \Pi_i + d_i \quad \text{for all } i \in \mathcal{E}$$

$$x_j - x_i \geq L_a \quad \text{for all } a = (i, j) \in \mathcal{A}^{fix}$$

$$x_i \in \mathbb{N} \quad \text{for all } i \in \mathcal{E}$$

Note that (Basic) can be solved efficiently by the forward phase of the critical path method (CPM), see (4). Adding a precedence constraint of type (6) or (7) does not change the structure of (Basic) such that the problem can still be solved by the following slightly modified approach. To this end let $\mathcal{A}^{\text{headway}}$ be the headway constraints $x_j - x_i \geq H_{ij}$ included in (Basic). Then we have to calculate

$$x_1 := \Pi_1 + d_1$$
$$x_{i+1} := \max\left\{\Pi_{i+1} + d_{i+1}, \max_{(a=(j,i)\in\mathcal{A}} x_j + L_a \right. \tag{11}$$
$$\left. \max_{a=(j,i)\in\mathcal{A}^{\text{headway}}} x_j + H_{ji}\right\}, \quad i = 2, \dots, n.$$

Before we can present the heuristic approaches, we need the following result for the case of only two events. To this end recall the headway slack $S^{ij} = \Pi_j - \Pi_i - H_{ij}$, see Definition 2.

Lemma 4 *Consider two events i and j scheduled at times Π_i, Π_j, and a pair of disjunctive capacity constraints according to* (5), *i.e.*,

$$x_i - x_j \geq H_{ij} \quad or \quad x_j - x_i \geq H_{ji}.$$

Furthermore, define

$$\eta_{ij} := \frac{w_j S^{ij} - w_i S^{ji}}{w_i + w_j} \quad \left(= \Pi_j - \Pi_i + \frac{w_i H_{ji} - w_j H_{ij}}{w_i + w_j}\right).$$

Then for given (source) delays d_i, d_j the optimal solution of (Cap) *satisfies*

- *event i is scheduled before j if $d_i - d_j < \eta_{ij}$,*
- *event i is scheduled after j if $d_i - d_j > \eta_{ij}$.*

Further, the optimal solution of (Cap) *is given as follows*:

$$x_i = \Pi_i + d_i, \qquad x_j = \max\{\Pi_i + d_i + H_{ij}, \Pi_j + d_j\} \quad \text{if } d_i - d_j < \eta_{ij},$$
$$x_i = \max\{\Pi_j + d_j + H_{ji}, \Pi_i + d_i\}, \qquad x_j = \Pi_j + d_j \quad \text{if } d_i - d_j > \eta_{ij}$$

For $d_i - d_j = \eta_{ij}$ both solutions are optimal.

Proof Let $i = (\bar{i}, u, dep)$, $j = (\bar{j}, u', dep)$ with trains $\bar{i}$ and $\bar{j}$. We set up the following table which specifies the delay of train $\bar{i}$ and of train $\bar{j}$ dependent on the order of the trains.

	delay train of $\bar{i}$	delay of train $\bar{j}$
$\bar{i}$ departs before $\bar{j}$	d_i	$\max\{d_i - S^{ij}, d_j\}$
$\bar{j}$ departs before $\bar{i}$	$\max\{d_j - S^{ji}, d_i\}$	d_j

As objective function value we hence obtain

i before j

$$f(x) = \begin{cases} d_i(w_i + w_j) - w_j S^{ij} & \text{if } d_i - d_j > S^{ij}, \\ d_i w_i + d_j w_j & \text{if } d_i - d_j \le S^{ij} \end{cases}$$

j before i

$$f(x) = \begin{cases} d_j(w_i + w_j) - w_i S^{ji} & \text{if } d_j - d_i \ge S^{ji}. \\ d_i w_i + d_j w_j & \text{if } d_j - d_i < S^{ji} \end{cases}$$

Using that $S^{ij} \le -S^{ji}$ (see second part of Lemma 3) it is sufficient to distinguish the following three cases:

Case 1. $-S^{ji} < d_i - d_j$: In this case we obtain

$$d_i(w_i + w_j) - w_j S^{ij} \le w_i d_i + w_j d_j$$
$$\Longleftrightarrow d_i - d_j \le S^{ij}$$

which never occurs due to the assumption of case 1. Hence, in case 1, the minimum is attained if j takes place before i.

Case 2. $S^{ij} < d_i - d_j \le -S^{ji}$: In this case we obtain

$$d_i(w_i + w_j) - w_j S^{ij} \le d_j(w_i + w_j) - w_i S^{ji}$$
$$\Longleftrightarrow d_i - d_j \le \eta_{ij},$$

and it holds that $S^{ij} \le \eta_{ij} \le -S^{ji}$. Hence, i should be scheduled before j if $d_i - d_j \le \eta_{ij}$, otherwise j should take place before i.

Case 3. $d_i - d_j < S^{ij}$: In the remaining case we obtain

$$w_i d_i + w_j d_j \le d_j(w_i + w_j) - w_i S^{ji}$$
$$\Longleftrightarrow d_i - d_j \le -S^{ji},$$

which is always satisfied in this case and hence i should go before j.

The result of the lemma follows by calculating the earliest starting times in (Basic), with one additional precedence constraint:

- In the case that $d_i - d_j \le \eta_{ij}$ we add the precedence constraint $x_j - x_i \ge H_{ij}$ to (Basic).
- The precedence constraint $x_i - x_j \ge H_{ji}$ is added to (Basic) in the case that $d_i - d_j > \eta_{ij}$. □

Corollary 2 *For the simple case of two events i, j with scheduled times $\Pi_i < \Pi_j$, and equal weights $w_i = w_j$ we get*:

- *If there is only one source delay $d_i > 0$ for event i we obtain*:
  If $d_i \le \Pi_j - \Pi_i$: do not change the order (i.e. schedule i before j), otherwise change the order (i.e. schedule j and then i).

- *If the headways are equal, we obtain the rule first-come-first-served: If* $x_i \le x_j$ *: schedule* i *before* j*, otherwise schedule* j *first and then* i.

Note that adding a precedence constraint of type (6) or (7), i.e. something like $x_i - x_j \ge H_{ji}$ to the program (DM) does not change the structure of (DM) since it is just one more constraint of type (2). Hence, we can use any algorithm for (DM) to solve capacitated problems, if the headway constraints have been replaced by simple precedence constraints. Our first approach makes use of this fact.

Our first heuristic *first-scheduled-first-served (FSFS)* is motivated by the result of Corollary 2: In a first step we fix the headway constraints according to the originally planned schedule, and in a second step we solve the remaining delay management problem with additional precedence constraints. We obtain:

Heuristic First-Scheduled-First-Served

Input: `(Cap-DM)`
Step 1: `For all` $e \in E$ `and all` $i, j \in \mathcal{E}(e)$ `add one of the following constraints to (DM):`

$$\begin{cases} x_j - x_i \ge H_{ij} & \texttt{if} \quad \Pi_i \le \Pi_j, \\ x_i - x_j \ge H_{ji} & \texttt{if} \quad \Pi_j < \Pi_i \end{cases}$$

Step 2: `Solve (DM) together with the new constraints using branch & bound.`

As justified by part 1 of Corollary 2, Heuristic FSFS is reasonable for smaller delays.

The second approach called **first-rescheduled-first-served (FRFS)** proceeds the other way round: First the uncapacitated delay management problem is solved, then a re-scheduling phase with fixed precedence constraints according to the optimal solution of (DM) is added.

Heuristic First-Rescheduled-First-Served

Input: `(Cap-DM)`
Step 1: `Solve the corresponding (DM) without capacity constraints. Let` x, z `be an optimal solution. Let` $A^{fix} = \{a \in \mathcal{A}_{change} : z_a = 0\}$.
Step 2: `For each` $e \in E$ `and each pair` $i, j \in \mathcal{E}(e)$ `add one of the following precedence constraints to (Basic):`

$$\begin{cases} x_j - x_i \ge H_{ij} & \texttt{if} \quad x_i \le x_j, \\ x_i - x_j \ge H_{ji} & \texttt{if} \quad x_j < x_i \end{cases}$$

Step 3: `Solve (Basic) with the precedence constraints added in Step 2.`
Output: `A feasible solution` (x, z)

We finally use the result of Lemma 4 to decide about the precedence constraint to add while solving the uncapacitated delay management problem. Before we do so, we present the following observations.

Lemma 5 *The following hold.*

1. $-\eta_{ij} = \eta_{ji}$
2. $d_i - d_j \leq \eta_{ij} \iff d_j - di \geq \eta_{ji}$, *i.e. the decision from the pair* i, j *is also optimal for the pair* j, i.
3. *For constant weights and constant headways, the following holds: If* $d_i - d_j \leq \eta_{ij}$ *and* $d_j - d_k \leq \eta_{jk}$ *then* $d_k - d_i \geq \eta_{ki}$, *i.e. the decision taken from two headway constraints is valid for the third one. This does not hold in general.*

Proof Statement 1 follows directly from the definition and statement 2 from statement 1. For the third statement, let $d_i - d_j \leq \eta_{ij}$ and $d_j - d_k \leq \eta_{jk}$. For arbitrary headways and arbitrary weights we then get

$$\begin{aligned}
d_k - d_i &= -(d_j - d_k) - (d_i - d_j) \\
&\geq -\eta_{jk} - \eta_{ij} \\
&= -\Pi_k + \Pi_j - \frac{w_j H_{kj} - w_k H_{jk}}{w_j + w_k} - \Pi_j + \Pi_i - \frac{w_i H_{ji} - w_j H_{ij}}{w_i + w_j} \\
&= \Pi_i - \Pi_k + \frac{w_k H_{jk} - w_j H_{kj}}{w_j + w_k} + \frac{w_j H_{ij} - w_i H_{ji}}{w_i + w_j} \\
&= \eta_{ki} - \frac{w_k H_{ik} - w_i H_{ki}}{w_k + w_i} + \frac{w_k H_{jk} - w_j H_{kj}}{w_j + w_k} + \frac{w_j H_{ij} - w_i H_{ji}}{w_i + w_j}.
\end{aligned}$$

Hence it holds,

$$d_k - d_i \geq \eta_{ki} \iff \frac{w_k H_{jk} - w_j H_{kj}}{w_j + w_k} + \frac{w_j H_{ij} - w_i H_{ji}}{w_i + w_j} \geq \frac{w_k H_{ik} - w_i H_{ki}}{w_k + w_i}.$$

This is true for equal headways and equal weights, and e.g. not true for equal headways, $w_j = 0$ and $w_k > w_i$. □

The next heuristic uses branch & bound together with the optimal scheduling decision according to Lemma 4. It is called (B&B-OS).

Branch & Bound for (Cap-DM) including optimal-served (B&B-OS)

Input: `(Cap-DM) Use the branch & bound approach from page 139 with the following modifications:`

- `In Step 3, use a heuristic which fixes the headway constraints as in the rule given in Lemma 4. If the weights and headways are all equal, the order of these decisions is not`

```
relevant (part 3 of Lemma 5). In the other cases, fix the
precedence constraints in the natural order of the events.
• In Step 4, use the LP-relaxation of the corresponding (pure)
delay management problem as relaxation. In 4.1 test not only
if the z_a variables are boolean, but also if the capacity
constraints are satisfied.
```

We tested FSFS and B&B–OS on real-world data from the region of Harz, Germany. The data consists of 183 stations, 1962 trains, and roughly 8400 connections. We assumed equal headways for each edge, and equal weights, such that B&B–OS reduces to first-come-first-served according to Corollary 2.

As headways we considered four cases corresponding to the four columns a, b, c, d in the tables. Case (a) neglects the headway (i.e. considers headways of zero). In the other cases (b), (c), and (d), we proceed as follows: For edge e we use a headway of

$$\min\{\Pi_j - \Pi_i : i, j \in \mathcal{E}(e), H_e\},$$

where $H_e = 3, 5,$ and 10 minutes for case (b), (c), and (d), respectively. The delay scenarios we consider are typical examples from practice. They consist of 1, 3, or 5 source delays, as indicated in the first column of the tables.

Two observations can be seen from the tables: First, the branch & bound approach using Lemma 4 performs equally good or better in all our examples. Second, the headway constraints do not influence the objective function value as worse as one might have expected. In many cases the optimal objective value does not change at all, and if it changes, the increase of the objective is usually not too high.

Table 1 Results for FSFS for different headways

No. of source delays	sum of delays				dropped connec.				objective function			
	a	b	c	d	a	b	c	d	a	b	c	d
1	107.2	107.2	107.2	107.2	1	1	1	1	167.2	167.2	167.2	167.2
1	212.8	212.8	212.8	220.2	0	0	0	0	212.8	212.8	212.8	220.2
1	131.2	131.2	131.2	131.6	0	0	0	0	131.2	131.2	131.2	131.6
1	224.4	224.4	232.4	281.2	1	1	1	1	284.4	284.4	292.4	341.2
3	345.2	440.2	515.9	521.3	5	5	5	5	645.2	740.2	815.9	821.3
3	174.2	174.2	174.2	174.2	3	3	3	3	354.2	354.2	354.2	354.2
3	524.6	524.6	526.4	529.2	3	3	3	3	704.6	704.6	706.4	709.2
3	564.0	564.0	580.0	637.6	1	1	1	1	624.0	624.0	640.0	697.6
5	663.6	736.2	818.3	827.7	6	6	6	6	1023.6	1096.2	1178.3	1187.7
5	367.1	367.1	367.1	367.1	4	4	4	4	607.1	607.1	607.1	607.1
5	514.0	521.8	535.8	477.9	3	3	3	4	694.0	701.8	715.8	717.9
5	847.7	847.7	870.7	919.5	3	3	3	3	1027.7	1027.7	1050.7	1099.5

Table 2 Results for B&B–OS for different headways

No. of source delays	sum of delays				dropped connec.				objective function			
	a	b	c	d	a	b	c	d	a	b	c	d
1	107.2	107.2	107.2	107.2	1	1	1	1	167.2	167.2	167.2	167.2
1	212.8	212.8	212.8	220.2	0	0	0	0	212.8	212.8	212.8	220.2
1	131.2	131.2	131.2	131.6	0	0	0	0	131.2	131.2	131.2	131.6
1	224.4	224.4	232.4	281.2	1	1	1	1	284.4	284.4	292.4	341.2
3	290.0	290.0	290.0	291.2	5	5	5	5	590.0	590.0	590.0	591.2
3	174.2	174.2	174.2	174.2	3	3	3	3	354.2	354.2	354.2	354.2
3	524.6	524.6	526.4	529.2	3	3	3	3	704.6	704.6	706.4	709.2
3	564.0	564.0	580.0	637.6	1	1	1	1	624.0	624.0	640.0	697.6
5	630.6	632.6	651.4	659.0	6	6	6	6	990.6	992.6	1011.4	1019.0
5	367.1	367.1	367.1	367.1	4	4	4	4	607.1	607.1	607.1	607.1
5	514.0	521.8	535.8	477.9	3	3	3	4	694.0	701.8	715.8	717.9
5	807.7	807.7	815.7	864.5	3	3	3	3	987.7	987.7	995.7	1044.5

The running time of the approaches was within seconds for FSFS and within minutes for B&B–OS. Both heuristics are much faster than finding an optimal solution using an integer programming solver. This is in particular true when the network size increases. Such comparisons and a detailed numerical analysis of first-scheduled-first served and first-rescheduled-first served approaches are currently under research, see Schachtebeck and Schöbel (2009).

6 Conclusion and future work

In this paper we developed a model which allows to integrate capacity constraints in the delay management problem. We derived structural properties and first heuristic solution approaches. Further improvements of the presented algorithms (e.g. strengthening the relaxation in Step 4 and finding a good order of events when fixing the precedence constraints in Step 3) and exact solution approaches are under research. Moreover, the relation to job-shop scheduling problems is under investigation.

The model proposed can easily be extended to keep track of delays which are transfered from the end of one trip of a vehicle to the beginning of its next trip by adding *circulation activities*. They can be treated as waiting or driving activities in the solution approaches. The model can also be used as a partial model when e.g. the decisions for the long-distance trains have been fixed beforehand.

However, some extensions are still open and yield challenging future extensions of the delay management model. One of them is robustness of disposition timetables which is currently investigated within the European project ARRIVAL. For first results using the new concept of recovery robustness, we refer to Cicerone et al. (2008). Other interesting issues are to include the capacity constraints within stations and to allow a re-routing of passengers in the objective function.

Acknowledgements The author wants to thank Deutsche Bahn AG for providing the data, and C. Jobmann for implementing the algorithms.

References

Adenso-Díaz B, Oliva González M, González-Torre P (1999) On-line timetable re-scheduling in regional train services. Transp Res 33B:387–398

Billionnet A (2003) Using integer programming to solve the train platforming problem. Transp Sci 37(2):213–222

Bissantz N, Güttler S, Jacobs J, Kurby S, Schaer T, Schöbel A, Scholl S (2005) DisKon—Disposition und Konfliktlösungs-management für die beste Bahn. Eisenb Rundsch (ETR) 45(12):809–821 (in German)

Brucker P, Heitmann S, Knust S (2002) Scheduling railway traffic at a construction site. OR Spektr 24(1):19–30

Cicerone S, Di Stefano G, Schachtebeck M, Schöbel A (2008) Dynamic algorithms for recoverable robustness problems. In: Fischetti M, Widmayer P (eds) ATMOS 2008—8th workshop on algorithmic approaches for transportation modeling, optimization, and systems. Dagstuhl seminar proceedings

Cordeau J-F (1998) A survey of optimization models for train routing and scheduling. Transp Sci 32(4):380–404

De Giovanni L, Heilporn G, Labbé M (2008) Optimization models for the single delay management problem in public transportation. Eur J Oper Res 189(3):762–774

Gatto M, Glaus B, Jacob R, Peeters L, Widmayer P (2004) Railway delay management: Exploring its algorithmic complexity. In: Proc 9th Scandinavian workshop on algorithm theory (SWAT). LNCS, vol 3111, pp 199–211

Gatto M, Jacob R, Peeters L, Schöbel A (2005) The computational complexity of delay management. In: Kratsch D (ed) Graph-theoretic concepts in computer science: 31st international workshop (WG 2005). LNCS, vol 3787

Ginkel A, Schöbel A (2007) To wait or not to wait? The bicriteria delay management problem in public transportation. Transp Sci 41(4):527–538

Heidergott B, de Vries R (2001) Towards a control theory for transportation networks. Discrete Event Dyn Syst 11:371–398

Jacobs J (2004) Reducing delays by means of computer-aided ‘on-the-spot’ rescheduling. In: Computers in railways (Comprail) IX, pp 603–612

Jobmann C, Schöbel A (2007) Approaches for solving delay management problems. Project work within DisKon

Nachtigall K (1998) Periodic network optimization and fixed interval timetables. Deutsches Zentrum für Luft- und Raumfahrt, Institut für Flugführung, Braunschweig. Habilitationsschrift

Norio T, Yoshiaki T, Noriyuki T, Chikara H, Kunimitsu M (2005) Train rescheduling algorithm which minimizes passengers dissatisfaction. In: LNAI, vol 3533. Springer, Berlin, pp 829–838

Pachl J (2000) Safe disposition and scheduling in railway operation. Signal + Draht 92(5):38–41

Ryan DM, Foster BA (1981) An integer programming approach to scheduling. In: Wren A (ed) Computer scheduling of public transport urban passenger vehicle and crew scheduling. North-Holland, Amsterdam, pp 269–280

Schachtebeck M, Schöbel A (2009) To wait or not to wait and who goes first? Delay management with priority decisions. Working paper

Schöbel A (2001) A model for the delay management problem based on mixed-integer programming. Electron Notes Theor Comput Sci 50(1)

Schöbel A (2006) Customer-oriented optimization in public transportation. Optimization and its applications. Springer, New York

Schöbel A (2007) Integer programming approaches for solving the delay management problem. In: Algorithmic methods for railway optimization. LNCS, vol 4359. Springer, Berlin, pp 145–170

Szpigel B (1973) Optimal train scheduling on a single track railway. In: Ross M (ed) Operational research '72: Proceedings of the 6th IFORS international conference on operational research. North-Holland, Amsterdam, pp 343–352

Törnquist J (2005) Computer-based decision support for railway traffic scheduling and dispatching: A review of models and algorithms. In: Proceedings of ATMOS 2005

Velasquez R, Ehrgott M, Ryan D, Schöbel A (2005) A set-packing aproach to routing trains through railway stations. In: 40th annual conference of the operations research society of New Zealand, pp 305–314

Wegele S, Schnieder E (2005) Dispatching of train operations using genetic algorithms. In: 1st international seminar on railway operations modelling and analysis, Delft

Zwaneveld PJ (1996) Routing trains through railway stations: Model formulation and algorithms. Transp Sci 30:181–194

Zwaneveld PJ, Kroon LG, Romeijn HE, Salomon M, Dauzere-Peres S, van Hoesel SPM, Ambergen HW (1996) Routing trains through railway stations: Model formulation and algorithms. Transp Sci 30(33):181–194