Delay Management with Integrated Decisions on the Vehicle Circulations

Vera Grafe
Technische Universität Kaiserslautern, Germany

Alexander Schiewe
Fraunhofer-Institut für Techno- und Wirtschaftsmathematik ITWM, Kaiserslautern, Germany

Anita Schöbel
Technische Universität Kaiserslautern, Germany
Fraunhofer-Institut für Techno- und Wirtschaftsmathematik ITWM, Kaiserslautern, Germany

Abstract

The task of delay management in public transport is to decide whether a vehicle should wait for a delayed vehicle in order to maintain the connection for transferring passengers. So far, the vehicle circulations are often ignored in the optimization process, although they have an influence on the propagation of the delay through the network. In this paper we consider different ways from literature to incorporate vehicle circulations in the delay management stage of public transport planning. Since the IP formulation for the integrated problem is hard to solve, we investigate bounds and develop several heuristics for the integrated problem. Our experiments on close-to real-world instances show that integrating delay management and decisions on vehicle circulations may reduce the overall delay by up to 39 percent. We also compare the runtimes and objective function values of the different heuristics. We conclude that we can find competitive solutions in a reasonable amount of time.

2012 ACM Subject Classification Applied computing → Transportation

Keywords and phrases Public Transport, Delay Management, Vehicle Circulations, Integer Programming

Digital Object Identifier 10.4230/OASIcs.ATMOS.2022.7

1 Introduction

Public transportation plays an essential role in passenger mobility. An important factor for the satisfaction of the passengers, and therefore also for the economic success of the transportation company, is reliability. However, guaranteeing this is not an easy task: Unplanned disturbances are inevitable in transportation networks. Because of numerous interdependencies, these can have a huge impact on the overall network. Hence, an important task in everyday business is to react to disturbances in the best possible way. The most crucial decision to be made in this context is whether a vehicle should wait for a delayed feeder vehicle. If the connecting vehicle does not wait for the feeder vehicle, the passengers on the latter wishing to transfer miss their connection and have to wait for the next ride. Especially in networks with a low frequency this is very frustrating for the passengers. On the other hand, waiting for the delayed vehicle adds further delay in the network, since all passengers on the connecting vehicle are then also affected by the delay and maybe even miss a later transfer themselves. This way, the delay can propagate through the entire network. Hence, the task of *delay management* is to make waiting decisions and find a feasible *disposition timetable* keeping the passengers' dissatisfaction to a minimum.
A further aspect which has to be considered are the *vehicle schedules* or *rolling stock circulations*: If a vehicle arrives at the final destination of a trip with a delay and is scheduled to serve another trip subsequently, it is possible that the latter cannot start on time. This

also may propagate the delay. This type of delay propagation has only been sparsely treated in the literature. In this paper, we suggest an approach to mitigate this effect, namely by replanning the circulations of the vehicles: If there is another vehicle available, it can serve the second trip, possibly even without delay. In order to make use of such a rescheduling, we integrate the planning of vehicle circulations into the delay management problem.

Literature review. Delay Management has been studied extensively over the last two decades. One of the first models, which is based on mixed-integer programming, was introduced in [22, 23]. Different extensions to this model have been made. In [16] the limited capacity of the track system was taken into account by adding headways to the integer programming formulation and presenting heuristic solution approaches. The capacity of the stations was considered in [4]. These models rest on the assumption that the passengers continue their journey as planned in case of delays. In reality, passengers might adapt their routes to the current situation. This possibility was first considered in [6]. Heuristics for solving the problem with re-routing were presented in [3] and a software tool was introduced in [14]. For literature reviews on delay management, we refer to [12, 7].

Integrating delay management on a macroscopic level and train scheduling on a microscopic level was studied in [2]. In [5] a sequential approach for rescheduling timetable, rolling stock and crew was presented. Integrating rescheduling of vehicle circulations and delay management was considered in [11], where especially the rolling stock constraints are modelled in detail but wait/no-wait decision with regard to passenger transfers are not considered. First ideas for integrating vehicle circulation planning in delay management have been sketched in [8, 15]. For an overview on vehicle scheduling, see [1].

Contribution of the paper. We consider three different models from literature: First, the classic delay management problem where the vehicle circulations are ignored. Second, a model which respects the planned vehicle circulations taking into account that delay is propagated along the vehicles' circulations. Third, we present an integrated model in which the circulations are re-optimized within delay management. We analyze the models and their relations and give bounds on the optimal objective values of the integrated formulation. We also develop three heuristics to solve it and evaluate our approaches on close-to real-world datasets.

2 Including vehicle circulations in delay management

In this section, we present integer programming formulations for three different models for the delay management problem, differing in the extent to which the vehicle circulations are considered.

All three models take an *event-activity-network* $\mathcal{N}_{\text{pure}} = (\mathcal{E}, \mathcal{A}_{\text{pure}})$ based on a set of trips $\mathcal{T}$ and a set of stations $v \in V$ as input. A *trip* $t \in \mathcal{T}$ represents a section that needs to be served by a single vehicle, e.g., a line. *Arrival* events and *departure* events are given by

$$\mathcal{E}_{\text{arr}} = \{(v, t, \text{arr}) : t \in \mathcal{T} \text{ arrives at } v \in V\} \text{ and } \mathcal{E}_{\text{dep}} = \{(v, t, \text{dep}) : t \in \mathcal{T} \text{ departs at } v \in V\}$$

and $\mathcal{E} = \mathcal{E}_{\text{arr}} \cup \mathcal{E}_{\text{dep}}$. We assume that a timetable $\pi \in \mathbb{N}^{|\mathcal{E}|}$ is given and fixed, assigning the scheduled time π_i to event $i \in \mathcal{E}$. The events are connected by the activities $\mathcal{A}_{\text{pure}} := \mathcal{A}_{\text{drive}} \cup \mathcal{A}_{\text{wait}} \cup \mathcal{A}_{\text{transfer}}$, where

$$\begin{aligned}
\mathcal{A}_{\text{drive}} &:= \{((v_1,t,\text{dep}),(v_2,t,\text{arr})) \in \mathcal{E}_{\text{dep}} \times \mathcal{E}_{\text{arr}} \colon t \text{ serves } v_2 \text{ directly after } v_1\},\\
\mathcal{A}_{\text{wait}} &:= \{((v,t,\text{arr}),(v,t,\text{dep})) \in \mathcal{E}_{\text{arr}} \times \mathcal{E}_{\text{dep}}\},\\
\mathcal{A}_{\text{transfer}} &:= \{((v,t_1,\text{arr}),(v,t_2,\text{dep})) \in \mathcal{E}_{\text{arr}} \times \mathcal{E}_{\text{dep}} \colon \text{there is a transfer from } t_1 \text{ to } t_2 \text{ at } v\},
\end{aligned}$$

are the *drive*, *wait* and *transfer* activities, respectively. Each activity $a \in \mathcal{A}_{\text{pure}}$ has a corresponding lower bound $L_a \in \mathbb{N}$, the minimal time needed to perform the activity. Note that if delays occur, possible upper bounds for activities often cannot be respected any more, which is why we ignore them and only consider the lower bounds. A timetable is feasible if it respects the bounds on the activities, i.e., if for all $a = (i,j) \in \mathcal{A}_{\text{pure}}$ $\pi_j - \pi_i \geq L_a$ holds.

When dealing with delays, we distinguish the following two types of delays:

- *Source delays* are caused by external factors, e.g., damaged tracks.
- *Propagated delays* evolve within the transportation system. E.g., if a train arrives at a station with a delay and hence also departs with a delay, then this departure delay has been propagated along the waiting activity and is called *propagated delay*.

The goal of delay management is to adapt to these delays, i.e., to find a *disposition timetable* $x \in \mathbb{N}^{|\mathcal{E}|}$ where x_i denotes the time of event $i \in \mathcal{E}$. When determining the disposition timetable, we have to make two different kinds of decisions:

- Wait/depart decisions: For every transfer $a \in \mathcal{A}_{\text{transfer}}$ we have to decide whether or not it should be maintained.
- Circulation decisions: We have to decide which vehicle operates which trip.

Note that for train transportation also headway activities need to be considered. For the sake of simplicity these are neglected here, but could be added easily to all three models to be presented.

In order to handle the circulation decisions, we define *circulations* $c = (t_{c_1}, \ldots, t_{c_n})$ consisting of a list of trips which are operated consecutively by the same vehicle. Since an event i uniquely determines its corresponding trip, we can model the circulations by defining the following new types of events and activities,

$$\begin{aligned}
\mathcal{E}_{\text{first}} &:= \{i \in \mathcal{E}_{\text{dep}} \colon i \text{ is the first departure of a circulation}\}\\
\mathcal{E}_{\text{last}} &:= \{i \in \mathcal{E}_{\text{arr}} \colon i \text{ is the last arrival of a circulation}\}\\
\mathcal{E}_{\text{start}} &:= \{i \in (\mathcal{E}_{\text{dep}} \backslash \mathcal{E}_{\text{first}}) \colon i \text{ is the first event of a trip}\}\\
\mathcal{E}_{\text{end}} &:= \{i \in (\mathcal{E}_{\text{arr}} \backslash \mathcal{E}_{\text{last}}) \colon i \text{ is the last event of a trip}\}\\
\mathcal{A}_{\text{circ}} &:= \{((v_1,t_1,\text{arr}),(v_2,t_2,\text{dep})) \in \mathcal{E}_{\text{end}} \times \mathcal{E}_{\text{start}} \colon t_1 \text{ and } t_2 \text{ can be operated within}\\
&\qquad \text{the same circulation}\}
\end{aligned}$$

and the set of all activities as $\mathcal{A} := \mathcal{A}_{\text{pure}} \cup \mathcal{A}_{\text{circ}}$. The EAN including the circulation activities is denoted by $\mathcal{N} = (\mathcal{E}, \mathcal{A})$. From the set of all possible circulation activities a subset has to be chosen such that for every $i \in \mathcal{E}_{\text{start}} \cup \mathcal{E}_{\text{end}}$ there is exactly one ingoing and one outgoing activity. We set $\mathcal{A}^{\text{fix}}_{\text{circ}} := \{a \in \mathcal{A}_{\text{circ}} \colon a \text{ is chosen}\}$. Note that the circulations do not refer to cycles in the event-activity-network which is an acyclic time-expanded network. We are not requiring periodic networks here.

Apart from the EAN, the input data contain the source delays and passenger weights. We have two types of source delays, namely source delays d_i at events $i \in \mathcal{E}$ and source delays d_a on activities $a \in \mathcal{A}_{\text{train}} := \mathcal{A}_{\text{drive}} \cup \mathcal{A}_{\text{wait}}$. An event may just start late for some external reason, e.g., a driver coming too late to work, while an activity may have a longer duration as anticipated, e.g., due to a speed reduction on a piece of a track. Correspondingly, for the

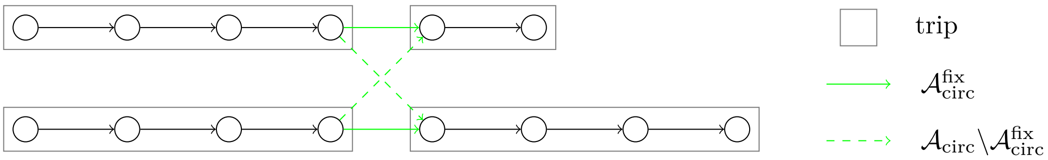


Figure 1 An EAN with circulation activities.

passengers weights we use w_i for $i \in \mathcal{E}$ as the number of passengers reaching their destination at event i. For $a \in \mathcal{A}_{\text{transfer}}$ the number of transferring passengers is given by w_a. The total sum of passengers' delays can be approximated by only using the values w_a and w_i as shown in [23]. We do not consider routing decisions depending on the disposition timetable.

We consider three different models for the delay management problem. In all of them the goal is to find a disposition timetable minimizing the total delay of all passengers. The first model is the usual model for delay management (see [23]) which ignores that delays may be propagated along circulation activities and concentrates on the wait/depart decisions. We call this model (DM). However, the resulting disposition timetable might not be operable if delay is propagated along circulation activities. In the second model, (DM-fix), the circulation activities are fixed beforehand and respected when computing the disposition timetable. It finds a disposition timetable which can be operated. In the third model, (DM-opt), we go a step further and allow that the circulation activities may be changed if this fits better to the current situation. This means we optimize the circulations while computing the disposition timetable. In order to formulate the models as integer programs, we encode the decisions to be made as binary variables,

$$y_a = \begin{cases} 1 & \text{if the transfer } a \text{ is cancelled} \\ 0 & \text{otherwise} \end{cases}$$

for the transfer activities $a \in \mathcal{A}_{\text{transfer}}$ and

$$v_{(i,j)} = \begin{cases} 1 & \text{if circulation activity } (i,j) \text{ is chosen} \\ 0 & \text{otherwise} \end{cases}$$

for the circulation activities $(i,j) \in \mathcal{A}_{\text{circ}}$. Note that the second set of variables is only present in (DM-opt).

2.1 The classic delay management formulation (DM)

The formulation for the first model, (DM), is the "classic" delay management first been proposed in [22].

$$\begin{aligned} z = \min \quad & \sum_{i \in \mathcal{E}} w_i(x_i - \pi_i) + T \sum_{a \in \mathcal{A}_{\text{transfer}}} w_a y_a && && \text{(DM)} \\ \text{s.t.} \quad & x_i \geq \pi_i + d_i && i \in \mathcal{E} && (1) \\ & x_j - x_i \geq L_a + d_a && a = (i,j) \in \mathcal{A}_{\text{train}} && (2) \\ & M_1 y_a + x_j - x_i \geq L_a && a = (i,j) \in \mathcal{A}_{\text{transfer}} && (3) \\ & x_i \in \mathbb{N} && i \in \mathcal{E} && (4) \\ & y_a \in \{0,1\} && a \in \mathcal{A}_{\text{transfer}}. && (5) \end{aligned}$$

The objective function minimizes the approximated total delay the passengers have at their final destination. If a passenger misses a transfer, we assume that after a time period T everything is on time again and we therefore penalize the missed transfer accordingly in the second sum of the objective. The constraints (1) ensure that the source delays of the events are respected by the disposition timetable. The propagation of the delays along the activities $a \in \mathcal{A}_{\text{pure}}$ is enforced by constraints (2) and (3). For $a \in \mathcal{A}_{\text{transfer}}$ this is only necessary if the transfer is maintained, which is why we have the big-M-constraints here.

2.2 Delay management with fixed circulations (DM-fix)

In the second model, (DM-fix), the originally planned circulation activities $\mathcal{A}_{\text{circ}}^{\text{fix}}$ are included.

$$
\begin{aligned}
z^{\text{fix}} = \min \quad & \sum_{i \in \mathcal{E}} w_i(x_i - \pi_i) + T \sum_{a \in \mathcal{A}_{\text{transfer}}} w_a y_a & & \text{(DM-fix)} \\
\text{s.t.} \quad & x_j - x_i \geq L_a & a = (i,j) \in \mathcal{A}_{\text{circ}}^{\text{fix}} \quad & (6) \\
& (1) - (5).
\end{aligned}
$$

For (DM-fix) the objective function and constraints (1) to (5) are the same as for (DM) while constraints (6) make sure that the circulation activities (chosen beforehand) are respected. Note that in [7] the circulation activities are called *turn-around activities* $\mathcal{A}_{turn}$.

2.3 Integrating delay management and vehicle scheduling (DM-opt)

Finally, in the third model, (DM-opt), we allow to make the decisions about the circulations together with the decisions in delay management.

$$
\begin{aligned}
z^{\text{opt}} = \min \quad & \sum_{i \in \mathcal{E}} w_i(x_i - \pi_i) + T \sum_{a \in \mathcal{A}_{\text{transfer}}} w_a y_a & & \text{(DM-opt)} \\
\text{s.t.} \quad & M_2(1 - v_{ij}) + x_j - x_i \geq L_a & a = (i,j) \in \mathcal{A}_{\text{circ}} \quad & (7) \\
& \sum_{i \in \mathcal{E}_{\text{end}}:(i,j) \in \mathcal{A}_{\text{circ}}} v_{ij} = 1 & j \in \mathcal{E}_{\text{start}} \quad & (8) \\
& \sum_{j \in \mathcal{E}_{\text{start}}:(i,j) \in \mathcal{A}_{\text{circ}}} v_{ij} = 1 & i \in \mathcal{E}_{\text{end}} \quad & (9) \\
& v_{ij} \in \{0,1\} & (i,j) \in \mathcal{A}_{\text{circ}} \quad & (10) \\
& (1) - (5).
\end{aligned}
$$

The IP for the third model (DM-opt) (sketched in [8]) also contains all the constraints from (DM). Additionally, we have constraints (7) to incorporate the delay propagation along the circulation activities. Let $i = (v_1, t_1, arr) \in \mathcal{E}_{\text{end}}$ and $j = (v_2, t_2, dep) \in \mathcal{E}_{\text{start}}$ be the end and the start event of two trips. Then choosing $v_{(i,j)} = 1$ means that t_1 and t_2 appear consecutively in a circulation, i.e., the vehicle operating the trip t_1 proceeds from its last station v_1 of trip t_1 to the first station v_2 of trip t_2 and then operates trip t_2. Often $v_1 = v_2$, i.e., t_1 ends at the same station from which t_2 departs. In case that $v_1 \neq v_2$ we have an *empty trip* between the two stations. In contrast to (DM-fix), the circulation activities are not fixed beforehand, but are determined in the optimization process. We hence have another type of big-M-constraints here. It was shown in [15] how to compute reasonable values for M_1 and M_2. Furthermore, we have to ensure that for every $i \in \mathcal{E}_{\text{end}}$ and every $j \in \mathcal{E}_{\text{start}}$ there is exactly one circulation activity starting respectively ending in this event. This means we have to find a perfect matching, which is included by constraints (8) and (9). All three models are totally unimodular if the binary variables y_a and v_{ij} are given and fixed. We hence need not explicitly restrict x_i to be integer if all delays d_i and d_a are integer values.

Note that (DM) and (DM-fix) are similar: We simply add the delay propagation constraints for $\mathcal{A}_{\text{circ}}^{\text{fix}}$, which are of the same form as those for $\mathcal{A}_{\text{train}}$. Hence, (DM-fix) can be solved in the same way as (DM). (DM-opt), on the other hand, has a different structure, since it includes the matching constraints and has big-M-constraints for the circulation activities. Hence, it is much more difficult to solve, which also becomes apparent in the numerical results in Section 5. However, from a practical point of view it is the preferable of the three models, since it allows to adapt the circulations in a realistic way to the current situation. Therefore, our aim is to solve the problem (DM-opt).

3 Analyzing the models

As already said, we are interested in solving (DM-opt). Unfortunately, this model is hard to solve. On the other hand, (DM) is the classic delay management problem for which many solution algorithms exist (see e.g. [12, 7]) and (DM-fix) can also be interpreted as a classic delay management problem, just with a larger set of fixed activities. We hence can compute the values of (DM) and (DM-fix). Our first result shows that these two values can be used as bounds on the objective function value z^{opt} of (DM-opt). Recall that delays d_a are only relevant for activities $a \in \mathcal{A}_{\text{train}}$. To simplify notation, we set $d_a := 0$ for all $a \in \mathcal{A}\backslash\mathcal{A}_{\text{train}}$.

▶ **Lemma 1.** *For the optimal objective values of the three problems the following holds:*

$$z \leq z^{\text{opt}} \leq z^{\text{fix}}.$$

Proof. Every feasible solution for (DM-fix) is also feasible for (DM-opt) with appropriately chosen v and every solution for (DM-opt) yields a feasible solution for (DM). ◀

As shown in [15], we can bound the maximal delay of the events in an optimal solution.

▶ **Lemma 2** ([15]). *For each of the models (DM), (DM-fix) and (DM-opt) there is an optimal solution (x, y) respectively (x, y, v) such that for all events $i \in \mathcal{E}$ the following holds:*

$$x_i - \pi_i \leq d_{\max}^{\mathcal{E}} + \max_{p \in P_i} \sum_{a \in p} d_a,$$

where $P_i := \{p\colon\ p \text{ is a directed path from an arbitrary node to } i\}$ and $d_{\max}^{\mathcal{E}} := \max_{i\in\mathcal{E}} d_i$. In the special case that $d_a = 0$ for all $a \in \mathcal{A}$ this simplifies to $x_i - \pi_i \leq d_{\max}^{\mathcal{E}}$.

As seen in the previous lemma we can use the solution of (DM) as a lower bound while every solution to (DM-fix) is feasible for (DM-opt). Hence, solving (DM-fix) can be seen as a heuristic for solving (DM-opt). In the following we discuss how good this approach is. We first bound the value of (DM-fix).

▶ **Lemma 3.** *For the optimal objective value z^{fix} of (DM-fix) the following holds:*

$$z^{\text{fix}} \leq \sum_{i \in \mathcal{E}} w_i \cdot (d_{\max}^{\mathcal{E}} + \max_{p \in P_i} \sum_{a \in p} d_a) =: B.$$

For the special case $d_a = 0$ for all $a \in \mathcal{A}$ we have that $z^{\text{fix}} \leq P \cdot d_{\max}^{\mathcal{E}}$, where $P := \sum_{i \in \mathcal{E}} w_i$.

Proof. Let $x_i := \pi_i + d_{\max}^{\mathcal{E}} + \max_{p\in P_i} \sum_{a\in p} d_a$ for all $i \in \mathcal{E}$ and $y_a := 0$ for all $a \in \mathcal{A}_{\text{transfer}}$. Then (x, y) is a feasible solution for (DM-fix):

- For all $i \in \mathcal{E}$ we have

$$x_i = \pi_i + d_{\max}^{\mathcal{E}} + \max_{p \in P_i} \sum_{a \in p} d_a \geq \pi_i + d_i.$$

- For all $a = (i,j) \in \mathcal{A}_{\text{train}} \cup \mathcal{A}_{\text{transfer}} \cup \mathcal{A}_{\text{circ}}^{\text{fix}}$ we have

$$x_j - x_i = (\pi_j + d_{\max}^{\mathcal{E}} + \max_{p \in P_j} \sum_{a' \in p} d_{a'}) - (\pi_i + d_{\max}^{\mathcal{E}} + \max_{p \in P_i} \sum_{a' \in p} d_{a'})$$
$$\overset{(*)}{\geq} \pi_j - \pi_i + d_a \geq L_a + d_a,$$

where $(*)$ follows from the fact that every path $p \in P_i$ can be extended to a path $p' \in P_j$ with $\sum_{a' \in p} d_{a'} + d_a = \sum_{a' \in p'} d_{a'}$.

Therefore, the corresponding objective value

$$\sum_{i \in \mathcal{E}} w_i(x_i - \pi_i) + T \sum_{a \in \mathcal{A}_{\text{transfer}}} w_a y_a = \sum_{i \in \mathcal{E}} w_i \cdot (d_{\max}^{\mathcal{E}} + \max_{p \in P_i} \sum_{a \in p} d_a)$$

is an upper bound for z^{fix}. ◀

Using this lemma, we can restrict the approximation ratio of the heuristic solution (DM-fix), namely $z^{\text{fix}} \leq B \cdot z^{\text{opt}}$, if $z \geq 1$ (and hence $z^{\text{opt}} \geq 1$).

An idea for another bound is to ignore the delay of passengers exiting at events i which cannot be influenced by decisions on the circulations. From now on, let $\mathcal{E}_j^+$ denote the set of events that are reachable from $j \in \mathcal{E}$, i.e., $i \in \mathcal{E}_j^+ \subset \mathcal{E}$ iff there is a directed path from j to i in $(\mathcal{E}, \mathcal{A})$ and $\mathcal{A}_{\text{circ}}^i := \{(i', j) \in \mathcal{A}_{\text{circ}} \colon i \in \mathcal{E}_j^+\}$ the set of circulation activities from which i can be reached (see Figure 2).

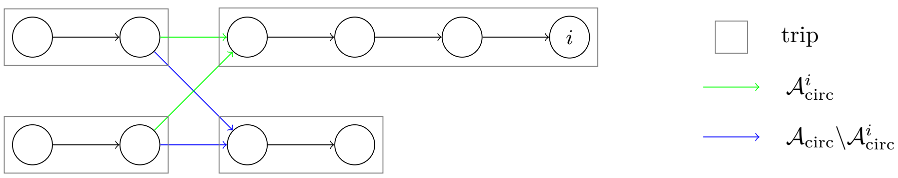


Figure 2 EAN showing notation $\mathcal{A}_{\text{circ}}^i$.

This allows for another bound between z^{fix} and z.

▶ **Proposition 4.** *For the optimal objective values of (DM-fix) and (DM) we have*

$$z^{\text{fix}} \leq z + \underbrace{\sum_{i \in \mathcal{E} : \mathcal{A}_{\text{circ}}^i \neq \emptyset} w_i (d_{\max}^{\mathcal{E}} + \max_{p \in P_i} \sum_{a \in p} d_a)}_{=:\tilde{B}} =: B'.$$

For the special case $d_a = 0$ for all $a \in \mathcal{A}$ we have $z^{\text{fix}} \leq z + d_{\max}^{\mathcal{E}} \sum_{i \in \mathcal{E} \colon \mathcal{A}_{\text{circ}}^i \neq \emptyset} w_i$.

Proof. We consider an optimal solution $(\bar{x}, \bar{y})$ of (DM) as in Lemma 2 and construct a feasible solution (x, y) of (DM-fix) as follows: We set $y := \bar{y}$ and

$$x_i := \begin{cases} \bar{x}_i, & \text{if } \mathcal{A}_{\text{circ}}^i = \emptyset, \\ \pi_i + d_{\max}^{\mathcal{E}} + \max_{p \in P_i} \sum_{a \in p} d_a, & \text{otherwise.} \end{cases}$$

Note that for $(i,j) \in \mathcal{A}$, $\mathcal{A}_{\text{circ}}^i \neq \emptyset$ implies $\mathcal{A}_{\text{circ}}^j \neq \emptyset$, since every path to i can be extended to j. Furthermore, by the definition of x and Lemma 2, for all $i \in \mathcal{E}$ we have $\bar{x}_i \leq x_i \leq \pi_i + d_{\max}^{\mathcal{E}} + \max_{p \in P_i} \sum_{a \in p} d_a$. Then (x, y) is indeed a feasible solution of (DM-fix):

- For all $i \in \mathcal{E}$ we have $x_i \geq \bar{x}_i \geq \pi_i + d_i$.

- For $a = (i, j) \in \mathcal{A}_{\text{transfer}}$ we have:

$$M_1 y_a + x_j - x_i = M_1 \bar{y}_a + x_j - \bar{x}_i \geq M_1 \bar{y}_a + \bar{x}_j - \bar{x}_i \geq L_a + d_a, \text{ if } \mathcal{A}^i_{\text{circ}} = \emptyset$$

and

$$\begin{aligned} M_1 y_a + x_j - x_i =& M_1 \bar{y}_a + (\pi_j + d^{\mathcal{E}}_{\max} + \max_{p \in P_j} \sum_{a' \in p} d_{a'}) - x_i \\ \geq& M_1 \bar{y}_a + (\pi_j + d^{\mathcal{E}}_{\max} + \max_{p \in P_j} \sum_{a' \in p} d_{a'}) - (\pi_i + d^{\mathcal{E}}_{\max} + \max_{p \in P_i} \sum_{a' \in p} d_{a'}) \\ \geq& M_1 \bar{y}_a + \pi_j - \pi_i + d_a \\ \geq& L_a + d_a, \text{ otherwise.} \end{aligned}$$

- Analogously, we can show that $x_j - x_i \geq L_a + d_a$ for $a = (i, j) \in \mathcal{A}_{\text{train}} \cup \mathcal{A}^{\text{fix}}_{\text{circ}}$. For the detailed proof we refer to [9].

Thus, all constraints of (DM-fix) are fulfilled and for the optimal objective value it follows that

$$\begin{aligned} z^{\text{fix}} &\leq \sum_{i \in \mathcal{E}} w_i (x_i - \pi_i) + T \sum_{a \in \mathcal{A}_{\text{transfer}}} w_a y_a \\ &= \sum_{i \in \mathcal{E} : \mathcal{A}^i_{\text{circ}} = \emptyset} w_i (\bar{x}_i - \pi_i) + \sum_{i \in \mathcal{E} : \mathcal{A}^i_{\text{circ}} \neq \emptyset} w_i (d^{\mathcal{E}}_{\max} + \max_{p \in P_i} \sum_{a \in p} d_a) + T \sum_{a \in \mathcal{A}_{\text{transfer}}} w_a \bar{y}_a \\ &\leq z + \sum_{i \in \mathcal{E} : \mathcal{A}^i_{\text{circ}} \neq \emptyset} w_i (d^{\mathcal{E}}_{\max} + \max_{p \in P_i} \sum_{a \in p} d_a). \end{aligned}$$ ◀

We obtain that $z^{\text{fix}} - z^{\text{opt}} \leq \tilde{B}$ and can use B' for another estimate on the approximation ratio analogously to B: If $z \geq 1$ we receive that $z^{\text{fix}} \leq B' \cdot z^{\text{opt}}$. Note that B and B' have no general order.

So far, we used that $z^{\text{opt}} \leq z^{\text{fix}}$ to derive bounds on (DM-opt). The following lemma presents a bound on (DM-opt) which uses z: If (DM) finds a solution without any delay for the passengers also (DM-opt) and (DM-fix) have solutions without any passengers' delay.

▶ **Proposition 5.** *Let $w_i > 0$ for all $i \in \mathcal{E}_{\text{end}}$ and $z = 0$. Then also $z^{\text{fix}} = z^{\text{opt}} = 0$.*

Proof. Let (x, y) be an optimal solution of (DM). We show that it is also feasible for (DM-fix). Since $z = 0$, it holds $x_i = \pi_i$ for all i with $w_i > 0$. In particular, this is true for all $i \in \mathcal{E}_{\text{end}}$. Hence, for $a = (i, j) \in \mathcal{A}_{\text{circ}}$ it follows that

$$x_j - x_i = x_j - \pi_i \overset{(*)}{\geq} \pi_j + d_j - \pi_i \overset{(**)}{\geq} L_a + d_j \geq L_a,$$

where $(*)$ follows from constraints (1) and $(**)$ from π being a feasible timetable. Thus, x fulfils the constraints (6). All other constraints of (DM-fix) are naturally fulfilled, since (x, y) is feasible for (DM). It follows that (x, y) is a feasible solution for (DM-fix) and hence, $z^{\text{fix}} = 0$. By Lemma 1, this also implies $z^{\text{opt}} = 0$. ◀

4 Algorithmic approaches

For large instances, it is not possible to solve the IP formulation in reasonable time (see Section 5). Thus, we consider three different heuristics. The first and the second ones look for local improvements when changing the circulation activities while the third one iteratively solves the delay management problem and optimizes the circulations.

4.1 NEI: Next-Event-Improve

A first heuristic approach is to compute a solution for (DM-fix), look for local improvements of the matching problem, and solve (DM-fix) again with the newly chosen circulations. We continue doing so until the solution does not improve any more.
Choosing the circulation activities means solving a perfect matching in the graph $\mathcal{N}[\mathcal{A}_{\text{circ}}]$: We have to match every event from $\mathcal{E}_{\text{end}}$ to an event from $\mathcal{E}_{\text{start}}$ by some circulation activity. For evaluating the quality of such a matching, for every trip we look at the delay which is propagated to the next event right after the start of the trip. Hence, for $(i,j) \in \mathcal{A}_{\text{circ}}$ we consider the time x_k at event $k \in \mathcal{E}$, where $(j,k) \in \mathcal{A}_{\text{drive}}$ is the first activity of the trip starting at j. By (1) it has to hold $x_k \geq \pi_k + d_k$. Furthermore, (2) implies $x_k \geq x_j + L_{(j,k)} + d_{(j,k)}$, so we need $x_k \geq \max(\pi_k + d_k, x_j + L_{(j,k)} + d_{(j,k)})$. Analogously, it holds $x_j \geq \max(\pi_j + d_j, x_i + L_{(i,j)})$ by (1) and (7) if $v_{ij} = 1$. Assuming that possible transfers to j are not maintained, for the disposition time of event k used in the objective function of Algorithm 1 (see appendix) we obtain the approximation $\tilde{x}_k = \max(\pi_k + d_k, \max(\pi_j + d_j, x_i + L_{(i,j)}) + L_{(j,k)} + d_{(j,k)})$ if $v_{ij} = 1$, where x_i is the time of event i in the incumbent solution.
For fixed circulations given by v we denote the corresponding instance of (DM-fix) by (DM-fix)(v).

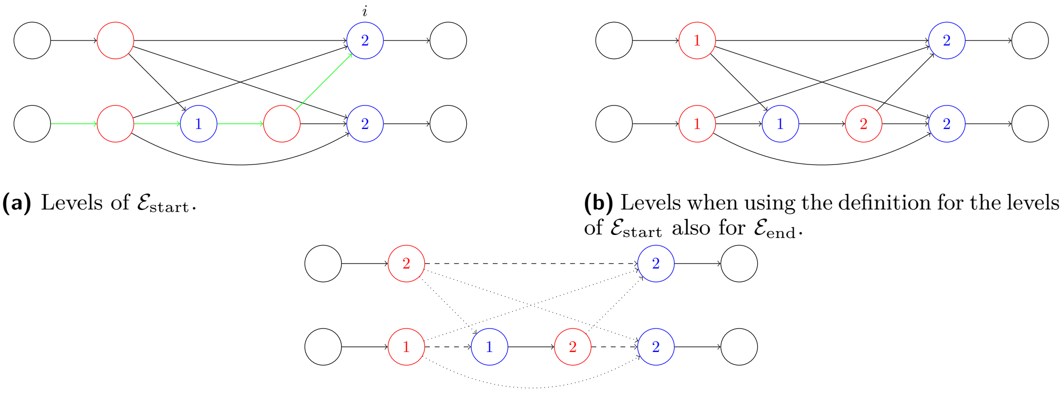
4.2 RE: Reachable Events

An idea for improving the running time of the algorithm is to not compute a matching for the whole EAN in every iteration, but to do so successively. The intuition behind this is that the choice of "later" circulation activities depends on the choice of "earlier" circulation activities. Hence, we fix the "early" circulation activities first and the "late" ones afterwards. Since an EAN is a time-expanded network, this can be expressed in terms of reachability. For $j \in \mathcal{E}_{\text{start}}$ let l be the maximal number of start events on a directed path in $\mathcal{N} = (\mathcal{E}, \mathcal{A})$ from an arbitrary node to j. We call l the *level* of j and denote it by $lv(j)$ and the maximal level is denoted by $l^{\max}$. An example with five trips is shown in Figure 3a. The red nodes are the end events $\mathcal{E}_{\text{end}}$, while the blue nodes are the start events $\mathcal{E}_{\text{start}}$. The green path ending at i contains two blue nodes and there is no path ending at i with more than two blue nodes. Hence, we have $lv(i) = 2$. We could define the levels of the end events $\mathcal{E}_{\text{end}}$ analogously, namely by counting the maximal number of end events on a path to $i \in \mathcal{E}_{\text{end}}$. This would lead to the levels shown in Figure 3b. However, in this case both levels contain an odd number of nodes, so we cannot find a perfect matching within the single levels (which is what we want to do in the heuristic). We can fix this problem by adapting the definition of the end events $\mathcal{E}_{\text{end}}$. For an end event $i \in \mathcal{E}_{\text{end}}$ we define $lv(i) := lv(j)$, where $j \in \mathcal{E}_{\text{start}}$ such that $(i,j) \in \mathcal{A}^{\text{fix}}_{\text{circ}}$. This way it is ensured that there always exists a perfect matching within the single levels, namely the one given by $\mathcal{A}^{\text{fix}}_{\text{circ}}$. As an example we have a look at Figure 3c. The dashed arcs represent the set $\mathcal{A}^{\text{fix}}_{\text{circ}}$ and the dotted arcs the set $\mathcal{A}_{\text{circ}} \backslash \mathcal{A}^{\text{fix}}_{\text{circ}}$.

A disadvantage of the objective function in Algorithm 1 is that it is "too local", i.e., the set of events we consider when fixing the matching is quite small. Thus, in our next approach we want to extend the objective function to not only take into account the delay at the next events, but at all reachable events. Furthermore, we want to fix the matching "level-wise" with the above definition of a level.

For some level l we denote by $\mathcal{E}^l := \{i \in \mathcal{E} \colon lv(i) = l\}$ the set of events on level l and $\mathcal{E}^l_{\text{start}} := \mathcal{E}^l \cap \mathcal{E}_{\text{start}}$ and $\mathcal{E}^l_{\text{end}} := \mathcal{E}^l \cap \mathcal{E}_{\text{end}}$. Furthermore, $\mathcal{A}^l_{\text{circ}} := \{(i,j) \in \mathcal{A}_{\text{circ}} : i,j \in \mathcal{E}^l\}$ is the set of circulation activities between vertices of the same level l.

Let $\mathcal{N}^l_{\text{circ}} = (\mathcal{E}^l_{\text{end}} \cup \mathcal{E}^l_{\text{start}}, \mathcal{A}^l_{\text{circ}})$ be the subgraph induced by the circulation activities on level l. Furthermore, we denote the start and end events on level l together with all nodes reachable from these by $\mathcal{E}^{l+}_{\text{all}} := \mathcal{E}^l_{\text{end}} \cup \mathcal{E}^l_{\text{start}} \cup \{i \in \mathcal{E} \colon i \text{ is reachable from } j \text{ for some } j \in \mathcal{E}^l_{\text{start}}\}$



(a) Levels of $\mathcal{E}_{\text{start}}$.

(b) Levels when using the definition for the levels of $\mathcal{E}_{\text{start}}$ also for $\mathcal{E}_{\text{end}}$.

(c) Levels according to our definition.

Figure 3 Example showing the definition of the levels.

and set $\mathcal{N}_{\text{all}}^{l+} := (\mathcal{E}_{\text{all}}^{l+}, \mathcal{A}_{\text{all}}^{l+}) := \mathcal{N}[\mathcal{E}_{\text{all}}^{l+}]$ the subgraph induced by $\mathcal{E}_{\text{all}}^{l+}$. Analogously to the subsets of $\mathcal{A}$, also for $\mathcal{A}_{\text{all}}^{l+}$ we define the subsets $\mathcal{A}_{\text{train}}^{l+} = \mathcal{A}_{\text{drive}}^{l+} \cup \mathcal{A}_{\text{wait}}^{l+}$, $\mathcal{A}_{\text{transfer}}^{l+}$ and $\mathcal{A}_{\text{circ}}^{l+}$. Now in every step of the heuristic we want to solve (DM-fix) for $\mathcal{N}_{\text{all}}^{l+}$ while simultaneously reoptimizing the circulation activities in $\mathcal{A}_{\text{circ}}^{l}$ and using the values x_i for $i \in \mathcal{E}_{\text{end}}^{l}$ we fixed in the iteration before. This corresponds to solving the IP (DM-opt-l-all), which can be found in the appendix, together with the resulting heuristic given in Algorithm 2.

4.3 DM-VS

Next we want to pursue a different approach, where we alternately solve (DM-fix) and optimize the vehicle circulations. Hence, we first introduce the vehicle scheduling problem. Usually, this problem is considered in the so-called trip graph, see e.g. [1]. However, to be consistent with our notation, we formulate it in the EAN. For an overview of different vehicle scheduling models we refer to [1]. A vehicle schedule is an assignment of vehicles to trips such that every trip is covered exactly once. This corresponds to an assignment fulfilling the constraints (8) to (10). The task of the vehicle scheduling problem (VS) is to find cost-minimal vehicle circulations, where the costs are most often given by a weighted sum of different cost shares. In our case, we are considering the fixed costs for using a vehicle, the covered distance and the driving time of the vehicles. Recall that in the definition of $\mathcal{E}_{\text{start}}$ and $\mathcal{E}_{\text{end}}$ we omitted the first departure and the last arrival of every circulation. In particular, if a vehicle depot is considered, the trips of the vehicles from the depot to the first trip of a circulation and from the last trip of a circulation back to the depot, and therefore also the number of necessary vehicles, are not changed. We can therefore omit the fixed costs for using a vehicle. Hence, we obtain the following formulation for a given disposition timetable x:

$$\min \sum_{(i,j)\in\tilde{\mathcal{A}}_{\text{circ}}} v_{ij} \cdot l(i,j) \tag{VS(x)}$$

$$\sum_{i\in\mathcal{E}_{end}:(i,j)\in\tilde{\mathcal{A}}_{\text{circ}}} v_{ij} = 1 \qquad j \in \mathcal{E}_{start} \tag{8}$$

$$\sum_{j\in\mathcal{E}_{start}:(i,j)\in\tilde{\mathcal{A}}_{\text{circ}}} v_{ij} = 1 \qquad i \in \mathcal{E}_{end} \tag{9}$$

$$v_{ij} \in \{0,1\} \qquad (i,j) \in \mathcal{A}_{\text{circ}}, \tag{10}$$

where $l(i,j)$ is the length (in kilometers) of a shortest path from the station of event i to the station of event j. Note that the set $\tilde{\mathcal{A}}_{\text{circ}} = \{(i,j) \in \mathcal{A}_{\text{circ}} \colon x_j - x_i \geq L_{(i,j)}\}$ of available circulation activities depends on the given timetable x and since the timetable is fixed we may omit the driving time of the vehicles from the objective function.

The idea of the new heuristic is the following: We first solve our instance of (DM-fix) and obtain a disposition timetable x. Next, we solve (VS)(x), i.e. we compute optimal vehicle circulations based on the times from the disposition timetable. This gives us a set of circulation activities with incidence vector v^{new}, which we then use to solve the delay management problem again, i.e., we solve (DM-fix)(v^{new}). We iterate until there is no improvement to the objective value of (DM-fix). The heuristic is given in Algorithm 3 (see appendix).

5 Computational results

In this section, we evaluate the results from the previous sections computationally using close-to real-world data. We compare the results for the three models (DM), (DM-fix) and (DM-opt) and analyze the performance of the heuristics developed in Section 4.

For all experiments we use the open-source software framework LinTim, see [17, 18]. We tested various close-to real-world datasets including representations of the metro system of `Athens`, the bus system of the German city `Göttingen` and several datasets depicting parts of the German high-speed railway network. An overview of the used datasets is given in Table 3 in the appendix. We use a given timetable repeated periodically every hour with a given vehicle schedule which covers 24 hours. For the three delay management models, we use the four hour time interval from 8:00 am to 12:00 pm. Information about the resulting EAN is given in Table 4 in the appendix. We use a LinTim procedure to generate uniformly distributed source delays. An interval for the size of the delays as well as the number of delays (given as percentage of the number of events and driving activities) are given as parameters. We consider several settings, which are given in Table 5 (see appendix). We implemented the IP models in Python and ran them on an Acer laptop with Intel(R) Core(TM) i5-7200U CPU @2.5 GHz and 8 GB RAM using the solver Gurobi 9.0.1 ([10]). In order to provide exact results, the instances which could not be solved within one hour were additionally run on a compute server with 12 cores of Intel(R) Xeon(R) X5680 @3.3 GHz and 128 GB RAM.

5.1 Comparison of (DM), (DM-fix) and (DM-opt)

Objective Values. We start by comparing the optimal objective values of the three models (DM), (DM-fix) and (DM-opt). Recall Lemma 1: $z \leq z^{\text{opt}} \leq z^{\text{fix}}$. Figure 4 shows the gap between z^{fix} and z^{opt} and between z^{fix} and z for different percentages of events/activities with a source delay from Table 5. For all datasets we observe that z^{opt} and z are very close. The only exception is the setting with 5% of source delays for `Göttingen`: here we have $z \approx 0.73 z^{\text{fix}}$ and $z^{\text{opt}} \approx 0.86 z^{\text{fix}}$, i.e., z is almost 15% smaller than z^{opt}. The deviation of z^{fix} and z^{opt} is much bigger, but depends a lot on the used dataset. For `Athens` the objective value is improved by almost 39% in the setting of 3% source delay. Interestingly, for `Germany` there is hardly any difference between the three values, probably due to the decreased number of possible circulations in a (relatively) sparse railway network. We conclude that the quality of the lower bound given by z is rather good, while (DM-opt) improves the solution significantly compared to (DM-fix).

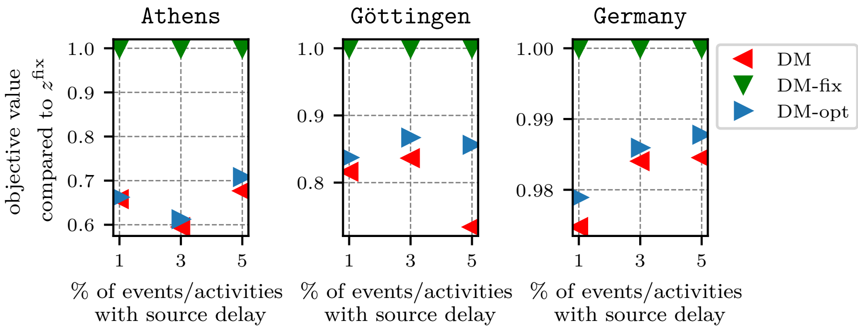


Figure 4 Comparison of objective values in settings 1 to 3.

Table 1 Average computation times (in seconds) for different percentages of events/activities with a source delay. Instances which could not be solved within one hour are marked as "limit".

	Athens			Göttingen			Germany		
	1%	3%	5%	1%	3%	5%	1%	3%	5%
(DM)	0.26	0.93	0.34	8.52	19.10	510.09	2.65	3.53	3.81
(DM-fix)	0.30	0.77	0.45	12.96	153.78	1160.62	2.62	3.74	3.79
(DM-opt)	16.93	138.65	limit	470.17	limit	limit	6.29	8.85	9.08

Computation times for IP-formulations. Next, we investigate the computation times for solving the integer programs, see Table 1. The time for reading the necessary data, which is never more than a few seconds, is not included in these numbers. First, we consider the results for `Athens`. Both (DM) and (DM-fix) could be solved within a second. While (DM-opt) for 1% delays was still quite fast with about 17 seconds, the computation time increased rapidly with an increasing number of delays. With 3% delays more than two minutes were needed and the instance with 5% delays could not be solved within the time limit of one hour. However, the optimality gap at the end of the time limit was only 0.58%. For `Göttingen` the situation is worse. Here, already (DM) needed nearly 9 minutes in the case of 5% delays and for (DM-fix) it were almost 20 minutes. For solving (DM-opt) with 1% delays about 8 minutes were needed, while neither for 3% nor for 5% the IP could be solved within the time limit. A gap of 2.42% respectively 16.59% was left.
The results for `Germany` paint a completely different picture. All instances could be solved in at most 9 seconds. This is surprising since the infrastructure network as well as the sets $\mathcal{E}$ and $\mathcal{A}_{\text{pure}}$ for `Germany` are larger than for both of the other instances, see Tables 3 and 4. However, the set $\mathcal{A}_{\text{circ}}$ of all possible circulation activities is much smaller, which makes finding the matching when solving (DM-opt) an easier task.

5.2 Heuristics

As established in the previous section, while (DM-opt) for `Germany` can be solved in seconds, this is not the case for `Athens` and `Göttingen`. Hence, in this section we evaluate the performance of the different heuristics from Section 4 for both of these datasets. Recall that in the level-wise heuristic RE only those arcs $(i, j) \in \mathcal{A}_{\text{circ}}$ with $lv(i) = lv(j)$ are considered. For `Athens` this set contains 2706 arcs, for `Göttingen` it is 6200, which is in both cases significantly smaller than the original set $\mathcal{A}_{\text{circ}}$.

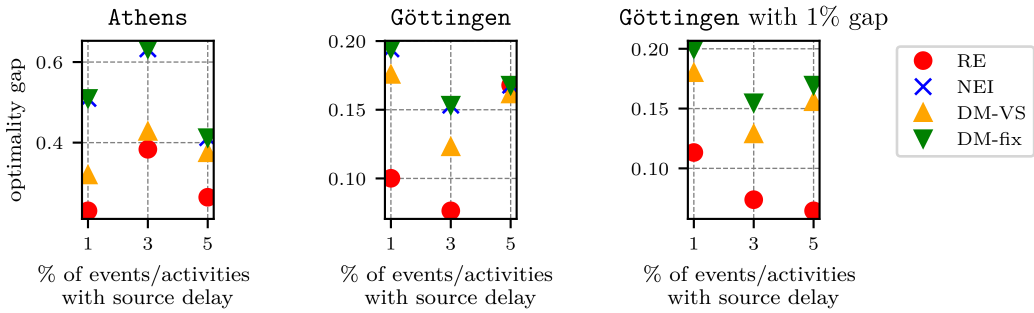


Figure 5 Quality of heuristics in settings 1 to 3: `Athens`, `Göttingen` and `Göttingen` with 1% gap.

Table 2 Average computation times (in seconds) of heuristics for different percentages of events/activities with source delays. Instances which could not be solved within one hour are marked as "limit".

	Athens			Göttingen			Göttingen with 1% gap		
	1%	3%	5%	1%	3%	5%	1%	3%	5%
RE	42.32	44.87	49.56	138.68	2831.62	limit	54.26	110.52	271.02
DM-VS	24.15	25.52	47.07	79.70	3166.24	limit	22.75	46.26	95.91
(DM-fix)	0.30	0.77	0.45	12.96	153.78	1160.62	2.63	2.96	10.48

Solution Quality. We start by assessing the quality of the solutions, i.e., we compare the objective values to z^{opt}, see Figure 5. If the time limit of one hour was reached, the best found solution is given. Note that all algorithms start by solving (DM-fix), so their computation time will be larger than simply solving (DM-fix). Hence, it only makes sense to use these heuristics if they provide solutions with an objective value smaller than z^{fix}. While Algorithm NEI fails to produce any solutions with objective value better than z^{fix}, the others yield significantly better results and were able to improve the solution given by (DM-fix) in almost all cases. On the `Athens` data Algorithm RE is always better than DM-VS. We note that for `Athens`, although significantly better than (DM-fix), the solution quality is still quite poor for all algorithms: the smallest gap compared to z^{opt} we obtained is 23%. The results are much more promising for `Göttingen`, where we were able to obtain a gap of less than 8% for the setting with 3% delays. The case of 5% of source delays was the only one in which RE could not improve the solution of (DM-fix). DM-VS only yields a slight improvement.

Computation Times. For comparing the computation times of the heuristics we omitted NEI, since it is inferior to (DM-fix). As can be seen in Table 2, both heuristics take much longer than solving (DM-fix). However, for the `Athens` dataset they still run in reasonable time: all instances could be solved within one minute, where RE is always a bit slower than DM-VS. For the `Göttingen` data the computation times are even longer. While the instance with 1% delays could be solved within a few minutes, for 3% delays the computation times increased significantly. Even the faster of the heuristics took about 47 minutes, which is not acceptable. With 5% delays the time limit was reached for both algorithms. A possibility to mitigate this problem is to allow a small optimality gap when solving the integer programs used in the algorithms. With such a gap of 1% we repeated the experiments for `Göttingen`. As Table 2 shows, the effect is enormous: in all cases the computation times reduced to less than 5 minutes. As can be seen in Figure 5, in the first two settings the solution quality is as good as in the previous experiments. For the setting with 5% source delays we get a significant improvement: RE now finishes within the time limit and yields good results with about 6% optimality gap.

6 Outlook

In this paper we showed the potential of including decisions on vehicle circulations in delay management and we developed and analyzed three heuristics for the integrated problem, two of them providing very good solutions in our experiments. We have also seen that the *price of sequentiality* (see [21, 19]), i.e., the ratio between the optimal objective without integrating the vehicle circulation decisions and the optimal objective value for the integrated problem, can be bounded theoretically, but the bound can become very high. This coincides with our experiments in which we show that integrating vehicle circulation decisions in delay management may reduce the delay for the passengers significantly.

There are several aspects for ongoing research. First, a further speed-up of the heuristics is relevant. Second, some aspects of delay management were not considered here. This includes headway constraints between vehicles as well as realistic passenger behavior, e.g., in the form of rerouting. The latter may be included along the lines of [20]. Finally, it would be best to avoid delays as much as possible. This can be done by making the timetable more robust. Many research papers are devoted to the topic of robust timetabling, see, e.g., [13] and references therein. In the context of our paper, a timetable is *robust* if for any delay scenario there exists a solution to the delay management problem with acceptable passengers' delays. This is a further interesting topic for future research.

References

1 Stefan Bunte and Natalia Kliewer. An overview on vehicle scheduling models. *Public Transport*, 1:299–317, 2009.

2 Twan Dollevoet, Francesco Corman, Andrea D'Ariano, and Dennis Huisman. An iterative optimization framework for delay management and train scheduling. *Flexible Services and Manufacturing Journal*, 26:490–515, 2012.

3 Twan Dollevoet and Dennis Huisman. Fast heuristics for delay management with passenger rerouting. *Public Transport*, 6:67–84, 2011.

4 Twan Dollevoet, Dennis Huisman, Leo Kroon, Marie Schmidt, and Anita Schöbel. Delay management including capacities of stations. *Transportation Science*, 49(2):185–203, 2015.

5 Twan Dollevoet, Dennis Huisman, Leo Kroon, Lucas Veelenturf, and Joris Wagenaar. Application of an iterative framework for real-time railway rescheduling. *Computers & Operations Research*, 78:203–217, 2017.

6 Twan Dollevoet, Dennis Huisman, Marie Schmidt, and Anita Schöbel. Delay management with rerouting of passengers. *Transportation Science*, 46(1):74–89, 2012.

7 Twan Dollevoet, Dennis Huisman, Marie Schmidt, and Anita Schöbel. *Delay Propagation and Delay Management in Transportation Networks*, pages 285–317. Springer, 2018.

8 Holger Flier, Marc Nunkesser, Michael Schachtebeck, and Anita Schöbel. Integrating rolling stock circulation into the delay management problem. NAM preprint series, Georg-August-Universität Göttingen, 2007.

9 Vera Grafe. Delay management with integrated vehicle scheduling: Analysis and algorithms. Master's thesis, Technische Universität Kaiserslautern, 2020.

10 Gurobi Optimization, LLC. Gurobi optimizer reference manual, 2020. URL: `http://www.gurobi.com`.

11 Rowan Hoogervorst, Twan Dollevoet, Gábor Maróti, and Dennis Huisman. Reducing passenger delays by rolling stock rescheduling. *Transportation Science*, 54(3):762–784, 2020.

12 Eva König. A review on railway delay management. *Public Transport*, 12:335–361, 2020.

13 Richard M. Lusby, Jesper Larsen, and Simon Bull. A survey on robustness in railway planning. *European Journal of Operational Research*, 266:1–15, 2018.

14 Ralf Rückert, Martin Lemnian, Christoph Blendinger, Steffen Rechner, and Matthias Müller-Hannemann. Panda: A software tool for improved train dispatching with focus on passenger flows. *Public Transport*, 9:307–324, 2017.

15 Michael Schachtebeck. *Delay Management in Public Transportation: Capacities, Robustness, and Integration*. PhD thesis, Georg-August-Universität Göttingen, 2009.

16 Michael Schachtebeck and Anita Schöbel. To wait or not to wait and who goes first? delay management with priority decisions. *Transportation Science*, 44:307–321, 2010.

17 Alexander Schiewe, Sebastian Albert, Vera Grafe, Philine Schiewe, Alexander Schöbel, and Felix Spühler. LinTim - Integrated Optimization in Public Transportation. Homepage. https://www.lintim.net/. open source.

18 Alexander Schiewe, Sebastian Albert, Vera Grafe, Philine Schiewe, Anita Schöbel, and Felix Spühler. LinTim: An integrated environment for mathematical public transport optimization. Documentation for version 2021.12. Technical report, Fraunhofer-Institut für Techno- und Wirtschaftsmathematik, 2021. URL: `https://nbn-resolving.org/urn:nbn:de:hbz:386-kluedo-66870`.

19 Philine Schiewe. *Integrated Optimization in Public Transport Planning*, volume 160 of *Optimization and Its Applications*. Springer, 2020. `doi:10.1007/978-3-030-46270-3`.

20 Philine Schiewe and Anita Schöbel. Periodic timetabling with integrated routing: Towards applicable approaches. *Transportation Science*, 54(6):1714–1731, 2020.

21 Philine Schiewe and Anita Schöbel. Integrated optimization of sequential processes: General analysis and application to public transport. Technical report, TU Kaiserslautern, 2021. submitted for publication.

22 Anita Schöbel. A model for the delay management problem based on mixed-integer-programming. *Electronic Notes in Theoretical Computer Science*, 50(1):1–10, 2001.

23 Anita Schöbel. Integer programming approaches for solving the delay management problem. In *Algorithmic Methods for Railway Optimization*, volume 4359 of *Lecture Notes in Computer Science*, pages 145–170. Springer, 2007.

A Algorithms

Algorithm 1 NEXT-EVENTS-IMPROVE (NEI).

Input : EAN $\mathcal{N} = (\mathcal{E}, \mathcal{A})$, incidence vector v^{old} of $\mathcal{A}^{\text{fix}}_{\text{circ}}$
Output : A feasible solution (x, y, v) of (DM-opt)

Solve (DM-fix)(v^{old}). Let (x, y) be an optimal solution and $\tilde{z}$ the optimal objective value.
while *true* **do**
  Compute a perfect matching in $\mathcal{N}_{\text{circ}} := \mathcal{N}[\mathcal{A}_{\text{circ}}]$ with incidence vector v^{new} such that

$$\sum_{(i,j)\in\mathcal{A}_{\text{circ}}} v^{\text{new}}_{ij} \sum_{k\in\mathcal{E}:(j,k)\in\mathcal{A}} w_k(\tilde{x}_k - \pi_k)$$

  is minimal, where $\tilde{x}_k = \max(\pi_k + d_k, \max(\pi_j + d_j, x_i + L_{(i,j)}) + L_{(j,k)} + d_{(j,k)})$.
  if $v^{\text{old}} \neq v^{\text{new}}$ **then**
    Solve (DM-fix)(v^{new}). Let $(\bar{x}, \bar{y})$ be an optimal solution and $\bar{z}$ the optimal objective value.
    if $\bar{z} < \tilde{z}$ **then**
      $v^{\text{old}} = v^{\text{new}}, x = \bar{x}, y = \bar{y}, \tilde{z} = \bar{z}$
    else
      return (x, y, v^{old})
    end
  else
    return (x, y, v^{old})
  end
end

$$\min \sum_{i\in\mathcal{E}^{l+}_{\text{all}}} w_i(x^l_i - \pi_i) + T \sum_{a\in\mathcal{A}^{l+}_{\text{transfer}}} w_a y_a \tag{DM-opt-l-all}$$

$$x^l_i = x^{l-1}_i \qquad i \in \mathcal{E}^l_{\text{end}} \tag{11}$$

$$x^l_i \geq \pi_i + d_i \qquad i \in \mathcal{E}^{l+}_{\text{all}} \backslash \mathcal{E}^l_{\text{end}} \tag{12}$$

$$x^l_j - x^l_i \geq L_a + d_a \qquad a = (i,j) \in \mathcal{A}^{l+}_{\text{train}} \tag{13}$$

$$My_a + x^l_j - x^l_i \geq L_a \qquad a = (i,j) \in \mathcal{A}^{l+}_{\text{transfer}} \tag{14}$$

$$x^l_j - x^l_i \geq L_a \qquad a = (i,j) \in (\mathcal{A}^{l+}_{\text{circ}} \backslash \mathcal{A}^l_{\text{circ}}) \cap \mathcal{A}^{\text{fix}}_{\text{circ}} \tag{15}$$

$$M(1 - v_{ij}) + x^l_j - x^l_i \geq L_a \qquad a = (i,j) \in \mathcal{A}^l_{\text{circ}} \tag{16}$$

$$\sum_{i\in\mathcal{E}^l_{\text{end}}:(i,j)\in\mathcal{A}_{\text{circ}}} v_{ij} = 1 \qquad j \in \mathcal{E}^l_{\text{start}} \tag{17}$$

$$\sum_{j\in\mathcal{E}^l_{\text{start}}:(i,j)\in\mathcal{A}_{\text{circ}}} v_{ij} = 1 \qquad i \in \mathcal{E}^l_{\text{end}} \tag{18}$$

$$x^l_i \in \mathbb{N} \qquad i \in \mathcal{E}^{l+}_{\text{all}} \tag{19}$$

$$y_a \in \{0,1\} \qquad a \in \mathcal{A}^{l+}_{\text{transfer}} \tag{20}$$

$$v_{ij} \in \{0,1\} \qquad (i,j) \in \mathcal{A}^l_{\text{circ}} \tag{21}$$

Algorithm 2 Reachable-Events (RE).

Input : EAN $\mathcal{N} = (\mathcal{E}, \mathcal{A})$, incidence vector v^{old} of $\mathcal{A}^{\text{fix}}_{\text{circ}}$
Output : A feasible solution (x, y, v) of (DM-opt)

Solve (DM-fix)(v^{old}). Let (x, y) be an optimal solution and $\tilde{z}$ the optimal objective value.
for l in $\{1, \dots, l^{\max}\}$ **do**
  Solve (DM-opt-l-all). Let (x^l, y^l, v^l) be an optimal solution and v^{new} the incidence vector of $\mathcal{A}_{\text{circ}}$ when replacing v^{old}_{ij} by v^l_{ij} for $(i, j) \in \mathcal{A}^l_{\text{circ}}$.
  if $v^{\text{old}} \neq v^{\text{new}}$ **then**
    Solve (DM-fix)(v^{new}). Let $(\bar{x}, \bar{y})$ be an optimal solution and $\bar{z}$ the optimal objective value.
    if $\bar{z} < \tilde{z}$ **then**
      $v^{\text{old}} = v^{\text{new}}, x = \bar{x}, y = \bar{y}, \tilde{z} = \bar{z}$
end
return (x, y, v^{old})

Algorithm 3 DM-VS.

Input : EAN $\mathcal{N} = (\mathcal{E}, \mathcal{A})$, incidence vector v^{old} of $\mathcal{A}^{\text{fix}}_{\text{circ}}$
Output : A feasible solution (x, y, v) of (DM-opt)

Solve (DM-fix)(v^{old}). Let (x, y) be an optimal solution and $\tilde{z}$ the optimal objective value.
while *true* **do**
  Solve (VS)(x). Let v^{new} be the incidence vector of the corresponding circulation activities.
  if $v^{\text{old}} \neq v^{\text{new}}$ **then**
    Solve (DM-fix)(v^{new}). Let $(\bar{x}, \bar{y})$ be an optimal solution and $\bar{z}$ the optimal objective value.
    if $\bar{z} < \tilde{z}$ **then**
      Set $v^{\text{old}} = v^{\text{new}}, x = \bar{x}, y = \bar{y}, \tilde{z} = \bar{z}$.
    else
      return (x, y, v^{old})
    end
  else
    return (x, y, v^{old})
  end
end

B Data

Table 3 Size of the PTN in the used datasets.

Dataset	Number of stops	Number of edges	Number of OD-pairs	Number of passengers
`Athens`	51	52	2385	63323
`Göttingen`	257	548	58226	406146
`Germany`	319	452	77878	4183088

Table 4 Size of the EAN in the used datasets.

Dataset	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{A}_{\text{pure}}\rvert$	$\lvert\mathcal{A}_{\text{circ}}\rvert$	$\lvert\mathcal{A}_{\text{circ}}^{\text{fix}}\rvert$
`Athens`	5551	7142	72535	387
`Göttingen`	17798	33318	81844	412
`Germany`	21466	46385	30428	666

Table 5 Parameters for the delay generation and the resulting sum of delays for the used datasets.

Setting	Interval for source delays (s)	% of events/ activities with source delay	Sum of source delays		
			`Athens`	`Göttingen`	`Germany`
1	$[60, 900]$	1%	42093	129015	144771
2	$[60, 900]$	3%	120616	389729	456749
3	$[60, 900]$	5%	202511	627346	752768