



Extending robustness concepts and model formulations for the timetable-constrained railway station routing problem using absorption graphs

Philipp Widmann^{1,2} · Norman Weik²

Accepted: 20 March 2026
© The Author(s) 2026

Abstract

Ensuring the feasibility and robustness of train operations in stations and network nodes is receiving more and more attention given the trend towards a stronger integration of timetabling and network planning in railway systems. Existing approaches for robust station routing most commonly rely on the maximization of buffer times between consecutive trains. This view, however, neglects indirect dependencies between trains which may exacerbate delay propagation in the station, as well as dwell-time supplements. In this work we develop a new exact robust routing approach for stations that allows to minimize the expected delay caused by primary delays, generalizing delay absorption concepts used in timetable stability analysis and recoverable robustness. The methodology is based on a mixed-integer linear programming (MILP) approach for modeling delay propagation that takes both direct and indirect train dependencies into account. We test our approach on three railway stations from the Netherlands, where it is found that the optimized routings of the new method can reduce the delay propagation compared to buffer-based methodologies.

Keywords Train routing · Railway stations · Robustness · Mixed-integer programming

✉ Philipp Widmann
philipp.widmann@dlr.de

Norman Weik
norman.weik@tum.de

¹ German Aerospace Center (DLR), Institute of Transportation Systems, Lilienthalplatz 7, 38108 Braunschweig, Germany

² Technical University of Munich, TUM School of Engineering and Design, Professorship for Design and Operation of Public Rail Transport Systems (RTS), Parking 35, 85748 Garching, Germany

1 Introduction

Rail traffic planning consists of multiple consecutive stages and subproblems representing different scopes and granularity of information. Depending on their time horizon, problems are classified into three levels: Strategic planning problems that include, e.g., network design, infrastructure dimensioning and line planning, which have the potential to shape operations years or even decades in advance; tactical problems that concern timetabling, routing, rolling stock and crew scheduling for the next (annual) timetable period; as well as operational problems that address short-term re-planning and real-time management of train operations in reaction to disruptions and delays (Sartor et al. 2023; Duarte et al. 2025).

Focusing on the tactical level associated with timetabling and routing in railway systems, the problem is typically too large to be solved in full detail on the whole network, and decomposition approaches are considered along, e.g., geographic, temporal or train-based dimensions (Leutwiler and Corman 2023). One line of work focuses on the decomposition into the macroscopic, network-wide generation of transport concepts and timetables as well as the detailed, microscopic planning (Bešinović et al. 2016), often carried out specifically for station nodes (Sels 2016; Lamorgese et al. 2017). Both scopes, featuring different data granularity, tools and algorithms are often approached in a sequential or iterative manner, where the network-wide timetable is generated first and potentially modified after the microscopic routing step depending on the feasibility in the nodes of the network (Lusby et al. 2011a).

Railway stations, where multiple different railway lines intersect and scheduled connections between different services are required, are particularly critical for creating a reliable, high-quality network timetable. To provide connectivity between different lines, ensure access to workshops and depot areas, as well as to maintain a certain flexibility for traffic management in perturbed situations, a more complex infrastructure allowing for different routing options upon entry or exit of the station is maintained. The problem of how to route trains in these complex station nodes is referred to as the train routing or train platforming problem (Lusby et al. 2011a; Caprara et al. 2011). Here, we consider a problem setting that includes both the platforming of the trains, as well as the choice of route through the station (out of potentially multiple options per platform).

The first objective for this task lies in ensuring the feasibility of the chosen routes for all trains, i.e., to find a routing plan that is both conflict-free as well as consistent with the arrival and departure times of the desired network timetable. Beyond this, an increasing focus has been put on optimizing additional quality metrics of the routing plan, e.g., its robustness against delays which are expected to occur during railway operations. Ideally, incorporating these considerations during the tactical planning process simplifies problems occurring in later planning phases.

Previous optimization approaches for robustness have mainly focused on different auxiliary metrics like the buffer time between train movements or heuristic approaches (cf. Section 2.2). More direct representations for the spread of delay have so far only been explored for evaluating delays (Delorme et al. 2009) or minimizing worst-case bounds (Caprara et al. 2014). An exact optimization approach focusing on the average behavior in the station context is missing. Thus, in this paper,

we approach the station routing and platforming problem with a focus on the direct minimization of expected delay propagation. The main contributions presented in the following are:

- We introduce the concept of absorption, which has previously been discussed for stability assessment in Delorme et al. (2009), directly into an optimization model for robust routing in railway station areas. The approach is based on a detailed, microscopic representation of the infrastructure and rail traffic control in station areas featuring blocking times and sectional route release.
- We formulate this as a mixed-integer linear program (MILP) that allows to directly minimize the expected impact of isolated primary delays on the delay of outbound trains leaving the station. The model encodes both direct and indirect train dependencies, as well as dwell-time supplements.
- Based on three railway stations in the Netherlands, we perform a comparative evaluation of the new absorption-based metric and a buffer time-based concept for robustness optimization from the literature. We observe slight robustness benefits through the application of the absorption-based view and highlight differences in the inner workings of the two frameworks.

The subsequent presentation is structured as follows: Section 2 discusses existing approaches and related work in robust station routing. Then, the delay propagation model and objective are developed in Section 3. Section 4 gives details on how to transfer these theoretical considerations into a MILP model for robust station routing, which is then applied to the three railway stations from the Netherlands in Section 5. Finally, Section 6 gives conclusions and an outlook on further research directions.

2 Related work

The station routing problem, being a central and critical part of railway timetabling and traffic planning, has received steady attention since the introduction of computer-based planning support in the 1990s (Zwaneveld et al. 1996; Cardillo and Mione 1998). While early works were often primarily concerned with finding a feasible solution, recent research has put an increasing focus on including different robustness considerations. Our discussion of related work is based on these two perspectives: We start by giving an overview of different modeling choices used to represent the general feasibility problem for stations in Section 2.1. Afterwards, we analyze metrics that have been proposed to measure the robustness of the local station routings and how they have been utilized to find improved solutions in Section 2.2. In addition to our subsequent discussion, an in-depth review of these topics is also conducted in the following surveys: Lusby et al. (2011a) give a detailed overview of different mathematical models used to ensure feasibility and Lusby et al. (2018) review robustness concepts both in station routing as well as in the broader context of railway planning. Bešinović (2020) as well as Ge et al. (2022) treat different notions of resilience and robustness in the area of railway and public transport planning. Finally,

the embedding of the station routing problem in the general landscape of operations research in railway planning is discussed by Duarte et al. (2025).

2.1 Train routing and platforming problem in stations

The input to the train routing problem in stations consists of a timetable proposed by a network-wide procedure that prescribes the arrival and departure times of trains in the station. The task is then to assign to each train a stopping platform as well as a route from its arrival to its departure point within the considered station infrastructure. In order to obtain a feasible solution, the paths assigned to trains need to be conflict-free, i.e., the occupation times of infrastructure elements (block segments) for different trains are not allowed to overlap. Various modeling choices to represent these system requirements, which are determined and enforced by the design of the signaling and interlocking systems, have been proposed and are discussed in the following.

One of the first and most prominent formulations by Zwaneveld et al. (1996) models these constraints as a node-packing problem in a conflict graph, with nodes representing the choice of a train path and edges indicating incompatibilities. To determine a feasible routing, a stable set in the graph needs to be found. To tighten the LP relaxation of the corresponding integer problem, the authors do not only rely on pairwise incompatibilities, but include cliques between a single train path of one train and all train paths of a second train that are incompatible with it. No cliques between more than two trains are considered due to the potentially exponential number of cliques and the difficulty of determining them.

Subsequent work by Kroon et al. (1997) established that the train routing problem is NP-complete if each train has three path alternatives and Zwaneveld et al. (2001) proposed to split the train path into inbound, platform and outbound sections to reduce combinatorial possibilities. The node-packing formulation forms the basis of multiple later works on robustness in station routing optimization (Caprara et al. 2011, 2014; Dewilde et al. 2014).

A related formulation by Caimi et al. (2011) splits the routes and assigns binary variables to track sections (sub-segments of the infrastructure on which the blocking times do not change), reusing common segments between paths. This exposes more problem structure and transforms the task of assigning train paths into a multicommodity-flow problem. Furthermore, the authors describe how to obtain clique constraints between all trains (per track section) to obtain a tighter LP-relaxation. Empirical performance gains are demonstrated for routing optimization using fixed platforms compared to the node-packing formulation.

Closely related, Caprara (2015) compares two general modeling approaches – termed path and arc-node formulations – appearing in different contexts in railway planning, which generally correspond to the formulations by Zwaneveld et al. (1996) and Caimi et al. (2011), respectively. For the combination with column generation and heuristic approaches, the author identifies path-based formulations as the more promising formulation.

Lusby et al. (2011b) also split the infrastructure into track sections, but keep a path-based representation of the binary choice variables and formulate incompatibili-

ties through the usage of time interval track sections, i.e., whether a path occupies a track section in a given interval. The authors show that this large set-packing model has a tighter LP relaxation than the node-packing formulation and propose a branch-and-price framework to solve the model efficiently.

Finally, other methods (e.g., Sels et al. 2014; Burggraeve and Vansteenwegen 2017) avoid the pre-computation of train conflicts, opting instead to include the occupation times of track segments directly as variables in the model and using them to formulate constraints based on blocking time overlaps. This allows a more fine-grained representation and adaption of blocking times and is thus also suited for the problem of real-time dispatching optimization, where similar formulations are used, e.g., in the RECIFE-model (Pellegrini et al. 2014, 2015).

2.2 Robustness in station routing

The methodologies described in the previous section exemplify different approaches for finding a conflict-free, feasible routing. In addition to this, research has been increasingly targeted towards identifying a routing plan that is not only feasible, but of high quality. In particular, routing plans have been studied for their capabilities of handling perturbations, which occur frequently during railway operations. However, in contrast to real-time railway optimization, no precise delay scenario is known at this stage of the planning process. Instead, the goal is to identify routing plans that can adapt well to a vast number of possible delay scenarios; closely related to classical concepts of robust optimization. Given the difficulty of the problem, different auxiliary metrics and heuristic approaches have been described.

The buffer time between two consecutive train paths, which describes the smallest time distance between their blocking times on common infrastructure segments, forms one such prominent metric. An increase of the buffer times, hence reducing the strength of delay transmission paths, or the avoidance of interdependencies altogether, are straightforwardly linked to increased robustness of the timetable. Different objectives utilizing this metric have been proposed: Caprara et al. (2010, 2011) associate each buffer time between train paths with a penalty (with smaller buffers incurring larger penalties) and minimize the sum over all train path pairs. Dewilde et al. (2013, 2014) also employ the sum of weighted buffer times as the auxiliary metric, with the inverse buffer time chosen as penalty. Building on their work, van Hoeck and Vansteenwegen (2024) apply a similar objective to the optimization of the routing in a larger bottleneck area containing multiple stations. Zeng et al. (2022) also optimize robustness based on inverse buffer times in a heuristic approach, using a basic delay propagation model for a posteriori validation. Finally, Burggraeve and Vansteenwegen (2017) follow a more unconventional process targeted at the largest, most critical nodes in the network which determine the network timetable instead of being constrained by it: The train routes are fixed first based on a balanced usage of the station infrastructure before the timetable is decided by maximizing a combined objective consisting of the unweighted sum of all buffers as well as the minimal buffer present in the routing. As a common trend, we note that these methods put particular emphasis on small buffers, as these should be violated more often and by a larger margin in practice.

Still, buffer times can not paint a full picture of the way delay spreads within a train routing plan. The effectiveness of a buffer not only depends on its size, but also on where the buffer is placed in the train routing plan. In particular, it depends on the local density of the timetable, i.e., on whether the secondary delay that could not be absorbed by the buffer may spread further to multiple follow-up trains. Delorme et al. (2009) extend the buffer concept to include such indirect dependencies. Using a shortest-path methodology in a buffer-time graph, they determine the smallest primary delay of one train that has no effect on a second train, through direct or indirect delay propagation (in the context of $(\max,+)$ -approaches for timetable stability analysis, this metric is referred to as recovery time (Goverde 2007)). The stability of a railway station timetable for a given primary delay level is the sum of these potential secondary delays, and the authors propose to evaluate multiple timetables at different delay levels to find those timetables that are efficient. The application of the concepts, however, remains bound to the quantification and analysis of robustness; the paper does not include any component to improve or optimize the stability.

Independently, Caprara et al. (2014) describe a methodology that tracks the spread of delay in the same fashion and includes it in a MILP model. However, the authors use it to find a routing that minimizes the worst-case delay across a set of scenarios with bounded primary delay. Their approach is an application of the recoverable robustness concept of Liebchen and Stiller (2009) to station routing, with recovery actions in case of primary delays limited to delaying subsequent trains.

Finally, the delay robustness of a routing may also be evaluated empirically: Bešinović and Goverde (2019) include a delay propagation simulation to evaluate the effect of multiple small deviations from the planned timetable and determine the average delay. This robustness evaluation forms one of the building blocks for their objective, which also takes stability (through capacity occupation) and a balanced infrastructure usage into account. The authors propose a rule-based heuristic procedure to improve the objective through suitable route permutations, but do not explore exact solution strategies.

Table 1 summarizes the presented literature on robustness optimization in the station context. It is clear that, while the reduction of delay propagation is seen as an important goal, there is still a lack of work towards integrating such comparatively more complex metrics directly into exact optimization methods. In the following, we, thus, present an approach to directly minimize the *expected* secondary delay in railway station timetables, which builds on and combines the strengths of different robustness concepts and metrics for railway station routing discussed in this section. In Section 3 we motivate and define a generalized metric for the *absorption capability of a timetable* as a concept for including indirect dependencies and delay transmission between trains when measuring the robustness of timetables, and discuss how it can be used to define fitting optimization objectives. Section 4 then builds a MILP representation for modeling and optimizing the robustness of the station routing plan using this quantity.

Table 1 Summary of literature on robust station routing. Metrics considered only for evaluation of solutions are in *italics*.

Publication	Area	Robustness metric(s)	Method
Delorme et al. (2009)	Junction	<i>Delay propagation (eval.)</i>	(Only evaluation)
Caprara et al. (2010, 2011)	Station	Buffers	MILP
Dewilde et al. (2013, 2014)	Zone of stations	Buffers <i>Travel time extension (eval.)</i> <i>Delay propagation (eval.)</i>	Part MILP, part heuristic
Caprara et al. (2014)	Station	Delay propagation in the worst case	MILP
Burggraef and Vansteenwegen (2017)	Zone of stations	Balanced infrastructure usage Buffers	Sequential MILPs
Bešinović and Goverde (2019)	Station	Simulated delay propagation Capacity occupation Balanced infrastructure usage	Heuristic
Zeng et al. (2022)	Station	Buffers <i>Delay propagation (eval.)</i>	Heuristic
van Hoeck and Vansteenwegen (2024)	Partial network	Buffers	MILP
This paper	Station	Expected delay propagation	MILP

3 Improving delay robustness using absorption graphs

3.1 Problem statement

We consider the following setting: The desired train runs to be operated in the station are given as input from a network-wide timetabling step. For each train $k \in \mathcal{K}$, this consists of the entry and exit point of the train on the station infrastructure and the exact arrival and departure times at the platform (with potentially some small allowance for time shifts). No prior input on the platform or route choice within the station is provided, as this may not be available from a macroscopic network plan.

For each train, this input implies a set of possible train paths $\mathcal{P}_k$ throughout the station, each defining the platform assignment, the physical route to and from the platform as well as the exact time at which the train travels along the route. The goal of the methodology developed here is to assign each train a path such that all train paths are conflict-free according to the signaling logic and the timetable is robust towards small delays. Thus, the resulting path assignment extends the existing macroscopic timetable into a microscopic station plan and exposes local infeasibilities or robustness concerns, which could then act as feedback for potentially adapting the network-wide plan.

As a metric, the expected secondary delay impact of a random isolated primary delay smaller than a threshold H (with order of magnitude of a few minutes) is considered, where the primary delay is assumed to follow a modified exponential distribution. Since such small deviations from the timetable are practically unavoidable, this should produce a routing that allows for smoother operation under everyday conditions without the need for manual intervention.

These robustness considerations are included directly into a mixed-integer optimization model, implementing delay propagation through the concept of absorption

times and, thus, defining the local allocation of margins in the timetable. The model is capable of representing chains of delay propagation and incorporating the effect of dwell-time supplements, while making some simplifying assumptions in order to obtain a tractable problem: No dispatching actions in response to delays or changes of the pre-determined train order on any infrastructure segments are considered; the interaction of multiple (potentially correlated) primary delays is not optimized; and time loss due to unplanned braking and accelerating is not taken into account.

3.2 Absorption times for direct and indirect delay propagation

The risk of delay spreading within a station increases when multiple trains use the same resource in close temporal proximity. The predominant approach to avoid this is to minimize a weighted cost function of the buffer times between consecutive trains. In a delay scenario, this buffer time can then absorb delays of the preceding train and avoid impact on the following train, thus improving the robustness of operations. In particular, large weight is put on widening small buffers, which accounts for the fact that small buffers are both violated more often and the propagated delay in case of disruptions is larger.

However, buffer times in isolation only capture the influence that a train has on its direct successor. For highly utilized stations with complex layouts, it is likely that trains form "chains" of delay propagation, where the delay of one train cascades to multiple other trains. The avoidance of such chains is not considered in purely buffer-based methodologies, as exemplified in Fig 1: As the buffer times present in the two depicted dependency graphs are identical, their robustness is indistinguishable from the perspective of a buffer-based methodology. However, in the left dependency structure, a sufficiently high delay of train A indirectly affects also all other trains; an adverse cascading effect that would not occur in the right dependency structure, suggesting overall reduced delay propagation and improved robustness. In other words, one conceptual shortcoming of buffer-based methodologies is that the position of the buffer within the framework of the timetable is not considered.

To overcome this limitation, we propose to measure and optimize the robustness of the station routing more directly by including a representation of the delay propagation within the mathematical optimization model. This requires tracking the delay

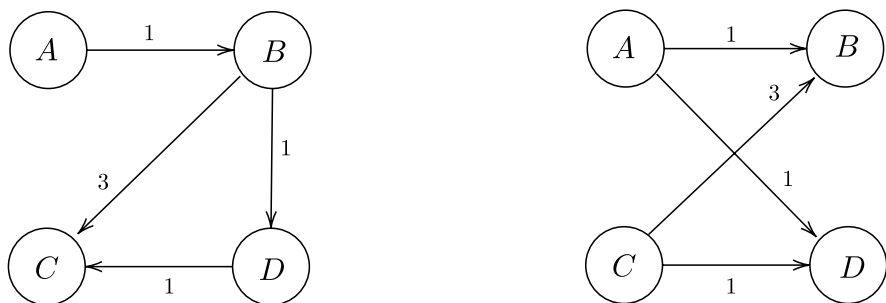


Fig. 1 Example of two different, fictive train dependency graphs. The nodes represent trains, while the edges represent buffer times between them

of the train at certain points of its journey through the station, which we denote as events $\mathcal{E}$. Of particular interest is the set of primary events $\mathcal{E}_{prim} \subset \mathcal{E}$, where delay can originate within the analyzed station context, and the set of secondary events $\mathcal{E}_{sec} \subset \mathcal{E}$, where the occurring delay has an adverse effect on the overall system. The focus of our approach is to view the station as a potential catalyst for network delays: This perspective aims to translate primary delays, with which a train enters the station context or which it suffers due to a delay when departing from the platform, into secondary delays that trains carry back into the network when leaving the station.

Previous work by Caprara et al. (2014) modeled such secondary delays directly in an optimization model, though focused only on worst case delays in a recoverable robust view. We aim to extend this approach into a more stochastic view. As a tool, we utilize a concept described for timetable assessment by Delorme et al. (2009), which we refer to as the absorption time between two events. Differently from Delorme et al. (2009), who use the absorption time for the analysis and comparison of timetables only, we aim to include it directly in the modeling and optimization step.

The absorption time between two train events is defined as the smallest time by which the first train event can be delayed without influencing – directly or indirectly – the second train event. Of course, some trains are fully independent of each other, e.g., because they operate on distinct parts of the station infrastructure. For ease of notation, we define the absorption for such pairs as infinite. We can immediately deduce that the buffer time between two trains forms a special case of the absorption time, as the buffer represents a direct dependency between two trains. However, absorption additionally covers indirect dependencies between events and, in this sense, it is a natural extension of buffer times.

There is a close link between the absorption times and the observed train delays: Assume that a delay d_e is present on one train event e , which has an absorption link with time $a_{e,e'}$ to one succeeding event e' . By definition, the delay of the succeeding train $d_{e'} = 0$ as long as $d_e \leq a_{e,e'}$. If the initial delay is higher than the absorption time, the trains begin influencing each other. In that case, we assume that the absorption times form a linear complement to the delays, i.e., exactly the amount of delay not covered by absorption is passed on to e' . Taking the possibility of multiple predecessors as well as a possible primary delay $\delta_{e'}$ into account, the delay $d_{e'}$ according to our model is given by

$$d_{e'} = \max \left(\delta_{e'}, \max_{e \in \mathcal{E}} (d_e - a_{e,e'}) \right). \quad (1)$$

Thus, when only one other event influences e' and no primary delay $\delta_{e'}$ occurs for e' itself, we assume that $d_{e'} = d_e - a_{e,e'}$. Otherwise, the delay of e' is determined as the maximum of all influencing factors. These assumptions, which are consistent with Delorme et al. (2009), sacrifice some accuracy about the realistic time necessary for unplanned slowdowns or stops, as well as omitting potential dispatching actions, in order to obtain a tractable optimization model.

Note that, under these modeling assumptions, the absorption time equals the length of the shortest directed path from e to e' in the dependency networks introduced previously in Fig. 1. Extending our previous example by looking at the left dependency

structure of Fig. 1, we visualize the difference between buffer and absorption times in Fig. 2, where subfigure (a) represents the previously seen buffer structure and (b) translates this into an absorption graph where both direct and indirect dependencies are made explicit.

Finally, besides the inclusion of indirect dependencies between trains, we also include the influence of dwell-time supplements on the spread of delay within the station. This aspect, which is often absent in previous work, arises from the same idea that some buffers (of equal size) are more valuable than others: If a train is planned with an extended stopping time, some delay during its arrival may be compensated within the station and not effect the larger network. Dwell-time supplements can be included by segmenting the train path into different events and introducing buffer values between the segments. A similar treatment would be possible for travel-time supplements, though we do not include them here: Due to the rather short travel times in the station area, their size is typically negligible.

3.3 Cost function for absorption

The overall goal of our delay robustness objective is to enable the station operation to perform smoothly under everyday conditions with small delays, without the need for dispatching measures. In this spirit, we are less interested in giving worst case guarantees – as was done by Caprara et al. (2014) – but rather in finding a routing and timetable plan that performs well in a wide range of cases. This naturally leads us to a stochastic analysis of the occurring secondary delays for different routings.

Within the station context, there are different timing points of particular interest for delay propagation: Trains may suffer a primary delay at the entry into the station infrastructure (because they arrive delayed from the network) as well as during the departure from the platform (if the passenger exchange took longer than anticipated). These events form the primary events $\mathcal{E}_{prim}$. Conversely, trains may cause adverse effects if they exit the station infrastructure delayed (carrying secondary delay back into the network), denoted by $\mathcal{E}_{sec}$.

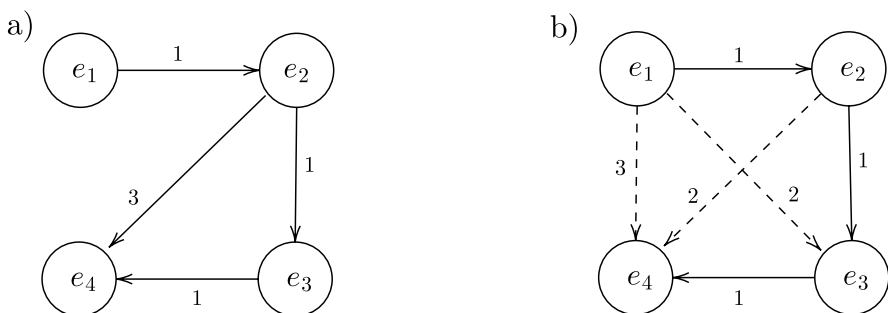


Fig. 2 Comparison of (a) buffer and (b) absorption times for a fixed schedule, continuing the left example of Fig. 1. Solid arrows show relations for which the buffer time equals the absorption time, dashed arrows show relations where the absorption time differs from the buffer time. For instance, on the relation (e_1, e_4) , there is no direct buffer, but an absorption of 3 caused by an indirect dependency. On relation (e_2, e_4) , the absorption of 2 falls below the direct buffer of 3 due to indirect dependencies

Our objective evaluates the *expected* delay impact on the exit events

$$c_{\text{delay_exp}}(x) = \sum_{e' \in \mathcal{E}_{\text{sec}}} \mathbb{E}[D_{e'}^x], \quad (2)$$

where $D_{e'}^x$ is a random variable for the delay of event e' under routing x . The distribution of these outbound delays depends both on the propagation model as well as on the distribution of the primary delays. The relationship (1) between primary and secondary delays is the maximum over multiple terms that are linear in the absorption times. In order to obtain a tractable objective based on absorptions, we only focus on evaluating the expected delay impact caused by a random, isolated primary delay D at one of the input events. While it is, in general, desirable to achieve robustness against multiple simultaneous delay sources, this is not considered here, as different delay sources or their effects may (partially) cancel each other out, which would severely complicate the objective function. However, as is empirically demonstrated on our test cases in Section 5, the routings optimized for isolated perturbations seem to generalize well to multiple perturbations.

Thus, we assume the following underlying random process for choosing the primary delay: First, choose one primary event $e \in \mathcal{E}_{\text{prim}}$ uniformly at random. Then, determine the delay according to a suitable distribution for primary delays, e.g., by a modified exponential distribution widespread for predictive modeling of delays (cf. Schwanhäuser 1974; Weik et al. 2016; Goverde et al. 2001; Yang et al. 2019): With a discrete likelihood of $1 - p$, a train is on time and thus $D = 0$. Otherwise, the primary delay follows an exponential distribution with scale parameter μ equal to the expected delay among all actually delayed trains. However, delays arising under this distribution are unbounded. As the delay propagation model presented here is most apt for small delays, in our methodology we restrict the primary delay to lie in the interval $[0, H]$ by resampling larger delays. This leads to the following continuous density component for the primary delays

$$f : [0, H] \rightarrow \mathbb{R} \quad (3)$$

$$f(D_e) = \frac{p_e}{K} \cdot \frac{1}{\mu_e} \cdot \exp\left(-\frac{D}{\mu_e}\right), \quad (4)$$

with an appropriate normalizing constant $K = 1 - \exp\left(-\frac{H}{\mu_e}\right)$ to account for the restriction to the interval $[0, H]$.

We can then find a closed form solution for the objective defined previously:

$$c_{\text{delay_exp}}(x) = \sum_{e' \in \mathcal{E}_{\text{sec}}} \mathbb{E}[D_{e'}^x] \quad (5)$$

$$= \sum_{e' \in \mathcal{E}_{\text{sec}}} \mathbb{E}\left[\max\left\{0, \max_{e \in \mathcal{E}_{\text{prim}}} (D_e - a_{e,e'}^x)\right\}\right] \quad (6)$$

$$= \frac{1}{|\mathcal{E}_{prim}|} \sum_{e' \in \mathcal{E}_{sec}} \sum_{e \in \mathcal{E}_{prim}} \mathbb{E} [\max \{0, D_e - a_{e,e'}^x\}], \quad (7)$$

$$= \frac{1}{|\mathcal{E}_{prim}|} \sum_{e' \in \mathcal{E}_{sec}} \sum_{e \in \mathcal{E}_{prim}} \frac{p_e \left(-1 + \exp \left(\frac{H - a_{e,e'}^x}{\mu_e} \right) + \frac{1}{\mu_e} (a_{e,e'}^x - H) \right)}{\mu_e \left(-1 + \exp \left(\frac{H}{\mu_e} \right) \right)} \quad (8)$$

where we used after (6) that each primary event $e \in \mathcal{E}_{prim}$ is sampled with the same likelihood $\frac{1}{|\mathcal{E}_{prim}|}$ and thus represents the maximum. The inner component signifies the expected impact of one primary delay for event e onto one secondary delay for event e' . Notably, the routing variable x influences this term only through its effect on the absorption times $a_{e,e'}^x$. The function is convex and can thus be approximated by piecewise linear components using an epigraph formulation, i.e., by including an additional continuous cost variable and constraints that bound it from below on all piecewise-linear components of the objective.

The proposed objective offers several benefits: By minimizing expected delays, it is directly interpretable and can help in contextualizing the achieved optimization improvements. Furthermore, it is easy to include more sophisticated assumptions on the distribution of primary delays: Distribution parameters can be chosen depending on the train type or even based on individual train statistics. The approach is not limited to a modified exponential distribution either, but could incorporate arbitrary, positive-valued distributions if the inner expectation is recomputed for the new distribution.

To assess its practical quality, we compare this absorption-based methodology with a classical buffer-based objective, which typically weights buffers such that particularly small buffer times receive higher penalties. An example is a capped version of the inverse function, as proposed by Dewilde et al. (2014):

$$C_{buffer_inverse}(x) = \sum_{e \in \mathcal{E}_{prim}} \sum_{e' \in \mathcal{E}_{sec}} w(b_{e,e'}^x), \quad (9)$$

$$w(b) = \begin{cases} 20 - 100 \cdot \frac{b}{60}, & \text{if } 0 \leq b \leq 6 \\ 1/\frac{b}{60}, & \text{if } 6 \leq b \leq H \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

The penalty constant for a buffer time of 0 has been increased slightly compared to the original formulation, in order to ensure that the function is convex on its whole domain. Additionally, as we implement buffer minimization as a special case of our model formulation, we also utilize a linearized version of this objective.

All objectives are to be minimized and are included in the MILP through a piecewise-linear approximation with granularity of 15s. Quantities representing times (e.g., buffer and absorption times or delays) are represented in seconds.

4 Modeling delay propagation in stations

So far, we have described absorption as a theoretical concept and utilized it to define fitting objectives to guard against delay propagation. In the following, we develop a mathematical optimization model which allows us to express the absorption times efficiently, which is necessary due to their more complex nature: The buffer time between two trains depends exclusively on the choice of the routing and timetable for these two trains. Thus, there is an inherently quadratic nature to buffer times, and a model may, e.g., pre-compute pairwise buffer times for all train paths and use them in a linearized objective based on the path choices, as is done by Dewilde et al. (2014).

In comparison, the absorption time between two trains may depend not only on these two trains themselves, but also on the schedule and routing of - in principle - any number of other trains, which could be part of the shortest delay propagation path between them. Clearly, pre-computing absorption times and formulating a linearized objective for all possible combinations in such a system is impractical. Instead, it is necessary to describe the absorption times themselves as variables in the model and include the direct train dependencies as constraints relating the absorption variables with each other. This model is similar to the delay propagation model proposed by Caprara et al. (2014), which operated directly on worst-case delays.

As a pre-requisite before analyzing robustness, it is necessary to formulate constraints that ensure the feasibility of the selected train routing, i.e., to make sure no train conflicts occur. To this end, we integrate a modified version of a multicommodity-flow formulation proposed by Caimi et al. (2011) into our model, which was previously used for finding feasible – but not robust – station routings and showed empirical performance gains compared to classical formulations.

In the following, the components of the optimization model are developed: We begin by describing the chosen infrastructure segmentation and granularity in Section 4.1, followed by the encoding of train paths in Section 4.2. Section 4.3 details constraints ensuring the feasibility, while Section 4.4 focuses on the model of absorption times.

4.1 Infrastructure model

Under block-based signalling systems, the safety of train movements is ensured by reserving infrastructure segments (block segments) for the exclusive use by one train at a time. The blocking time of these segments refers to the time frame the section is reserved in the signalling system, which, in addition to the actual block occupation by the train, also features processing times, as well as approach and clearing times enforced by the interlocking. For a more detailed discussion of the blocking time model and the interplay between block occupation and signalling systems, see, e.g., (Pachl 2018).

Block segments, which typically span from one signal to the next, exclude each other even if they overlap only partially. Furthermore, while they have to be reserved in full initially, some interlocking schemes allow the partial release of sections that have already been cleared by the train. For our modeling, we thus split the infrastruc-

ture into more fine-grained *track sections*, which form parts of the infrastructure such that

- (i) two different track sections do not overlap,
- (ii) a train always reserves and releases a track section in full, i.e., the blocking time is constant over the whole track section

In practice, these track sections may end, e.g., at signals, as well as after switches and crossings, where train detection systems enable the previously mentioned sectional release schemes. Figure 3 depicts a simple, exemplary station infrastructure, with dotted boxes symbolizing the split into track sections.

This infrastructure representation is the building block for tracking the spread of delay at the same, fine-granular level: Where previous methods considered full inbound or outbound train paths as one event, respectively (Caprara et al. 2014), we now track absorption times per track section. This includes potentially valuable problem structure in the form of similarities between different train paths in the model.

In case of delays, it allows encoding the assumption that the train travels up to the signal closest to the conflict and waits there until its path clears, which may be advantageous in larger station areas. Additional time necessary for acceleration and deceleration at such unplanned stops is not included in the model for complexity reasons, meaning that the propagated delay may be slightly underestimated. This effect is lessened, though, as many delays simply cause prolongations of planned stops (at the platform) and the speed of travel in stations is slower than on the open line. The feasibility of the planned solution remains unaffected, as no stops outside the platform can be planned by the model.

4.2 Train path model

We now utilize the structure provided by the infrastructure in our representation of the train paths through the station. Let $\mathcal{K}$ be the set of planned train movements, which are characterized by their entry and exit points in the station infrastructure, their planned arrival and departure times at a platform and potentially a pre-determined subset of valid stopping platforms. Note that these train movements comprise both the inbound and outbound segments together. We now convert these train movements into a collection of potential train paths that describe both the physical route as well as the timing, and then set up an optimization model that chooses the most appropriate.

In an automated pre-processing step, all topologically feasible routes through the infrastructure are determined, before eliminating those that are operationally non-

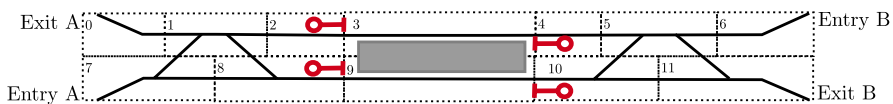


Fig. 3 A small station infrastructure containing signals (red), switches and a platform (gray). The dotted lines split the infrastructure into track sections, identified by small numbers for later reference. Entries and exits are the connection points to the open line

sensible (e.g., because they include brief detours through the opposite direction track). Then, we compute blocking times for each train movement along all compatible routes, forming train paths, and split the train paths along the infrastructure track sections. The blocking times are computed with a one-second accuracy. A lower accuracy would not reduce model size (as all of the train paths are treated as discrete options), but would run the risk of overestimating blocking times and, thus, leaving capacity unused.

Thus, after pre-processing, we obtain a set of train-path options $\mathcal{P}_k$ for each train $k \in \mathcal{K}$, where each train path $p_k \in \mathcal{P}_k$ consists of a sequence of *track section reservations* $(e_{i_1}, \dots, e_{i_m})$. Each track section reservation e is defined by the tuple $(k(e), \text{phase}(e), \text{ts}(e), t_{\text{start}}(e), t_{\text{end}}(e))$, encapsulating the train $k(e)$, the phase (i.e., inbound, platform stop or outbound movement) $\text{phase}(e)$, the physical track section $\text{ts}(e)$ as well as the start and end of the associated blocking time $t_{\text{start}}(e), t_{\text{end}}(e)$, respectively. For quick reference, the list of all sets and variables for both the feasibility and the robustness model is also collected in Table 2.

Different train paths belonging to the same train often share track section reservations, particularly at the platform, where arrival and departure times are fixed. Thus, organizing them in a way that makes use of this structure has the potential to significantly reduce the number of variables and capture valuable structure for the optimization model. To this end, the train-path options are organized in a flow graph

Table 2 List of sets and variables

Sets and constants	
$\mathcal{E} = \mathcal{E}_{\text{prim}} \cup \mathcal{E}_{\text{sec}}$	Set of relevant events, split into events where primary delay occurs and where secondary delay is evaluated
$\mathcal{K}$	Set of train movements from entry to exit
$\mathcal{P}_k$	Set of all possible train paths for the train $k \in \mathcal{K}$
$\text{Phase} = \{\text{in}, \text{platform}, \text{out}\}$	Set of phases within the train movement
$\text{TS} = \{1, \dots, q\}$	Set of physical track sections
$E = \{e_1, \dots, e_m\}$	Set of edges in the train path graph, each representing a track section reservation $e \in E$ encoding the train $k(e)$, phase $\text{phase}(e)$, track section $\text{ts}(e)$ and start and end of the associated blocking time $t_{\text{start}}(e), t_{\text{end}}(e)$
$\mathcal{O} = \{o_1, \dots, o_n\}$	Occupation segments with $o_i = (k(o_i), \text{phase}(o_i), \text{ts}(o_i))$. The associated track section reservations are denoted by $E(o_i) \subset E$.
$\mathcal{L} = \{(e, f, \text{sup}_f) \mid e, f \in \mathcal{O}, \text{sup}_f \geq 0\}$	Links between consecutive occupation segments e, f in a train path, with a dwell-time supplement sup_f
$\mathcal{B} = \{(e, f, c, \bar{b}^{e,c}, \vec{b}^{e,c}, \vec{b}^{e,c})\}$	Delay propagation dependencies between train paths, on occupation segments $e, f, c \in \mathcal{O}$, with buffer components $\bar{b}^{e,c} \in \mathbb{R}, \vec{b}^{e,c} \in \mathbb{R}^{ E(e) }, \vec{b}^{e,c} \in \mathbb{R}^{ E(c) }$
$H \in \mathbb{R}^+$	Largest considered delay horizon
Model variables	
$x_e \in \{0, 1\} \forall e \in E$	Choice of edges
$u_o \in \{0, 1\} \forall o \in \mathcal{O}$	Implied usage of occupation segments
$d_o^p \in [0, H] \forall p \in \mathcal{E}_{\text{prim}}, o \in \mathcal{O}$	Delay on occupation segment o caused by a primary delay of H at event p
$a_s^p \in [0, H] \forall p \in \mathcal{E}_{\text{prim}}, s \in \mathcal{E}_{\text{sec}}$	Absorption between primary event p and secondary event s

(V, E) , with edges E associated with track section reservations and nodes V simply acting as connecting points.

The graph network is set up iteratively over all train paths. Starting at the platform track section reservation in the middle of a train path, if there already is a fitting edge in the graph, the track section reservation is identified with this edge. Otherwise, a new platform edge is created. Then, we move forwards and backwards along the train path, reusing existing graph edges as long as the track section reservation fully matches an already existing one and inserting a new, branching edge if this is no longer the case. Finally, auxiliary edges connect the first and last track section reservation of each path with train-specific source and sink nodes, which in turn are connected to overall source and sink nodes of our flow graph. An example for such a resulting flow graph for two trains is depicted in Fig. 4.

This flow graph branching from the platform track section reservations in the middle is inspired by, but more complex than in the work by Caimi et al. (2011): There, the platform assignment was fixed and only the routing was optimized. Thus, the authors fully decouple the inbound and outbound paths and build the subgraphs of inbound and outbound movements separately in a tree-like structure. In comparison, here the platform to use is also optimized and we thus have to make sure that inbound and outbound paths are represented together.

4.3 Feasibility constraints

Using the flow graph, we can represent constraints for the feasibility of the train routing:

$$\sum_{e \in E^+(v)} x_e = \sum_{e \in E^-(v)} x_e \quad \forall v \in V \setminus \{v_{\text{source}}, v_{\text{sink}}\} \quad (11)$$

$$\sum_{e \in C} x_e \leq 1 \quad \forall C \in \mathcal{C} \quad (12)$$

$$x \in \{0, 1\}^{|E|} \quad (13)$$

The binary variables x_e represent the choice of a track section reservation. A complete path from source to sink corresponds to a valid train path, and this completeness of the paths is guaranteed through the flow conservation constraints (11), where $E^+(v)$ and $E^-(v)$ are the outgoing and incoming edges of node $v \in V$, respectively. Note that the subgraph for each train is connected by exactly one edge to the overall source and sink nodes, ensuring that we do not assign multiple paths to one train. If one complete path can be assigned to each train, then the full timetable can be operated.

Additionally, no blocking time conflicts between different train paths may be present in a feasible solution. While it would be possible to formulate constraints for pairwise exclusions of conflicting track section reservations, this leads to a comparatively

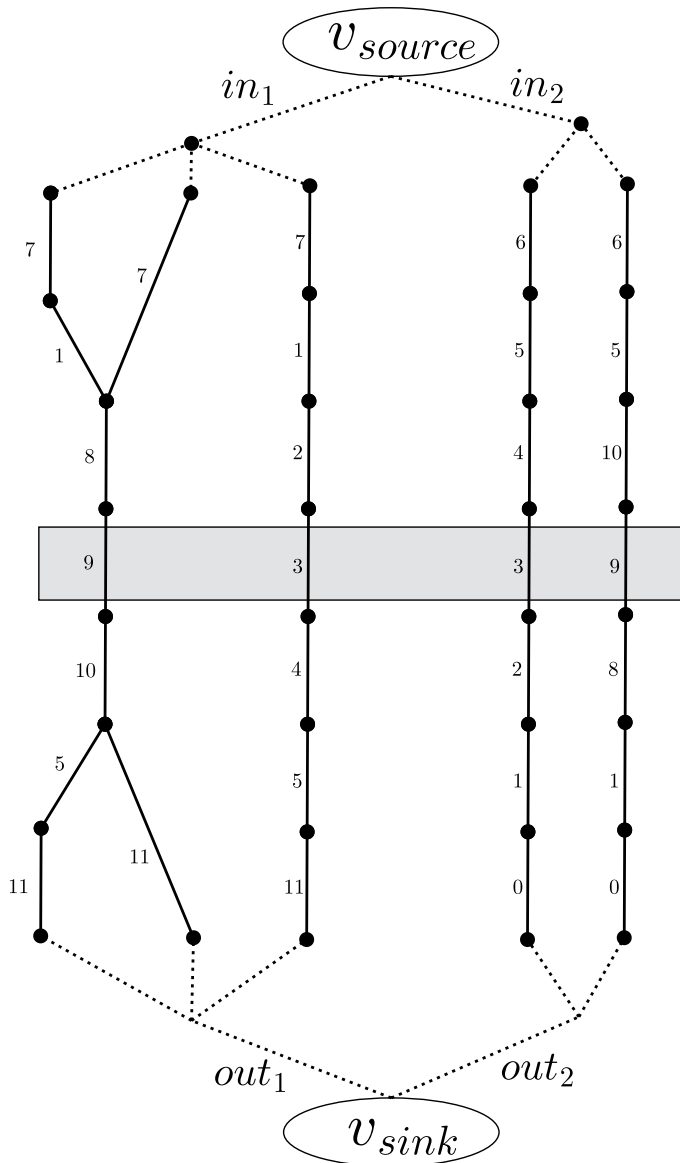


Fig. 4 Representation of train-path options as a flow graph for two train movements on the station infrastructure depicted in Fig. 3. The left subgraph for $train_1$ corresponds to a train movement from Entry A to Exit B, the right subgraph for $train_2$ to a train movement from Entry B to Exit A. Each edge represents a track section reservation, which is annotated by the physical track section it belongs to, while the gray bar marks the platform track section reservation. As an example, the left-most path in the visualization runs along the track section reservations 7, 1, 8, 9, 10, 5, 11 – corresponding to a route in the station infrastructure with two detours. It shares the central track section reservations 8, 9, 10 around the platform with the more direct route

loose model relaxation. Instead, we employ clique constraints (12) of multiple track section reservations, out of which only one may be chosen.

These are generated through the same approach presented by Caimi et al. (2011) to efficiently determine maximal conflict cliques $\mathcal{C}$ on a track section level. Briefly summarized, this approach comprehends the blocking time intervals for each physical track section as an interval graph. Then, in order to determine incompatibility cliques, it is sufficient to consider the release times of all track section reservations and form cliques $C \in \mathcal{C}$ of those that are active at this point in time. For each clique, a separate constraint is formulated. As shown by the authors, this captures all maximal conflict cliques on a track section level under mild assumptions on the structure of the reservation times that are fulfilled in practice.

The objective of the feasibility component is to find a routing for as many trains as possible. From a practical standpoint, this maximization of the number of routed trains is typically seen as more important than improving robustness. This is implemented using a two-step, hierarchical process, which first determines the number of feasible trains and then optimizes for robustness only among solutions that include as many trains as possible. We approach this by first determining the maximum source-sink-flow through the graph by optimizing $k^* = \max \sum_{e \in E^+(v_{\text{source}})} x_e$ on the pure feasibility model (11) - (13). Then the following constraint is added to the full robust model:

$$\sum_{e \in E^+(v_{\text{source}})} x_e = k^*, \quad (14)$$

An equivalent alternative to this hierarchical process would be directly solving the full, robust model using a weighted objective, containing both the robustness metric and the number of trains with a sufficiently high coefficient. However, apart from potential numerical issues due to the large coefficient required, empirical tests also indicated that this combined model would incur slower solution times compared to the chosen hierarchical approach.

4.4 Absorption constraints

4.4.1 Derivation and model formulation

In the following, we present a model formulation that encodes the absorption depending on the routing choice x defined in the feasibility model, which can then be used to describe the expected delay propagation objective.

We do so by setting up $2|\mathcal{K}|$ networks, each used to compute the absorption from a single primary event $p \in \mathcal{E}_{\text{prim}}$ at the station entrance or the platform onto all evaluation events $\mathcal{E}_{\text{sec}}$. This is closely related to the scenario-expansion approach utilized by Caprara et al. (2014). Instead of working directly with absorptions, we formulate our model for fixed input delays H and in the end utilize the complementary relation (1). This improves the readability of some of the additional constraints.

At its core, the delay network for each primary event $p \in \mathcal{E}_{prim}$ captures how much delay each train has when completing the individual parts of its journey through the station. Formally, it contains delay variables $d_o^p \in [0, H]$ for each occupation segment $o = (k(o), phase(o), ts(o)) \in \mathcal{O}$ associated with the train $k(o)$, its track section $ts(o)$ and phase $phase(o)$ of traveling through the station. Note that multiple track section reservations of the same train, phase and track section belong to the same occupation segment o even if they have differing blocking times, and this set of track section reservations is denoted by $E(o) \subset E$. In this sense, each occupation segment forms an envelope over the corresponding track section reservations used in the feasibility model. Such differing blocking times may occur when a train uses the same segment along different paths, but at slightly different times; or when train-path options with shifted arrival and departure times are included in the model. Preliminary tests showed that this formulation offers a better computational performance than modeling delay individually for each track section reservation $e \in E$.

Delay in this network spreads in two ways: Within a train path, delay is passed on from one segment to the next, potentially reduced by a dwell-time supplement (while travel-time supplements can be included through the same concept, we omit them from consideration in this paper). Thus, the delay variables must be connected along the links given by the set of all consecutive occupation segments

$$\mathcal{L} = \{(e, f, sup_f) \mid e, f \in \mathcal{O}, sup_f \geq 0\}, \quad (15)$$

which is extracted from the flow graph (V, E) described in Section 4.2. Here, e is the earlier and f the following occupation segment, with a time supplement of sup_f during segment f .

The more complex case concerns how delay is passed between train paths: If two trains use the same resource in quick succession (i.e., with buffer time smaller than the considered delay horizon H), a delay of the first train on the *initial* occupation segment e propagates to the second train due to the *conflicting* occupation segment c in case of a sufficiently large delay. However, the delay is not passed on directly at this point: To comply with the signaling logic and ensure safe train operations, the second train has to wait before the last signal, at the first *affected* occupation segment f to which delay is propagated.

The value of the propagated delay depends on the available buffer time between the occupation segments. Encoding this time in a linear constraint requires some care, as it depends not only on the occupation segment, but rather on the underlying track section reservations of both trains. To avoid a quadratic dependency, we split the buffer time into three components: A base buffer independent of the choice of exact track section reservations and two components which add the contribution of the first and second train independently.

This split is visualized in Fig. 5, which shows the blocking times for different possible track section reservations. The reference points of the split are $r_2^{e,c}$, the earliest possible start time of the second train, and $r_1^{e,c}$, the minimum over $r_2^{e,c}$ and the last possible release time of the first train. We then obtain a base buffer $\bar{b}^{e,c} = r_2^{e,c} - r_1^{e,c}$, as well as the individual components

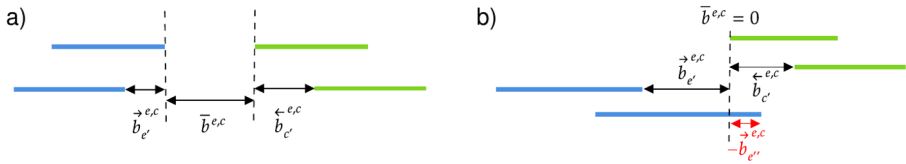


Fig. 5 Visualization of buffer constants. A colored line represents the blocking time of a potential track section reservation, with all same-colored lines belonging to one occupation segment (i.e., to the same train, phase and track section). The dotted lines are the reference points and the arrows show the buffer times, with red indicating a negative buffer constant. Case (a), where all combinations of track section reservations of the two occupation segments are compatible, is more commonly encountered. Meanwhile, case (b) shows the effect if some of the possible track section reservations of the two trains overlap and others are feasible

$$\vec{b}_{e'}^{e,c} = r_1^{e,c} - t_{end}(e') \quad \forall e' \in E(e) \quad (16)$$

$$\bar{b}_{c'}^{e,c} = t_{start}(c') - r_2^{e,c} \quad \forall c' \in E(c) \quad (17)$$

Note that both the base buffer as well as the buffer contribution of the second train $\bar{b}_{e,c}$ are always non-negative by definition, while the buffer contribution of the first train $\vec{b}_{e',c}$ may be negative if there are some overlapping track section reservations as depicted by the red arrow in Fig. 5. If a negative buffer exists, this must later be accounted for when setting the associated big-M constants in corresponding constraints.

Formally, we collect the potential buffer dependencies between occupation segments in the set

$$\mathcal{B} = \{(e, f, c, \bar{b}_{e,c}, \vec{b}_{e',c}, \bar{b}_{c',e}) \mid e, f, c \in \mathcal{O}, \bar{b}_{e,c} \in \mathbb{R}, \vec{b}_{e',c} \in \mathbb{R}^{|E(e)|}, \bar{b}_{c',e} \in \mathbb{R}^{|E(c)|}\} \quad (18)$$

One limitation of this representation is that the order between all track section reservations of $E(e)$ and $E(c)$ must be identical, i.e., for each physical track section it must be known beforehand if one train uses it before or after another. This is a valid assumption in our setting where the arrival and departure times are fixed by a network timetable. For settings where a large shift in the arrival and departure times should be considered (which can be represented through additional, time-shifted track section reservations), an extension using order variables seems possible. As this is not the focus of the current work, such an extension is left for future work.

If the timetable is periodic, the buffer dependencies $\mathcal{B}$ should also contain delay propagation links between periods, modeled simply by including additional propagation links from the "last" to the "first" trains. While this generates a cyclic delay propagation structure, we only analyze robustness against small, individual delays. For our use case, these initial delays never propagate to trains that are more than one period length in the future – thus, no more advanced modeling choices are required for the optimization approach.

Consolidating these thoughts, Fig. 6 shows an example of how delay propagation is considered within and between train paths. Crucially, when including these delay propagation links in the model, it must be ensured that delay links both within and

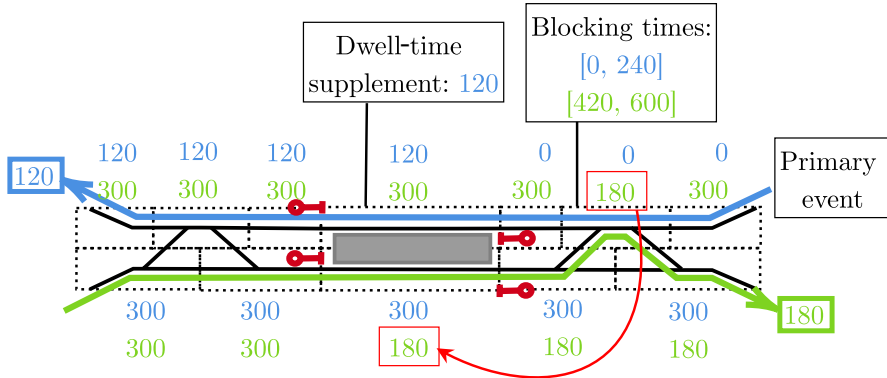


Fig. 6 The graph depicts the absorptions for primary delays at the entry event of the blue train, assuming the two highlighted train paths are chosen. Each track section is annotated with the absorption time of each train. Track sections that are not chosen for a train may receive the maximal absorption time $H = 300$. As there is a buffer of $420 - 240 = 180$ seconds between the blocking times on the common segment, the absorption value of the green train is 180 seconds starting from the previous signal. Here, the red arrow symbolizes the realized delay propagation relation. Independently, the blue train gains an absorption value of 120 seconds thanks to the available dwell-time supplement at its platform stop. As a result, the relevant absorption values between the entry of the blue train and the train exits from the station are 180s (green) and 120s (blue), respectively. Intuitively, this means that primary delay of the blue train entering the station can be compensated up to these values

between train paths are only active if the corresponding track section reservations are actually chosen in the solution, linking the absorption networks to the feasibility constraints. We employ a big-M formulation to model this, leading to the overall optimization problem

$$\min c(\{a_{p,s} \mid p \in \mathcal{E}_{prim}, s \in \mathcal{E}_{sec}\}) \quad (19)$$

$$\text{s.t. } a_{p,s} \geq H - \sum_{e \in \mathcal{O}(s)} d_e^p \quad \forall p \in \mathcal{E}_{prim}, s \in \mathcal{E}_{sec} \quad (20)$$

$$u_o = \sum_{e \in E(o)} x_e \quad \forall o \in \mathcal{O} \quad (21)$$

$$d_e^p = H u_e \quad \forall p \in \mathcal{E}_{prim}, e \in \mathcal{O}(p) \quad (22)$$

$$d_f^p \geq d_e^p - \text{sup}_f u_f - M(1 - u_f) \quad \forall p \in \mathcal{E}_{prim}, (e, f, \text{sup}_f) \in \mathcal{L} \quad (23)$$

$$d_f^p \geq d_e^p - \bar{b}^{e,c} - \sum_{c' \in E(e)} \bar{b}_{e',c}^{e,c} x_{e'} - \sum_{c' \in E(c)} \bar{b}_{c',c}^{e,c} x_{c'} - M(1 - u_f) - M(1 - u_c) \quad (24)$$

$$\forall p \in \mathcal{E}_{prim}, (e, f, c, \bar{b}^{e,c}, \bar{b}^{e,c}, \bar{b}^{e,c}) \in \mathcal{B}$$

$$d_o^p \in [0, H] \quad \forall p \in \mathcal{E}_{prim}, o \in \mathcal{O} \quad (25)$$

$$a_s^p \in [0, H] \quad \forall p \in \mathcal{E}_{prim}, s \in \mathcal{E}_{sec} \quad (26)$$

$$u_o \in \{0, 1\} \quad \forall o \in \mathcal{O} \quad (27)$$

$$\text{feasibility constraints (11) -- (14)}. \quad (28)$$

The objective, as defined in Section 3.3, depends on the absorption between primary and evaluation events.

The first constraint (20) transforms between the delay and absorption view, while (21) is a short-hand for the usage of a specific occupation segment depending on the choice of associated track section reservations. (22) fixes the starting delays for each absorption network to H if the corresponding train is included in the final routing. Then, (23) links delay values within a train path by enforcing that the delay at the later occupation segment must be at least as large as the delay up to the earlier occupation segment reduced by the time supplement. Analogously, constraint (24) links absorption variables along buffer dependencies between trains, but includes the buffer time between the trains as additional slack. To connect the absorption to the feasibility model, both of these propagation constraints are "deactivated" via big-M terms if the corresponding occupation segments are not chosen (i.e., if the train takes a different route).

There are two interesting observations about the delay propagation constraints (23) and (24): First, note that they only enforce a lower bound on the delay. This is sufficient, as all of the cost functions are monotonically decreasing in the absorption values, and it is thus always beneficial to choose the absorption values as high as possible, or conversely the delay as low as possible. The second observation is the lack of a big-M term for the initial occupation segment e . This term is not necessary, as the original delay value d_e^p may be 0 if the corresponding route choice variable $x_e = 0$. This is ensured as whenever d_e^p appears on the left side in a constraint, the upper bound contains a big-M term $M(1 - u_e)$.

4.4.2 Modifications for performance improvement

While the chosen model formulation captures some of the valuable problem structure, both the number of variables and constraints as well as the potentially weak LP relaxation of big-M constraints can pose challenges for the efficient solvability. In the following, we describe some slight modifications that enhance the performance without changing the general idea or optimal value of our solution approach.

Omit unnecessary dependencies: For one, some buffer constraints (24) can be omitted without changing the final absorption values: If the blocking times of two train paths overlap on some track sections, but only lie in close temporal proximity on others, then the latter buffer times are irrelevant as both paths can never be chosen together. Additionally, if multiple consecutive track section reservations from one train (without any incoming buffer dependencies of their own) affect the same track section reservation f of another, then only the dependency with the smallest buffer is relevant. In practice, this occurs when two trains travel along identical routes over

multiple track sections, but the position at which the following train must wait does not change.

Use tight big-M terms: Furthermore, to determine tighter upper bounds for the constants M and thus the right-hand side, we can compute the absorption while treating *all* track section reservations as active (i.e., ignoring feasibility completely). As this includes all restrictions on delay that could possibly be imposed, the resulting delay $\hat{d}_e^p$ surely is an upper bound on the actual delay d_e^p of any feasible routing. Therefore, wherever d_e^p appears on the right-hand side, a constant $M_e = \hat{d}_e^p$ is sufficient.

Omit some big-M terms: In addition to these measures, there are opportunities to eliminate big-M terms altogether: First, some of the absorption links (23) within a train path can be transformed into equality constraints without big-M term: If there is neither a merge nor a split in the feasibility flow graph between $(e, f, 0) \in \mathcal{L}$, then flow-conservation constraints (11) ensure that $u_e = u_f$. In case there is no supplement and no delay bounds on f from external buffer dependencies, this implies that $d_f^p = d_e^p$, as either both delay values are unconstrained or they must be identical.

Second, some of the big-M terms can be omitted from the model in the inequalities (24): Observe that each big M term on its own must be sufficiently large to deactivate the constraint. Thus, if we can guarantee that the choice of one track section reservation implies the other, e.g., $x_e = 1 \Rightarrow x_f = 1$, then it is sufficient to keep only the big-M term containing x_e (and vice-versa). With an analogous argument, the big-M terms in (23) can be omitted for links $(e, f, \text{sup}_f) \in \mathcal{L}$. These cases occur, e.g., when one of the occupation segments is part of all possible paths for a train, like at the start and end of its route through the station.

Sum delay propagation within train paths: Through the big-M terms, we ensure that a delay variable of an occupation segment that is not chosen is not constrained from below and may thus be set to 0 (the most advantageous value) during optimization. Thus, we can replace suitable delay variables by a summation over multiple options if we can be sure that only one of them is chosen simultaneously. This occurs naturally at switches and crossings, where a train may naturally be assigned only one of the options for arriving/continuing on this path. As a shorthand notation (and under slight abuse of notation), we write that $(\mathcal{L}_{in}, \mathcal{L}_{out}, \text{sup}_{\mathcal{L}_{out}}) \in \mathcal{L}$ if all combinations of occupation segments in the two sets are part of $\mathcal{L}$, i.e., $(e, f, \text{sup}_f) \in \mathcal{L} \forall e \in \mathcal{L}_{in}, f \in \mathcal{L}_{out}$. Then, we can replace constraints (23) with a tighter, summed version:

$$d_f^p \geq \sum_{e \in \mathcal{L}_{in}} d_e^p - \text{sup}_f u_f - M(1 - u_f) \quad \forall p \in \mathcal{E}_{prim}, (\mathcal{L}_{in}, f, \text{sup}_f) \in \mathcal{L} \quad (29)$$

These constraints provide a benefit for cases where an occupation segment may have multiple predecessors (of which naturally only one can be chosen). Furthermore, we can add additional constraints which address cases where segments have multiple successors:

$$\sum_{f \in \mathcal{L}_{out}} d_f^p \geq \sum_{e \in \mathcal{L}_{in}} d_e^p - \sum_{f \in \mathcal{L}_{out}} \sup_f u_f - M(1 - \sum_{f \in \mathcal{L}_{out}} u_f) \quad \forall (\mathcal{L}_{in}, \mathcal{L}_{out}, \sup_{\mathcal{L}_{out}}) \in \mathcal{L}, p \in \mathcal{E}_{prim} \quad (30)$$

As these constraints do not provide boundaries on the individual occupation segments, they can only be supplied as additional tightening constraints, not as replacements.

5 Results and discussion

We have described a methodology to find a train routing and platform assignment that directly minimizes the expected impact of isolated primary delays. In the following, we demonstrate this methodology and compare it to a buffer-based objective inspired by the literature, as well as the platform assignment currently in operation.

Computational experiments for this section were carried out on a compute server using 8 cores of an Intel(R) Xeon(R) Platinum 8358 CPU and 32GB of RAM. The MILPs were solved using Gurobi 12.0.1 and instantiated from its Python interface (Gurobi Optimization 2024), with a time limit of 24 hours for each experiment.

5.1 Test instances

5.1.1 Stations and timetables

To test our methodology, we consider the stations of Breda, Amersfoort and Arnhem in the Netherlands, for which infrastructure data representing the precise track layout was provided by infrastructure manager ProRail and Nederlandse Spoorwegen. The boundaries of the station were chosen such that the station area includes the full railway node, i.e., including the switch areas connecting the station to adjacent railway lines. Thus, we preserve all possible routing options and inbound and outbound train movements have exactly one sensible start and end point, determined by the correct geographical direction and respecting the convention of right-hand travel in the Netherlands.

Our chosen stations represent a class of railway stations where our methodology seems most apt: While the largest, most important stations of the railway network may directly influence or even shape the network timetable, the plans for many medium-sized stations may have to adhere to and best implement timetabling constraints imposed by a network-level plan. Optimizing this process may thus tackle an important subproblem in the timetabling process.

For the pre-determined network timetable and the corresponding train runs to be operated in the station, we use the hourly passenger and freight movements in 2025 as provided by Nederlandse Spoorwegen. We set the arrival and departure time at the platform as fixed, while the platform assignments themselves remain a free variable for our optimization model.

The timetables are periodic with a one hour period. Thus, it is sufficient to optimize a routing for one hour, which would then be rolled out for the whole day (where the optimization should be carried out on the busiest hour, as it is typically the most critical from a robustness perspective). This periodic construction naturally implies

that hourly services are always assigned to the same platform over the day. For services that are operated at a higher frequency this is not guaranteed a priori, but customarily done in practice. In the interest of constructing a timetable that is attractive for passengers and fairly comparable to the public timetable, we add additional constraints that ensure that higher-frequency lines are always assigned the same platform within the hour.

The infrastructure layout as well as the timetables are visualized in Fig. 7, and characteristics of our test instances are summarized in Table 3. Note that all stations connect to railway infrastructure lines into four different geographical directions

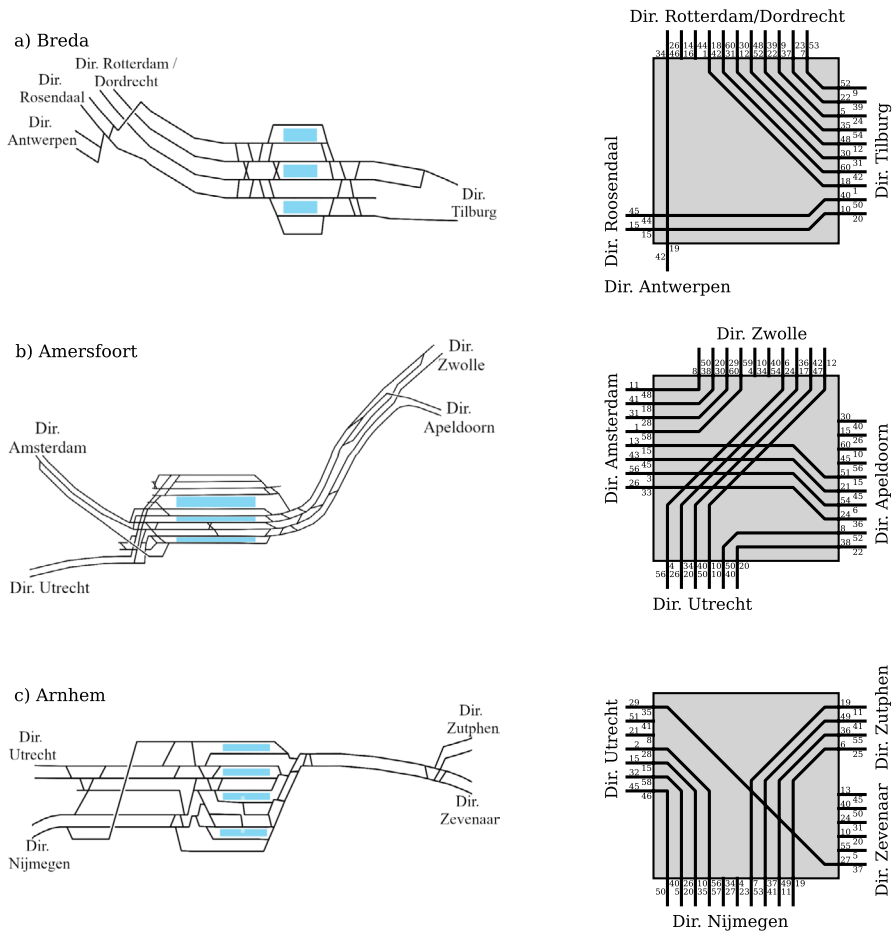


Table 3 Design parameters of railway stations from the Netherlands used as test instances for the robustness optimization.

Instance	# Platforms	# Entering lines	# Trains per hour	# Train paths	# Avg. paths per train
Breda	6	3/1	26	300	11.5
Amersfoort	6	2/2	34	458	13.5
Arnhem	10 (8)	2/2	27	656	24.3

(split into eastbound and westbound entering lines) and are intersection points of multiple railway lines. Compared to intermediate stations connecting only two directions of comparable station design and frequentation, a richer structure of the station routing can hence be observed. In Arnhem, 2 of the 8 platform tracks are split into sub-segments A and B, which are both actively used in operation. We included these as separate possible platform assignments in our model, but restricted the use of these tracks exclusively to short trains. Consistent with current practice, we do not allow a long train to use both of them simultaneously. To assess the scaling behavior of the algorithm, the test instances progressively increase in complexity, as can be seen in the number of train-path options (overall as well as per train). In particular, Arnhem offers a much larger amount of path options per train, mainly caused by the higher number of available platforms.

The potential train routes are first generated by finding all topologically possible routes from the entrances of the station to the platforms and from the platforms to the exits, as well as passthrough routes from entry to exit suitable for freight trains. As pre-processing, we then filter these based on conflict domination: For each route, we determine the set of all other routes that overlap with it at any point. Then, we compare each pair of routes (r_1, r_2) that begin and end in the same point: If the overlapping routes of r_1 are a subset of the overlapping routes of r_2 and the simulated travel time along r_1 is faster than along r_2 , we exclude r_2 from our pool of potential train routes as it is both slower and has a higher risk of propagating delay. The filtering process is repeated on the reduced route set until no new exclusions are generated, which occurs after one or two iterations in the considered examples. In practical application, this omits routes containing unwanted "detours", e.g., where a train would utilize the opposite track for a short segment between two switches (akin to the green route in Fig. 6). Table 3 gives the number of train paths after the filtering procedure is applied, i.e., the number of possible paths that are provided to the optimization model.

5.1.2 Optimization objectives

As the delay propagation model does not consider any remedial actions by the dispatcher (like changing the order of train movements or re-routing trains to an alternative platform during real-time operation), it is most appropriate for assessing robustness against small delays. For this study, we consider primary delays up to a horizon $H = 300$ seconds (i.e., 5 minutes), which lies in the range discussed in the literature (Delorme et al. 2009). This also implies that any buffer that is larger than 300 seconds fully absorbs the delay of the first train. Such delay links can hence be omitted altogether in our model without changing the objective.

We consider the following objective, described in detail in Section 3.3:

- **delay_exp**: Minimizing the expected delay propagation caused by a single (potentially) delayed train whose delay distribution is a modified exponential capped at 300s (by resampling larger values). We choose $\mu = 120s$ for the scale parameter and a delay likelihood $p = 50\%$. The primary delay events $\mathcal{E}_{prim}$ can be the entry into the station or the departure from the platform. The secondary events $\mathcal{E}_{sec}$ for evaluating delay propagation are the exits out of the station infrastructure.

Furthermore, we provide two comparison cases:

- **buffer_inverse**: Minimizing a buffer cost defined as the inverse buffer time between any two train pairs, capped at buffer times of 300s.
- **publicTimetable**: Comparison case using the platform assignments of the public timetable. As data on the actual routes chosen through the station is not available, we compute an associated routing using the **delay_exp** objective with fixed platforms.

The buffer cost minimization is implemented as a direct extension of our model: As noted earlier, the absorption values are determined using a scenario-expansion formulation, where each scenario tracks the absorption from a primary delay event $p \in \mathcal{E}_{prim}$ to all evaluation events in $\mathcal{E}_{sec}$. To compute simple buffer times instead, we can choose train entries as the primary events $\mathcal{E}_{prim}$, train exits as evaluation events $\mathcal{E}_{sec}$ and omit dwell time buffers as well as all edges of the delay propagation network in scenario p that connect two trains but do not originate at the initially delayed train (and would thus model indirect propagation). A more efficient implementation of the buffer-oriented approach may be possible, but was not the focus of this work.

5.2 Model characteristics

For the comparison of the approaches, we apply each optimization objective to all railway station test instances. In addition, we analyze extended scenario variants aimed at understanding the limitations imposed by following the network timetable exactly. In these, some flexibility is allowed with respect to the timing of the trains: The model is allowed to shift each train path by a fixed time forward or backward compared to the given timetable. While the default variant (“shift 0”) is fixed to the given timetable, the variant “shift 15” allows the model to shift each train path individually by -15, 0 or +15 seconds; the case “shift 25” is analogous. Finally, the variant “shift 15 or 25” allows the individual shift options -25, -15, 0, 15 or 25 seconds per train path. The value 25s is the largest possible shift in these instances for which the order of trains on all track sections is still pre-determined regardless of the chosen train path (cf. Section 4.4).

These variants are generated by providing copies of the default train paths with shifted blocking times to the model, essentially acting as a multiplicative factor on the number of train paths. Thus, they allow us to evaluate both the impact of more path choices on the characteristics and solution time of the model, as well as the benefit that small timetable modifications could provide locally to reduce delay propagation.

Table 4 Optimization model sizes and characteristics for objective `delay_exp`

Scenario		Train paths	Base model				Pre-processed model			
Station	Shift		Cont. vars [10 ³]	Binary vars [10 ³]	Con-straints [10 ³]	Non-zero entries [10 ³]	Cont. vars [10 ³]	Int. vars (of that binary) [10 ³]	Con-straints [10 ³]	Non-zero entries [10 ³]
Breda	0	300	132	7	119	324	17	1.0 (0.6)	50	173
	15	900	132	22	225	1,329	28	0.9 (0.9)	90	590
	25	900	132	22	254	1,676	32	1.0 (1.0)	111	779
	15 or 25	1,500	132	37	276	2,693	32	1.5 (1.4)	112	1,116
Amersfoort	0	458	313	12	513	1,687	54	0.3 (0.3)	186	612
	15	1,374	313	34	615	4,656	56	1.8 (1.3)	230	1,818
	25	1,374	313	34	640	5,062	58	1.3 (1.3)	239	1,976
	15 or 25	2,290	313	57	679	8,097	61	2.6 (2.2)	255	3,048
Arnhem	0	656	211	14	414	2,058	57	0.8 (0.8)	204	1,318
	15	1,968	211	42	489	6,725	58	2.7 (2.6)	231	3,920
	25	1,968	211	42	503	7,130	60	2.9 (2.7)	256	4,573
	15 or 25	3,280	211	70	548	12,088	60	4.2 (4.0)	259	7,782

Table 4 shows model sizes and characteristics of the optimization models using the `delay_exp` objective. These characteristics are provided both for the base model which is developed in this article, as well as for the model that is generated by Gurobi's automatic pre-processing, which is aimed at simplifying and making the model more easily solvable.

As noted previously, the number of train paths equals the number of shift options multiplied by the number of base train paths from Table 3. The same scaling behavior applies to the number of binary variables of the base model, as new binary variables are required to represent the shifted train paths. On the other hand, the number of continuous variables does not grow with additional time-shifted train-path options, because the continuous delay variables track the delay per primary delay event and occupation segment, forming an envelope over all timings as described in Section 4.4.

It is more difficult to assign direct meaning to the characteristics of the pre-processed model, as the commercial solver may apply different approaches and heuristics per model. Nevertheless, it is obvious that the size of the model can be reduced substantially across all indicators through these measures. This is "by design", as the fine-granular, track-section based representation of the developed model aims to capture the problem structure, but also likely contains a lot of duplicate information. For instance, many of the feasibility cliques in (12) are not maximal and thus dominated by other clique constraints; and neighboring track sections may often imply similar feasibility and delay propagation constraints when the train paths do not change drastically from one section to the next. Instead of trying to detect this manually, the task of filtering information is left to the solver.

Comparing the three station instances, we see that the trend for the model sizes is congruent with the number of possible train paths that are available to the model. Among the test set, the time-shifted instances for Arnhem appear most complex, though there appears to be no exponential growth in model size. While the size of a

MILP by itself is no direct predictor for the complexity of solving the optimization problem, it certainly can serve as an indicator in this case as all optimization problems are of the same general type.

5.3 Results and comparison of robustness optimization approaches

After optimizing the models, the results are depicted in Table 5, where we include the following dimensions for analysis: First, the summed delay of all outbound trains D_{out} leaving the station, assuming the same primary delay distribution used during optimization, i.e., one of the train entries or platform departures is chosen as the primary event, then a delay is chosen according to a modified exponential distribution with scale parameter $\mu = 120\text{s}$, $\text{cap}=300\text{s}$ and a delay likelihood $p = 50\%$. We sample 1,000,000 random seeds, and report the mean as well as – for ease of comparison – the deviation from the mean of the best-found solution per scenario. Then, as validation, we report the modeled expected delay propagation (i.e., the objective value of `delay_exp`) and the difference to the sampled mean. Note that, for `buffer_inverse` scenarios, this is not the actually optimized objective function; here we evaluated the value in a post-processing step by inputting the final optimal solution into the delay propagation model. Next, analogously to the singly disturbed case, the following two columns show summed outbound delay if all primary events may be affected by delay at the same time, according to the above-mentioned distribution and sampling procedure. Finally, we report the solution time for solving the optimization problems.

Across all instances, we observe the consistent trend that the optimal solutions under the `delay_exp` objective have the lowest expected outbound delay. The comparison objective `buffer_inverse` leads to slight increases in delay propagation ranging between 0.59% in the scenario Breda-0 and 2.32% in the scenario Arnhem-15-or-25. This indicates the potential for some robustness gains through our methodology, which is not limited to specific instances. A larger gap is apparent compared to the public timetable currently in operation. This highlights the importance and potential of taking robustness into consideration during planning, though we note that there are likely a range of other considerations for timetable quality that were not considered, e.g., planning short passenger transfers on the same platform or grouping trains and platforms according to the direction or service type. While their inclusion in the model is plausible, the focus of this work lies on the conceptual modeling approach and the task of producing fully realistic timetables is thus left for future work.

Comparing the sampled outbound delays with the modeled outbound delays (i.e., the objective value of `delay_exp`), we can observe a very high agreement, validating our theoretical derivation and underlining that `delay_exp` does indeed minimize the expected outbound delay. In particular, as is made explicit in the column Modeled D_{out} Diff., the consistent trend is a slight overestimation of the outbound delay in the modeled optimization value compared to the sampling procedure. This is plausible, as the model uses a piecewise linear approximation of the real, convex objective function.

Turning to the analysis of the practically relevant case in which multiple delays occur at the same time, the ranking of the methods remains consistent, though the relative difference between the methods is slightly reduced across all cases. Inter-

Table 5 Evaluation of the optimal solutions obtained for different scenarios and objectives

Scenario			Sampled D_{out}		Modeled D_{out}		Sampled D_{out} (mult.)		Solution
Station	Shift	Objective	Mean	Δ Best	Val.	Diff.	Mean	Δ Best	time
	[s]		[s]		[s]	[s]	[s]		[s]
Breda	0	delay_exp	47.09	*	47.17	0.08	1994.94	*	10
		buffer_inverse	47.37	+0.59%	47.45	0.08	2000.45	+0.28%	1
		publicTimetable	50.36	+6.94%	50.44	0.08	2075.23	+4.02%	<1
	15	delay_exp	45.27	*	45.35	0.08	1947.35	*	101
		buffer_inverse	45.86	+1.30%	45.94	0.08	1958.22	+0.56%	4
		publicTimetable	47.93	+5.88%	48.01	0.08	2013.78	+3.41%	<1
	25	delay_exp	44.42	*	44.50	0.08	1924.23	*	155
		buffer_inverse	45.16	+1.67%	45.24	0.08	1939.92	+0.82%	6
		publicTimetable	46.77	+5.29%	46.86	0.09	1984.73	+3.14%	<1
	15 or 25	delay_exp	44.40	*	44.49	0.09	1923.31	*	238
		buffer_inverse	44.63	+0.52%	44.72	0.09	1928.31	+0.26%	19
		publicTimetable	46.75	+5.29%	46.85	0.10	1983.85	+3.15%	1
Amersfoort	0	delay_exp	55.87	*	56.03	0.16	2923.01	*	89
		buffer_inverse	56.33	+0.82%	56.49	0.16	2929.81	+0.23%	2
		publicTimetable	60.61	+8.48%	60.77	0.16	3065.64	+4.88%	<1
	15	delay_exp	52.86	*	53.01	0.15	2826.06	*	1939
		buffer_inverse	53.28	+0.79%	53.43	0.15	2845.92	+0.70%	45
		publicTimetable	57.14	+8.10%	57.31	0.17	2962.56	+4.83%	1
	25	delay_exp	51.44	*	51.60	0.16	2779.22	*	2139
		buffer_inverse	51.97	+1.03%	52.11	0.14	2804.77	+0.92%	62
		publicTimetable	55.54	+7.97%	55.72	0.18	2913.84	+4.84%	2
	15 or 25	delay_exp	51.43	*	51.58	0.15	2779.76	*	3773
		buffer_inverse	51.91	+0.93%	52.06	0.15	2799.18	+0.70%	130
		publicTimetable	55.49	+7.89%	55.67	0.18	2913.16	+4.80%	2
Arnhem	0	delay_exp	37.70	*	37.79	0.09	1755.42	*	1751
		buffer_inverse	38.18	+1.27%	38.26	0.08	1773.18	+1.01%	55
		publicTimetable	38.64	+2.49%	38.73	0.09	1787.37	+1.82%	<1
	15	delay_exp	36.36	*	36.44	0.08	1708.74	*	51185
		buffer_inverse	36.99	+1.73%	37.08	0.09	1737.30	+1.67%	3324
		publicTimetable	37.10	+2.04%	37.17	0.07	1738.91	+1.77%	<1
	25	delay_exp	35.81	*	35.88	0.07	1700.24	*	86400 (Gap: 4.44%)
		buffer_inverse	36.61	+2.23%	36.71	0.10	1725.85	+1.51%	19441
		publicTimetable	36.40	+1.65%	36.48	0.08	1716.80	+0.97%	2
	15 or 25	delay_exp	35.70	*	35.78	0.08	1686.48	*	86400 (Gap: 8.46%)
		buffer_inverse	36.53	+2.32%	36.62	0.09	1723.22	+2.18%	23244
		publicTimetable	36.39	+1.93%	36.46	0.07	1716.29	+1.77%	2

estingly, the level of difference between the single- and multi-disturbed case varies noticeably across the different scenarios, hinting towards some instance-specific effects. In terms of the overall reduced benefit, we give the following hypothesis: In the multi-disturbed case, the impact of different primary delays may overlap, partially canceling out, meaning that some reduced propagation times may not actually materialize as they are dominated by other effects. Nevertheless, the general trend of improved delay robustness carries over into this setting.

With respect to the computational performance, however, we note that the solution time is significantly slower for `delay_exp` compared to `buffer_inverse` across all scenarios due to the need of modelling more complex delay relationships. While this represents a trade-off for the achieved optimization gains, the context of long-term planning in which this method is employed allows for longer computation times. The solution times appear to scale with the general complexity of the stations (cf. Tables 3, 4). While the number of train-path options alone is not a sufficient predictor of the runtime – comparing, e.g., scenario Breda-15 with $3 \cdot 300 = 900$ options, which solves significantly faster than scenario Arnhem-0 with 656 options – the ranking of the scenarios according to their pre-processed model sizes (e.g., non-zero entries) is closely reflected in the ranking of their solution times.

In accordance with the model sizes, we also note that the solution times for scenarios with allowed time shifts of $\pm 15s$, $\pm 25s$ and both get progressively longer. We assume that the higher allowed shift results in fewer directly infeasible train path combinations (which can be efficiently detected by the feasibility constraints) and thus more options whose objective differs only due to the propagated delay. The cases Arnhem-25 and Arnhem-15-or-25 thus are the only ones not solvable to proven optimality within 24 hours, though the remaining optimality gap of 4.44% and 8.46%, respectively, are moderate. Their significantly longer solution times likely stem from the exponential complexity of solving MILPs of increasing size.

These results overall support our initial hypothesis that a more detailed modeling of the delay propagation mechanisms, including dwell time buffers as well as the indirect propagation of delay captured through absorption, can improve the delay robustness of the planned station routings. However, these inclusions must be weighed against an increase in model complexity that negatively affects the runtime of the optimization process.

Finally, we analyze the results for the scenarios with small allowed time shifts ($\pm 15s$, $\pm 25s$ or both, respectively) compared to the default timetable. The sampled outbound delay values reveal a notable positive effect on the local robustness of the station timetable. The magnitude of this effect is relatively consistent across different objectives within each station, but differs between the stations. This indicates that the associated gain depends strongly on the specific station or timetable structure.

Comparing the different allowed shifts, we see a larger delay reduction for 25s than for 15s, though this effect is not fully proportional to the time shift itself: For instance, for Amersfoort-`delay_exp`, an allowed time shift of 15s and 25s (ratio: 1.67) reduces the sampled mean outbound delay by 3.01s and 4.43s (ratio: 1.47), respectively. A possible reason could be a higher relative benefit obtained by separating particularly close trains even slightly. Nevertheless, the fact that the objective difference between “shift 25” and “shift 15-or-25” for `delay_exp` is almost negligible across all

stations may indicate that an extremal shift (within the analyzed order of magnitude) is typically preferred and more beneficial than a smaller one. Overall, these results indicate that allowing even small modifications to the timetable can be locally beneficial, though it may not be necessary to allow shift options at a highly granular level.

Contextualizing these experiments, the shifted scenarios represent a slight deviation from the general setting that the timetable is fully pre-determined by a network algorithm. Notably, as presented, this analysis does not account for the associated network effects inherent with modifying the timetable: If the timetable changes, the running time supplements to the next station must change as well, potentially modifying the primary delay distribution that a train arrives with or requiring subsequent changes in other stations. Additionally, some of the allowed time shifts may be undesirable within the station because they, e.g., make passenger transfers between trains infeasible. Nevertheless, the positive effects in these scenarios suggest interesting opportunities for a careful interplay between station and network planning to improve the overall robustness.

As a complement to this quantitative analysis on delay propagation results, we also compare the optimal routings obtained by `delay_exp` and `buffer_inverse` in Fig. 8. The top heatmap shows the percentage of identical assignments on a per-train basis between the solutions produced by these two objectives: In particular for Breda and Amersfoort, the platform assignments are identical for 71%-100% of the trains, while the agreement interestingly is lower for some scenarios in Arnhem. Furthermore, the second row analyses the more restrictive requirement that the exact route (which naturally includes the platform assignment) also matches. These values are only marginally lower than the platform agreement, indicating that if the same platform is chosen by both methods, then oftentimes an identical route to this platform is assigned as well.

The general trend for the comparison of the time shifts is similar: In the base cases, where all time shifts must be 0, there is by design perfect agreement and the correla-

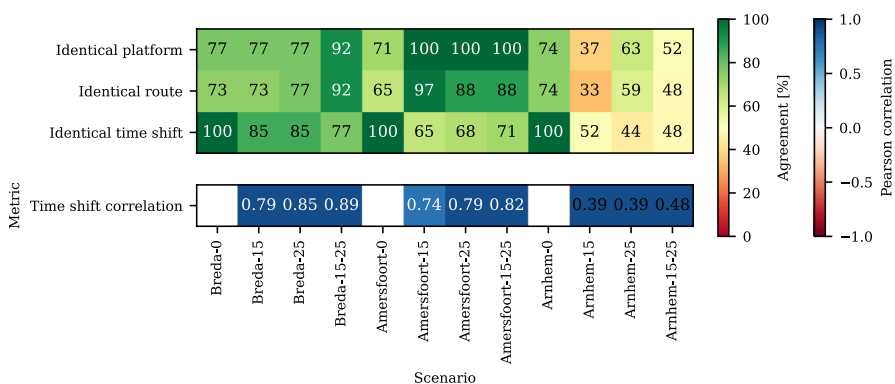


Fig. 8 Comparison between the optimal routings produced by `delay_exp` and `buffer_inverse` across different metrics, for each station and time shift scenario. The top heatmap shows for which percentage of trains the platform assignment, chosen route and chosen time shift are identical between the two objectives, respectively (left color bar). The bottom heatmap shows the correlation between the time shifts chosen by both objectives (right color bar)

tion is, therefore, undefined. For the remaining cases, the time shifts of both methods are positively correlated to varying degrees and exact agreement ranges from medium (44%) to high (85%) levels. Thus, while there is an apparent similarity between the solutions – which is unsurprising given the conceptual similarities between the buffer and absorption concepts – the different focus of the two objectives leads to a number of changes both in the timing of the trains and in the platform and route assignment.

The platforming component is further visualized for exemplary scenarios in Fig. 9: This figure plots the platform assignments in the solutions of *delay_exp* (x-axis) against *buffer_inverse* (y-axis), showing between which platforms the methods "exchange" trains. Cells on the main diagonal imply agreement between both methods. The cases in the top row of Fig. 9 correspond to the base versions of all three stations: There are some moderate shifts between the more central platforms 4 to 7 in Breda, and in Amersfoort-0 the only change between the methods is an exact flip of platforms 6 and 7 in the routing plan.

For Arnhem, we visualize all scenarios (top right and bottom row) to better understand where the comparatively larger differences seen in Fig. 8 stem from. Beyond some shifts to neighboring platforms, which were also present for the two other station instances, one can identify a number of trains whose assignments diverge more significantly, mainly between platforms 4a and platforms 10 or 11. Looking into the detailed solutions, this affects trains arriving from southwestern direction and traveling east through the station. For these, the algorithm has to decide whether or not to use the underpass leading to platforms 10 or 11, and the different objectives can lead to different results.

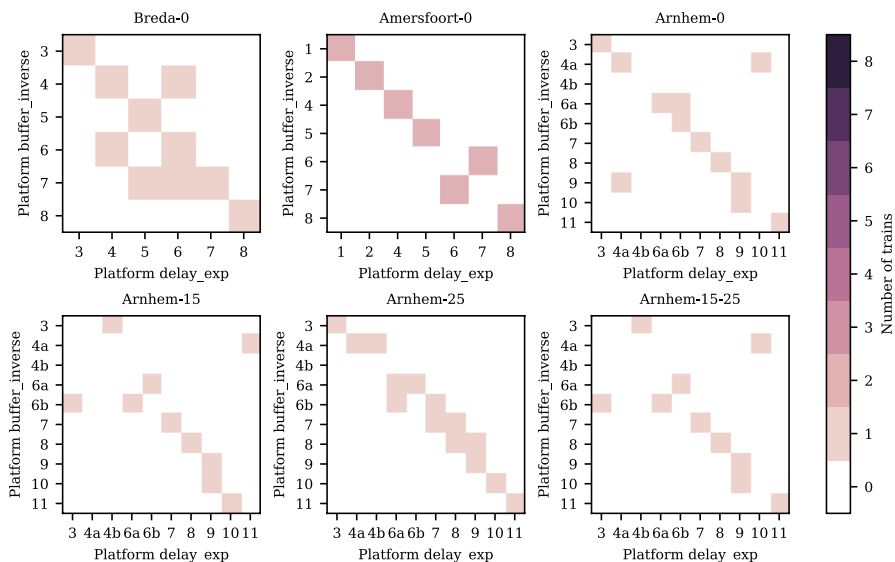


Fig. 9 Matrix of platform assignments in the optimal solutions produced by *delay_exp* and *buffer_inverse* for selected scenarios. Each cell shows the number of trains that are assigned to the platform by *delay_exp* (x-axis) and by *buffer_inverse* (y-axis). The main diagonal corresponds to agreement between the methods

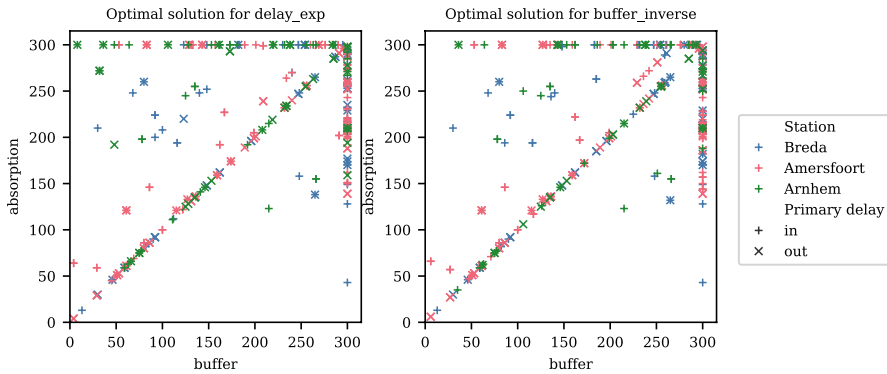


Fig. 10 Comparison of pairwise buffer and absorption times per dependent train pair in the optimal solution, across all stations for shift=0

5.4 Analysis of conceptual model differences

The different optimal routings generate distributions of buffers and absorptions between the trains, which we explore in Fig. 10 to gain a better understanding into the differences between the metrics. The solution of `delay_exp` is shown in the left, the one of `buffer_inverse` in the right plot. Each data point corresponds to a pair of primary and secondary delay events of different trains. As noted previously, the primary delay event can either be the entry into the station ("in") or the departure from the platform ("out") of the first train, while the secondary event is always the exit of the second train out of the station infrastructure.

The graphs show, on the x-axis, the direct buffer time between the two corresponding trains if they use the same infrastructure element somewhere on their path. On the y-axis, it shows the absorption time between the two events, which extends the direct buffer time by also taking dwell-time supplements and indirect delay propagation into account. As the objectives only take absorption or buffer values up to a limit of 300 seconds into account, a data point on this boundary should be interpreted as having at least 300 seconds of separation in the respective metric (with the true value being potentially higher or the trains being fully independent from each other). Pairs which are "independent" under this scheme (i.e. both absorption and buffer equaling 300 seconds) were omitted from the plot.

Within the graph, we see three immediately noticeable clusters: First, some points follow the identity diagonal, meaning that absorption and buffer time values match exactly. This matches cases where, in the absence of other factors, the direct buffer between two trains is indeed the dependency that causes delay propagation and the absorptions thus match the buffers exactly.

Second, there is a column of data points for buffer values equaling 300. This cluster corresponds to the cases motivating our absorption concept: The two trains never physically occupy the same segment (within 300 seconds), but there are indirect dependencies that can, nevertheless, lead to delay passing from one train to the other. Implicitly, this makes the contributing direct dependencies between the trains more impactful in an absorption-based view. It is noticeable that the main accumulation of

these cases occurs for absorption values exceeding roughly 120 seconds, while very small absorption values are largely absent. This is unsurprising, as each absorption in this cluster must be the result of at least two train dependencies, whose buffer time is summed. Furthermore, it is worth noticing that apart from this column there are few data points below the diagonal with buffer values smaller than 300 seconds. This indicates that there are few instances in which an indirect dependency supersedes the direct buffer if such a direct buffer exists.

Third, a line with absorption equaling 300 seconds is apparent, though there are also numerous other data points above the diagonal but below the absorption boundary. These points represent cases in which the shorter direct buffer time between trains is increased by additional dwell-time supplements. Thus, while the trains may affect each other within the station, there is also opportunity for them to immediately compensate the arising delay without influencing the wider network outside of the station. In an absorption-based analysis, this makes these cases less problematic than the direct buffer time would suggest.

Overall, on the level of individual train pairs, we observe numerous and significant deviations from the diagonal, indicating that there are relevant conceptual differences between absorption and buffer times, despite both objectives producing solutions of similar quality in terms of expected delay propagation. This opens the question if the solutions would differ more starkly under less restrictive constraints given by the network timetable.

Moving to a comparison between the two plots, there is a high similarity, which is consistent with the small differences in solution values between the different metrics. In details, however, some differences are observable: For Arnhem, the left solution produces three pairs with very low buffer, but very high absorption times. In comparison, only one of these pairs remains in the solution on the right. Conversely, the left solution shows a slightly reduced number and slightly increased absorption value over pairs with small absorption values up to 50 seconds (bottom left corner). These observations match the different optimization directions of the two metrics: `delay_exp` must attempt to shift points on the whole "up" and `buffer_inverse` to the "right".

5.5 Stability and sensitivity analysis against delay distribution parameters

5.5.1 Evaluation under differing delay distributions

The analysis in Section 5.3 targeted the impact of isolated primary delays which are distributed according to the same distribution that was used in the optimization objective. This raises two questions for the practical application of the method: First, any delay distributions assumed for a future tentative timetable must naturally be inexact. Thus, it is necessary to evaluate whether the comparative routing quality stays consistent even under different delay distributions which may not match the original optimization objective. Second, everyday operation typically suffers from the combination of multiple primary delays. To see practical benefits, the results from the isolated optimization need to carry over to this use case, as well.

To analyze these two questions, we once again sample delay scenarios and show the average delay propagation per objective and test instance in Fig. 11. In the left

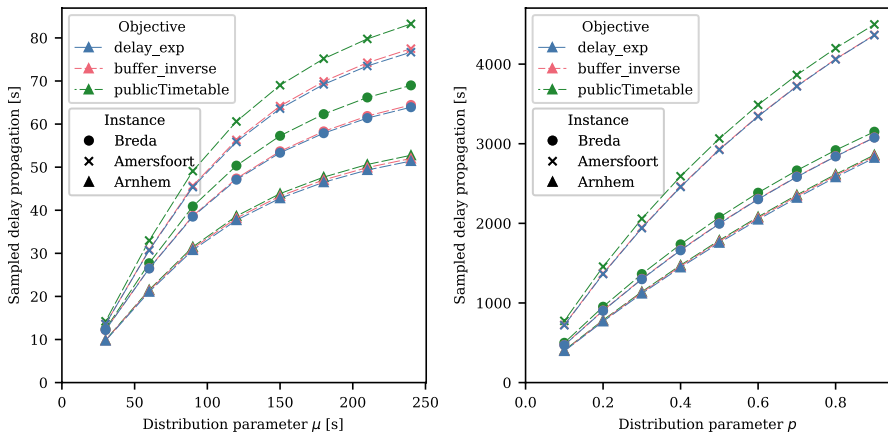


Fig. 11 Effect of variations in the sampled delay distribution parameters under the fixed, optimal routings. Left: Variation of the distribution parameter μ influencing the average delay of delayed trains for isolated primary delays (for fixed $p = 50\%$). Right: Cumulative effect of multiple delays for different train delay probabilities p (for fixed $\mu = 120s$)

plot, we vary the distribution parameter μ (which for the uncapped exponential distribution would be equal to the expected delay of the delayed trains), but keep the restriction to isolated delays. Naturally, increasing μ also increases the delay for all cases, though the increase lessens for higher values of μ . This is likely a side effect of the cap on the primary delays, which resamples primary delays above 300 seconds. Thus an increase of a small μ affects the expected primary delay in almost the same proportion, while for larger μ , the cap dampens the effect on the expected delay more prominently.

In the right plot, we keep $\mu = 120s$ fixed (the parameter value also used in the optimization objective), but analyze the cumulative impact of multiple primary delays. On the x-axis, we vary the probability of trains being delayed, simplified by assuming independence between the primary delays of multiple trains. We observe that the delay propagation increases with increasing delay likelihood, though also not fully linearly as one might expect at first glance. To interpret these findings, recall that our analyzed metric is the outbound overall delay of all trains. Thus, when multiple primary delays propagate to the same train (including the primary delay of the train itself), only the highest has an actual impact. For high delay likelihoods, the likelihood of such cases also increases and thus more delay is "canceled out".

Importantly in terms of answering the previously posed questions, however, the comparative performance between the routings stays consistent with only very small variation. This is true in both graphs, i.e., regardless of which delay parameter is varied. Thus, while we cannot directly make statements about the optimality in these settings, we conclude that the benefits seen for isolated primary delays empirically also have a positive impact when translated to a context with multiple primary delays or slightly varied delay distributions. It thus seems reasonable to expect that the routings found by the absorption methodology can reduce the delay propagation also under practical circumstances.

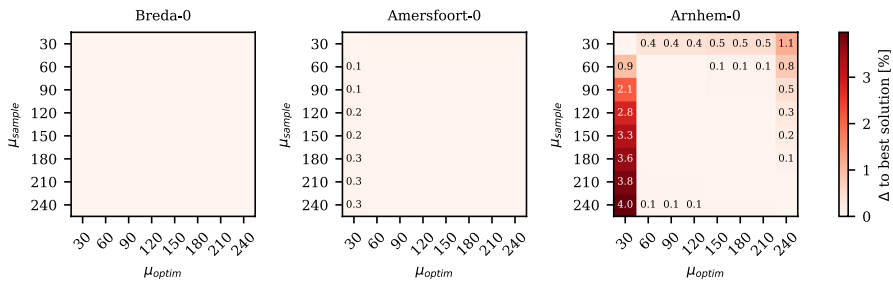


Fig. 12 Comparative analysis on the solution quality of delay_exp for isolated primary delays if the parameter μ_{optim} used for optimization differs from the μ_{sample} of the actually encountered delay distribution. The depicted value represents the delta to the best among this set of optimal solutions (i.e., per row), with empty cells corresponding to 0. For instance, the cell in the second row, first column for Arnhem indicates: If the sampled delay distribution uses $\mu_{sample} = 60\text{s}$, then the solution obtained from the model optimized for $\mu_{optim} = 30\text{s}$ causes expected secondary delay propagation that is 0.9% higher than the best models in this row (which are those optimized for $\mu_{optim} = 60\text{s}$, 90s or 120s)

5.5.2 Stability of solution quality for different delay parameters during optimization

An important, connected question is the impact of the delay distribution parameters in the optimization model on the obtained optimal solution of our objective delay_exp . For the likelihood p of delays occurring, this can be answered theoretically: As the optimization model only considers isolated delays, the delay likelihood p only acts as a constant factor in the objective for the occurrence of this isolated delay, and thus only scales the optimal value without affecting the objective characteristics or solution.

The parameter μ , on the other hand, changes the shape of the objective function by shifting the relative prevalence and importance of smaller or larger delays. Thus, solutions which are optimized for different μ may result in different optimal solutions. To assess how important it is that the parameter used for optimization (denoted in the following by μ_{optim}) closely resembles the delay distribution encountered "in practice" (denoted by μ_{sample}), we optimize a family of models and compare their relative solution quality.

The results are shown in Fig. 12 for the base scenario across all three stations: Each column represents one optimal solution and its solution quality for isolated primary delays under different sampled delay distributions. Conversely, each row represents the performance of all optimal solutions under the same sampled delay distribution and the depicted values are the relative difference to the performance of the best optimal solution in this row. An empty cell represents a solution with relative difference 0, i.e., the best performing solution(s).

As expected, the main diagonal is 0 in all instances, as each solution is optimal when evaluated under the same delay distribution used during optimization. Furthermore, the solution quality for Breda and Amersfoort appears very stable, with almost no difference in the average secondary delays. For Arnhem, a clearer trend is visible: The solution obtained for $\mu_{optim} = 30\text{s}$ performs progressively worse for increasing μ_{sample} , up to a relative quality decrease of 4.0%.

slightly worse with increasing divergence between μ_{optim} and μ_{sample} , though the effect is much less pronounced.

These findings lead to the following conjectures: The impact of the parameter μ may depend on how flexible the problem instance is, as the larger number of possible train paths for Arnhem gives more possibilities for the solutions to change under slightly different objective functions. Furthermore, while a slight divergence from the true delay distribution does not lead to drastically reduced solution quality, appropriate delay parameters (or even different delay distributions than the exponential distribution) can and should be used if there is statistical evidence to support them. This may be a worthwhile direction to incorporate statistical data on the delay of certain train types or train relations into planning processes.

Interestingly, the results suggest that underestimating the true delay distribution is more harmful than overestimating it. This may be caused by the skewed nature of the exponential distribution: If μ_{optim} is large, then the optimal solution represents a compromise between realizations with large and small primary delays, as both occur with relevant probability. In comparison, if μ_{optim} is small, then the likelihood of higher delays decreases to a point where they no longer meaningfully affect the objective. Thus, the optimal solution becomes exceedingly robust against very small delays at the cost of handling larger delays.

6 Conclusion

Train stations are often the neuralgic points for delay propagation within railway networks. In this work, we presented a methodology for the train routing problem in stations that directly models and optimizes expected secondary delays, taking both direct and indirect dependencies between trains as well as dwell-time supplements into account. The concept of absorption times between trains was introduced to relate the underlying delay propagation model to a stability assessment of station routings and to explore conceptual similarities and differences to buffer-based methods. The theoretical model was translated into a MILP model utilizing a fine-grained path representation and a new objective was proposed to minimize the expected delay propagation caused by isolated primary delays.

In a test on the three Dutch stations Breda, Arnhem and Amersfoort, we found that our methodology consistently yields slight robustness improvements compared to a buffer-based method. In an empirical analysis of the sensitivity of the results to delay input parameters, it was seen that these benefits also seem to extend to the practically relevant case of multiple primary delays and deviations in the primary delay distribution. However, the more complex nature of the model does require longer computation times, which becomes more apparent in instances with a high number of options per train. To this end, further work can focus on the solution technique and pre-processing measures to improve the efficiency of the solution process and to tackle even larger instances.

Even then, it is important to consider the appropriate level of complexity for the task at hand. In concrete tactical planning and when targeting stations known to be operated close to their limit, aiming for small improvements through more complex

planning methods may prove advantageous and justified. In comparison, particularly in long-term, tentative planning scenarios, the quality provided by buffer-based methods, which proved quite competitive, may already be sufficient.

This is also an important consideration for a more interconnected treatment of network and station planning – where station optimization would have to be applied at scale –, which appears promising from our results. While our main approach assumed fixed arrival and departure times, we showed that allowing even small shifts from the pre-determined network timetable can lead to relevant improvements in timetable robustness within the local station context. Verifying these findings under a more network-wide view of delay effects as well as identifying possibilities for an interaction between network and station timetable optimization are promising avenues for further research.

Finally, it will also be interesting to explore the interplay between infrastructure and routing more closely. With an increasing focus on integrated clock-face timetables, we expect strategic infrastructure development to be more closely targeted at enabling desired timetable concepts. This naturally leads to the question which timetables and routings can be realized on the infrastructure at a high quality and where infrastructure development offers the highest benefit, enabling strategic decisions that can contribute to a more punctual railway operation.

Acknowledgements The authors thank ProRail B.V. and N.V. Nederlandse Spoorwegen for providing us with infrastructure and timetable data of the considered Dutch railway stations.

Funding Open Access funding enabled and organized by Projekt DEAL. This research received funding by the European Union. Views and opinions expressed are, however, those of the authors only and do not necessarily reflect those of the European Union or the Europe's Rail Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them. The project “FP1-MOTIONAL” is supported by the Europe's Rail Joint Undertaking and its members.

Data availability The authors do not have permission to share data.

Declarations

Competing interests The authors have no competing interests to declare that are relevant to the content of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Bešinović N (2020) Resilience in railway transport systems: a literature review and research agenda. *Transp Rev* 40(4):457–478. <https://doi.org/10.1080/01441647.2020.1728419>
- Bešinović N, Goverde RMP (2019) Stable and robust train routing in station areas with balanced infrastructure capacity occupation. *Public Transport* 11(2):211–236. <https://doi.org/10.1007/s12469-019-00202-3>
- Bešinović N, Goverde RMP, Quaglietta E, Roberti R (2016) An integrated micro-macro approach to robust railway timetabling. *Transportation Research Part B Methodological* 87:14–32. <https://doi.org/10.1016/j.trb.2016.02.004>
- Burggraefe S, Vansteenwegen P (2017) Robust routing and timetabling in complex railway stations. *Transportation Research Part B Methodological* 101:228–244. <https://doi.org/10.1016/j.trb.2017.04.007>
- Caimi G, Chudak F, Fuchsberger M, Laumanns M (2011) A new resource-constrained multicommodity flow model for conflict-free train routing and scheduling. *Transp Sci* 45(2):212–227. <https://doi.org/10.1287/trsc.1100.0349>
- Caprara A (2015) Timetabling and assignment problems in railway planning and integer multicommodity flow. *Networks* 66(1):1–10. <https://doi.org/10.1002/net.21611>
- Caprara A, Galli L, Toth P (2011) Solution of the train platforming problem. *Transp Sci* 45(2):246–257. <https://doi.org/10.1287/trsc.1100.0366>
- Caprara A, Galli L, Stiller S, Toth P (2014) Delay-robust event scheduling. *Oper Res* 62(2):274–283. <https://doi.org/10.1287/opre.2014.1259>
- Caprara A, Galli L, Kroon L, Maróti G, Toth P (2010) Robust Train Routing and Online Re-scheduling. In: *10th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS'10)* T Erlebach, M Lübbecke (eds.), vol. 14 Open Access Series in Informatics (OASiCs) Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany:24–33 <https://doi.org/10.4230/OASiCs.ATMOS.2010.24>
- Cardillo DDL, Mione N (1998) k L-list λ colouring of graphs. *Eur J Oper Res* 106(1):160–164. [https://doi.org/10.1016/S0377-2217\(98\)00299-9](https://doi.org/10.1016/S0377-2217(98)00299-9)
- Delorme X, Gandibleux X, Rodriguez J (2009) Stability evaluation of a railway timetable at station level. *Eur J Oper Res* 195(3):780–790. <https://doi.org/10.1016/j.ejor.2007.06.062>
- Dewilde T, Sels P, Cattrysse D, Vansteenwegen P (2013) Robust railway station planning: an interaction between routing, timetabling and platforming. *Journal of Rail Transport Planning & Management* 3(3):68–77. <https://doi.org/10.1016/j.jrtpm.2013.11.002>
- Dewilde T, Sels P, Cattrysse D, Vansteenwegen P (2014) Improving the robustness in railway station areas. *Eur J Oper Res* 235(1):276–286. <https://doi.org/10.1016/j.ejor.2013.10.062>
- Duarte PJC, Dauzère-Pérès S, Huisman D, Mannino C, Sartor G, Weik N, Widmann P (2025) 50 years of OR in railway timetabling and rolling stock planning. *EURO Journal on Transportation and Logistics* 14:100155. <https://doi.org/10.1016/j.ejtl.2025.100155>
- Ge L, Voß S, Xie L (2022) Robustness and disturbances in public transport. *Public Transport* 14(1):191–261. <https://doi.org/10.1007/s12469-022-00301-8>
- Goverde RMP, Hansen IA, Hooghiemstra G, Lopuhaä H (2001) Delay distributions in railway stations. *9th World Conference on Transport Research, Seoul, Korea, July 22–27* (2001)
- Goverde RMP (2007) Railway timetable stability analysis using max-plus system theory. *Transportation Research Part B Methodological* 41(2):179–201. <https://doi.org/10.1016/j.trb.2006.02.003>
- Gurobi Optimization, LLC (2024) *Gurobi Optimizer Reference Manual*
- Kroon LG, Edwin Romeijn H, Zwaneveld PJ (1997) Routing trains through railway stations: complexity issues. *Eur J Oper Res* 98(3):485–498. [https://doi.org/10.1016/S0377-2217\(95\)00342-8](https://doi.org/10.1016/S0377-2217(95)00342-8)
- Lamorgese L, Mannino C, Natvig E (2017) An exact micro-macro approach to cyclic and non-cyclic train timetabling. *Omega* 72:59–70. <https://doi.org/10.1016/j.omega.2016.11.004>
- Leutwiler F, Corman F (2023) A review of principles and methods to decompose large-scale railway scheduling problems. *EURO Journal on Transportation and Logistics* 12:100107. <https://doi.org/10.1016/j.ejtl.2023.100107>
- Liebchen C, Stiller S (2009) The concept of recoverable robustness, linear programming recovery, and railway application. *Robust and Online Large-Scale Optimization*. Springer, Berlin, pp. 1–27. https://doi.org/10.1007/978-3-642-05465-5_1
- Lusby RM, Larsen J, Ehrgott M, Ryan D (2011) Railway track allocation: models and methods. *OR Spectrum* 33(4):843–883. <https://doi.org/10.1007/s00291-009-0189-0>

- Lusby RM, Larsen J, Ryan D, Ehrgott M (2011) Routing trains through railway junctions: A new set-packing approach. *Transp Sci* 45(2):228–245. <https://doi.org/10.1287/trsc.1100.0362>
- Lusby RM, Larsen J, Bull S (2018) A survey on robustness in railway planning. *Eur J Oper Res* 266(1):1–15. <https://doi.org/10.1016/j.ejor.2017.07.044>
- Pachl J (2018) Railway operation and control, 4th edn. VTD Rail Publishing, Mountlake Terrace
- Pellegrini P, Marliere G, Rodriguez J (2014) Optimal train routing and scheduling for managing traffic perturbations in complex junctions. *Transportation Research Part B Methodological* 59:58–80. <https://doi.org/10.1016/j.trb.2013.10.013>
- Pellegrini P, Marliere G, Pesenti R, Rodriguez J (2015) RECIFE-MILP: an effective MILP-based heuristic for the real-time railway traffic management problem. *IEEE Trans Intell Transp Syst* 16(5):2609–2619. <https://doi.org/10.1109/TITS.2015.2414294>
- Sartor G, Mannino C, Nygreen T, Bach L (2023) A MILP model for quasi-periodic strategic train timetabling. *Omega* 116:102798. <https://doi.org/10.1016/j.omega.2022.102798>
- Schwanhäuffer W (1974) Die Bemessung der Pufferzeiten im Fahrplangefüge der Eisenbahn. PhD thesis, Verkehrswissenschaftliches Institut der Rheinisch-Westfälischen Technischen Hochschule Aachen
- Sels P (2016) Large-scale, passenger oriented, cyclic railway timetabling and station platforming and routing. PhD thesis, Leuven: KU Leuven
- Sels P, Vansteenwegen P, Dewilde T, Catrysse D, Waquet B, Joubert A (2014) The train platforming problem: the infrastructure management company perspective. *Transportation Research Part B Methodological* 61:55–72. <https://doi.org/10.1016/j.trb.2014.01.004>
- van Hoeck I, Vansteenwegen P (2024) A MILP model to improve the robustness of a railway timetable by retiming and rerouting in a complex bottleneck area. *Journal of Rail Transport Planning & Management* 32:100488. <https://doi.org/10.1016/j.jrtpm.2024.100488>
- Weik N, Niebel N, Nießen N (2016) Capacity analysis of railway lines in Germany- a rigorous discussion of the queuing based approach. *Journal of Rail Transport Planning & Management* 6(2):99–115. <https://doi.org/10.1016/j.jrtpm.2016.06.001>
- Yang Y, Huang P, Peng Q, Li J, Wen C (2019) Statistical delay distribution analysis on high-speed railway trains. *Journal of Modern Transportation* 27(3):188–197. <https://doi.org/10.1007/s40534-019-0188-z>
- Zeng Q, Zhang Y, Su H, Hu R (2022) Robustness-oriented optimization of train platforming problem in busy complex railway station. *Math Probl Eng* 2022:1–10. <https://doi.org/10.1155/2022/5922653>
- Zwaneveld PJ, Kroon LG, Romeijn HE, Salomon M, Dauzère-Pérès S, van Hoesel SPM, Ambergen HW (1996) Routing trains through railway stations: model formulation and algorithms. *Transp Sci* 30(3):181–194. <https://doi.org/10.1287/trsc.30.3.181>
- Zwaneveld PJ, Kroon LG, van Hoesel SP (2001) Routing trains through a railway station based on a node packing model. *Eur J Oper Res* 128(1):14–33. [https://doi.org/10.1016/S0377-2217\(00\)00087-4](https://doi.org/10.1016/S0377-2217(00)00087-4)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.