Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

Improving the scheduling of railway maintenance projects by minimizing passenger delays subject to event requests of railway operators

Y.R. de Weert [a], K. Gkiotsalitis [b,*], E.C. van Berkum [c]

[a] ProRail, Capacity Management, P.O. Box 2038, 3500 GA Utrecht, The Netherlands
[b] National Technical University of Athens, Railways and Transport Laboratory, Dept of Transportation Planning and Engineering, Iroon Polytechniou 5, 15773 Athens, Greece
[c] University of Twente, Transport Engineering and Management, Dept of Civil Engineering, P.O. Box 217, 7500 AE Enschede, The Netherlands

ARTICLE INFO

Keywords:
Railway
Maintenance scheduling
Events
Passenger hindrance

ABSTRACT

In the Netherlands, it is expected that passenger activities on railway networks will double by 2050. To manage the passenger demand, railway capacity planning needs to be adapted. One fundamental aspect of the railway capacity planning is the scheduling of large maintenance projects. These maintenance requests should not be scheduled during major events to avoid the disruption of service. To do so, passenger operators can submit event request, i.e., a time period and location in which no maintenance project should be scheduled. Currently, these event requests are considered as hard constraints and the flexibility in maintenance scheduling decreases resulting in conflicts. In this study, the focus is on scheduling maintenance projects to minimize passenger delays while relaxing the hard constraints of event requests. This problem is addressed by introducing a Mixed Integer Linear Program (MILP) that minimizes passenger delays while scheduling maintenance projects, which includes capacity constraints for alternative services in event request areas during maintenance projects. The computational complexity of the proposed model is reduced by adding valid inequalities from the Single Machine Scheduling problem and using a simulated annealing meta-heuristic to find a favorable initial solution guess. Then, the MILP is solved to global optimality with Branch-and-Bound. A case study on the Dutch railway network shows improvements when event requests are not considered as hard constraints and an increase in the flexibility to schedule maintenance projects. This allows decision makers to choose from a set of optimal maintenance schedules with different characteristics.

1. Introduction

Railway infrastructure plays an important role in our society. It is important for the transportation of people and goods. Railway is one of the most energy efficient travel modes and could therefore play a key role in the energy transition towards a zero-carbon energy sector, according to IEA (2019). It is expected that passenger and freight activities on the railway network will double by 2050 given the current trends. This can only be achieved if the railway network is adapted to this expected increase in railway activities (Schasfoort et al., 2020; Trommelen et al., 2022). Therefore, the need for additional capacity in railway networks is rising.

Krueger (1999) defined the capacity of a railway line as "a measure of the ability to move a specific amount of traffic over a defined rail line with a given set of resources under a specific service plan". Therefore, one fundamental aspect of the capacity is the scheduling of maintenance works. This is because maintenance works on tracks cause the track segments to be unavailable for railway traffic, but are essential as these works are needed to maintain a functional railway network (Lidén, 2015).

Maintenance works can be divided into two categories: small routine works and projects (Budai et al., 2006; Gkonou et al., 2023). The first category concerns inspections and small reparations, such as inspection of rails, switch, level crossing, overhead wire, signaling system and switch lubrication. These type of works are often scheduled during nightly train-free periods. For example, van Zante-de Fokkert et al. (2007), Nijland et al. (2021), Buurman et al. (2023) conducted studies that concern optimization of scheduling small maintenance routine works in the nightly train-free periods.

The second category concerns maintenance projects that take more time, varying from a day to multiple days and include renewal works or bigger maintenance work that is less frequent (once or twice every few years). Examples of this category are ballast cleaning and tamping. Because the projects of the second category cannot be scheduled only during nightly train-free periods, train traffic is canceled or diverted

[*] Corresponding author.
E-mail addresses: Yoran.deWeert@prorail.nl (Y.R. de Weert), kgkiotsalitis@civil.ntua.gr (K. Gkiotsalitis), e.c.vanberkum@utwente.nl (E.C. van Berkum).

https://doi.org/10.1016/j.cor.2024.106580
Received 4 June 2023; Received in revised form 5 February 2024; Accepted 6 February 2024
Available online 8 February 2024
0305-0548/© 2024 Elsevier Ltd. All rights reserved.

due to these projects. This requires proper scheduling of the maintenance projects to minimize the impact of these projects on train traffic.

Particularly for the conveyance of people, maintenance projects affect the service quality of passenger operators. If trains are canceled due to these projects, alternative services are often provided to ensure that passengers can still continue their journey. This comes along with inconveniences for passengers as delay or additional transfers, and should be minimized to maintain a sufficient level of service.

Especially during events, which are for example soccer matches, concerts or festivals, maintenance projects should be scheduled carefully since events might have a disruptive effect on the rail network. These exemplary events attract visitors that often use public transport to reach their event venue, resulting in temporary high passenger demand. Without adjustments in the public transport schedule, these passenger demand peaks may cause overcrowded trains and stations (Robbins et al., 2007; Gkiotsalitis, 2023).

From the experience of the rail infrastructure manager, most issues arise from the event requests. In essence, there is a need to prevent overcrowding of stations or trains when events cause a temporary peak in the passenger demand. There is also a need to prevent situations where the event visitors are stuck at a station at night due to projects on their route. The passenger operator is responsible to indicate the events that will cause passenger demand peaks in some parts of the network as they will need additional trains for transporting event visitors. There are no costs bounded to an event request submitted by passenger operators, resulting in a large number of event requests. As the number of requests tends to become very large, it becomes virtually impossible to design a schedule where all demand for maintenance is met outside the periods of event requests. As a result, the best schedule should be found such that all demand for maintenance is met, even if that means that some event requests are violated. Given this context, the key is to find a maintenance schedule that minimizes the passenger delays.

In the Netherlands, the rail infrastructure manager (ProRail) introduced a system with event requests to prevent the scheduling of maintenance projects during extreme passenger demand peaks. Passenger operators can submit event requests, consisting of a set of track segments, such that no maintenance projects will be scheduled on these tracks during the event. ProRail tries, in favor of passengers and railway operators, to avoid scheduling maintenance activities on the specified track segments of the railway network that are indicated in the event request. Not being able to avoid the scheduling of the maintenance projects on these specified tracks results in a conflict between the operator and ProRail. At the moment, these conflicts are solved by iterative negotiations. This might lead to sub-optimal solutions for the passengers, leading to unnecessary increased travel times or detours, as ProRail does not have quantitative knowledge about passenger travel patterns from and towards events.

In this study, the aforementioned problem is addressed by constructing an optimal schedule for maintenance projects such that the passenger delays are minimized. The scheduling happens on a tactical level, meaning that only the date and time of the maintenance projects will be set. The schedule is published a year before it goes into effect, and any changes afterwards due to – for example – new information about events, are solved in an ad hoc manner (ProRail, 2022). The scope of this study is on the construction of the annual maintenance schedule. In particular, the focus is on the development of a method that optimally schedules maintenance projects to minimize passenger delays by considering the event requests of the passenger operator(s). This study extends the MSc thesis work of de Weert (2022). First, it refines a simulated annealing metaheuristic to find an improved initial solution guess to the problem. Second, it resolves the computational complexity and input size issues which were causing memory errors, allowing to solve larger problem instances. Third, it explains the use of Yen's kth shortest path algorithm to compute the k preferred routes of each origin–destination pair that are later used in the mathematical program. Finally, it includes experiments with additional datasets and a discussion concerning the relevant stakeholders.

The study is structured as follows. Section 2 provides a literature review on maintenance scheduling and the research goals. Section 3 contains the problem description and is followed by the methodology in Section 4. In Section 5 the computational times of the method are examined. Section 6 contains the results of a case study in The Netherlands. Section 7 concludes the study and offers future research directions.

2. Literature review

This section starts with a summary of the existing literature concerning maintenance scheduling. This is followed by the identification of research gaps and ends with the research questions and the contribution of the study to the topic of maintenance scheduling.

2.1. Railway maintenance scheduling

A categorization of different problems arising in railway maintenance coordination and scheduling is made by Lidén (2015). The problems are divided into strategical, tactical and operational problems. The first category contains problems concerning maintenance dimensioning, contract designs, maintenance resource dimensioning, and allocation. Possession scheduling, maintenance vehicle team and routing, and rescheduling are problems on a tactical level. The operational level contains maintenance project planning, working time and scheduling, and track usage planning. This study can therefore be classified as a problem on the tactical level of maintenance scheduling.

Higgins (1998) presents a mathematical model to allocate routine maintenance activities, maintenance projects and crew over a finite time horizon. In this model, the objective is to minimize the expected interference delay, i.e. delay due to unforeseen events which may occur to a train or activity, and the prioritized finishing time, which minimizes the time that track segments are below the required level of service. The mathematical model is solved using a tabu search metaheuristic since the problem size and complexity makes the computational burden too high. Cheung et al. (1999) proposes a method to assign maintenance jobs to track segments on a subway in Hong Kong. The system in Hong Kong works with job requests which should be scheduled in the five-hour period that the metro is not operational. Each job request has been given a certain priority and the goal is to maximize the number of jobs assigned from the requests without sacrificing higher priority jobs. They have a set of resource constraints that cannot be violated. The problem is solved using a resource allocation strategy based on constraint relaxation and the model was used to replace the manual job scheduling system.

Several studies have been conducted that address the preventive maintenance scheduling problem (PMSP) and variants of this problem. This problem concerns the scheduling of small routine works and bigger maintenance projects to minimize possession and maintenance costs considering one rail link. Budai et al. (2006) proposes several heuristics to solve this problem and a variant of this problem, which restricts the time periods between two consecutive executions of the same work to be exactly a pre-defined number of time periods. This problem is also addressed by Budai-Balke et al. (2009), extending the research by proposing more genetic and memoric metaheuristics for the PMSP. More recently, Macedo et al. (2017) solved the PSMP with resource constraints using a Variable Neighborhood Search (VNS) algorithm.

To improve the safety of rail-track workers, van Zante-de Fokkert et al. (2007) proposed a Mixed Integer Program (MIP) to construct a four-week schedule in which single track grids (STGs), which are sets of working zones on a railway corridor that can be blocked simultaneously, are closed to train traffic exactly once to perform small routine maintenance works. The MIP minimizes the number of nights and workload of the contractors, while safety is ensured by the definition of the STGs. Nijland et al. (2021) continues on the same problem by optimizing nightly maintenance schedules considering hindrance for train operators and the workload for track workers. Furthermore, this work divided small routine maintenance works in three components due to differences in the hindrance. This work found exact solutions for small instances using a MILP formulation. For larger instances, a combination of an exact method and meta-heuristics was proposed.

Buurman et al. (2023) optimized the nightly maintenance schedules considering the hindrance for train operators and the flexibility for contractors. The work studied how often, when and where nightly maintenance slots need to be reserved. A multi-objective optimization model was proposed and solved using heuristics. The result was a set of Pareto optimal solutions, giving the decision maker multiple options to choose from based on his/her preferences between different objectives.

The work of de Jonge (2017) focused on the scheduling of maintenance projects in the Netherlands. A MILP was developed to solve the problem where the scheduling of maintenance projects was subject to a set of constraints in the corridor book of ProRail. With the model, the set of constraints were analyzed to improve maintenance project scheduling. The results showed that it was not possible to schedule all maintenance projects whilst also respecting all constraints from the corridor book. Furthermore, a variant of the model was used to schedule maintenance projects considering the weights of event requests. A weight was assigned to each event request and weekend to maximize the sum over the violated event requests.

Zhang et al. (2013) considered the deterioration process of track segments to develop a monthly maintenance schedule for small, routine maintenance works. They developed a model for the monthly workload minimizing the effect on train operators and reducing potential costs. Each month, the state of the track segments were monitored and this study proposed a method to create a maintenance schedule based on the information from the monitoring procedure. The problem was solved using a Genetic Algorithm approach. A maintenance scheduling problem proposed by Boland et al. (2014) concerns the maximum flow problem with flexible arc capacities (MaxTFFAO). Here, track segments have no capacity when maintenance is performed on the track segment and are therefore unavailable at these moments. Boland et al. (2014) proposed an integer programming formulation for the maximum total flow in a network with flexible arc outages. Here, the goal was to schedule all given jobs and maximize the total flow over the planning time horizon. The problem was NP-hard and, for practical purposes, they proposed the use of heuristics. This model was made for a coal export supply chain, and Boland et al. (2013) continued on this problem and added more problem-specific constraints. Boland et al. (2016) expanded their original MaxTFFAO model by adding a limit to the number of jobs in a time period.

One of the first studies that considers the integration of train timetables and maintenance disruptions was conducted by Albrecht et al. (2013). The goal was to minimize the delays on maintenance and trains. For practical reasons and the need for quick rescheduling solutions, they applied the Problem Space Search meta-heuristic for large instances to generate a timetable for both train movements and track maintenance. Forsgren et al. (2013) presented a MIP model optimizing train timetables and preventive maintenance simultaneously. In their model, they allowed trains to be moved in time, redirected or canceled. Maintenance activities may not be canceled, although they are allowed to be moved to a pre-defined time window.

Lidén and Joborn (2017) developed a mixed integer linear program (MILP) to find an optimal long term tactical plan for train timetabling and maintenance windows for routine maintenance activities minimizing the total train running time, deviation of preferred departure, route cost, maintenance cost and indirect (setup/overhead) cost. They were able to solve the proposed MILP optimally within one hour on a conventional computer. Lidén et al. (2018) extended the research by explicitly focusing on crew resource constraints, meaning that there was a maximum limit of working hours and a minimum limit of rest time between working days. Furthermore, they presented a MILP for this problem. Zhang et al. (2019) also integrated train timetabling and maintenance scheduling, but for high-speed railway corridors with sunset departure, sunset arrival trains (SDSA-trains). These trains run in the night while regular maintenance is performed in the scheduled maintenance windows. They use linearization techniques to develop a MILP for the problem. Meng et al. (2017) addressed the integration of train timetabling and maintenance time windows scheduling, but considered speed restrictions on trains. They solved their problem also optimally by solving a MILP developed for their problem. Additionally, Pour et al. (2018) proposed a hybrid constraint programming/mixed-integer programming framework for the preventive signaling maintenance crew scheduling problem. Pour et al. (2018) used a warm-start strategy when solving their mixed-integer linear program. Finally, Pour et al. (2019) proposed a constructive framework for the preventive signaling maintenance crew scheduling problem in the Danish railway system. Their approach focused on planning the preventive maintenance of railway signals in Denmark using a MIP formulation.

To summarize, maintenance scheduling topics can roughly be divided into four categories: Possession Scheduling (PS), scheduling of routine maintenance works in predefined time windows or cyclical scheduling of Routine Works (RW), studies on the characteristics of Time Windows (TW), and integration of maintenance scheduling with train timetabling (TT). Furthermore, objectives of the different studies can be categorized in maintenance efficiency (ME), hindrance on trains (HT) or passengers (HP), and maximization of the throughput in a network (TM). We note that in this study the passenger hindrance is defined as the total increase in travel time of passengers. Table 1 provides a summary of the reviewed studies, the corresponding topics, and the objectives they address.

2.2. Research gaps

Topics concerning maintenance scheduling were examined by many studies. Notably, studies in the past years included more and more limitations in their models. For example, van Zante-de Fokkert et al. (2007) introduced the concept of STGs and Nijland et al. (2021) continued on this work by distinguishing the type of maintenance activities. Resource or workforce constraints were used in Cheung et al. (1999). In addition, Macedo et al. (2017) included resource constraints in another context. Meng et al. (2017) included speed limitations. Nevertheless, there are still relatively few developments on possession scheduling.

Most studies examine a static network, i.e. a network in which arc characteristics do not change over time or other factors. Budai et al. (2006) focused on a combination of routine works and minimized possession and maintenance costs. The work of de Jonge (2017) maximized the number of possessions. Although possession scheduling has a significant impact on the quality of train timetables and on train traffic, these factors were barely included in these studies.

Some studies do focus on dynamic networks. Boland et al. (2013) considered dynamic arc capacities but did not deal with passenger flows. The study of Buurman et al. (2023) modeled train routes in a network, but passenger routes were never considered. Furthermore, there is relatively little attention to abnormal circumstances that could, together with a maintenance schedule, heavily increase passenger hindrance. Most objectives in the construction maintenance schedules are based on the minimization of possession time, train traffic hindrance or track segment quality. External factors, as, for example, events or holidays, are not considered in these studies. To the best of the authors' knowledge, the work of de Jonge (2017) is the only past study that also schedules maintenance projects while considering events in the optimization phase by minimizing the number of projects during an event. Passenger hindrance is, however, not considered in that study.

The aim of this study is to find an optimal maintenance schedule that minimizes passenger delays, while considering the impact of the

Table 1
Topic and objectives of the reviewed literature on maintenance scheduling. (RW = routine maintenance works, PS = Possession scheduling, TW = Time windows, TT = Train timetable), ME = Maintenance efficiency, HT = Hindrance for trains, HP = hindrance for passengers, TM = Throughput maximization).

Author	Paper topic				Model objective			
	RW	PS	TW	TT	ME	HT	HP	TM
Higgins (1998)	✓				✓	✓		
Cheung et al. (1999)	✓				✓			
Budai et al. (2006)	✓	✓			✓			
Budai-Balke et al. (2009)	✓	✓			✓			
van Zante-de Fokkert et al. (2007)	✓				✓			
Albrecht et al. (2013)		✓		✓		✓		
Forsgren et al. (2013)		✓		✓		✓		
Boland et al. (2013)		✓						✓
Boland et al. (2014)		✓						✓
Boland et al. (2016)		✓						✓
Macedo et al. (2017)	✓	✓			✓			
Zhang et al. (2013)	✓				✓	✓		
Zhang et al. (2019)			✓	✓	✓			
Meng et al. (2017)				✓		✓		
Lidén and Joborn (2017)	✓		✓	✓	✓	✓		
Lidén et al. (2018)	✓		✓	✓	✓	✓		
de Jonge (2017)		✓			✓			
Pour et al. (2018)	✓				✓			
Pour et al. (2019)	✓				✓			
Nijland et al. (2021)	✓				✓	✓		
Buurman et al. (2023)			✓			✓	✓	

event requests made by the operator on the flexibility of maintenance schedule. This is done by developing a model that optimizes the scheduling of pre-selected maintenance activities considering passenger delays and event requests. The aim is translated into the following research question: how can we schedule optimally maintenance projects when minimizing passenger delays, while considering event requests?

2.3. Contribution

The study contributes to the topic of maintenance scheduling by presenting a framework to determine the optimal maintenance schedule such that passenger delays are minimized. Its closest prior art is the work of Boland et al. (2013). It alters the model of Boland et al. (2013) by converting the maximum flow problem with one origin and destination to a shortest path problem with multiple origins and destinations. Furthermore, the arc characteristics are extended such that not only the availability is known, but also the travel time that corresponds to the availability. This leads to a model that is able to determine the effect of maintenance scheduling on passenger delays by examining and modeling passenger route choices.

The model that is developed is a MILP and it is solved with exact methods, showing that optimal solutions can be attained for relatively small to medium-sized instances. The application in this study is on the consideration of event requests, although other practical restrictions on maintenance scheduling could also be examined with small adaptations to this model.

3. Problem description

3.1. Model description

The nomenclature used in this study is presented in Table 2.

The proposed model to address the problem is an adaptation of the model of Boland et al. (2014). Consider a railway network over a finite, discrete time horizon $\mathcal{T} = \{1, \dots, T\}$ with a node set $\mathcal{N}$ representing the stations, and an arc set $\mathcal{A}$. An arc $a \in \mathcal{A}$ represents a direct train connection, meaning that there exists a train that travels directly between the stations at the ends of arc a. Additionally, each arc is characterized by an expected travel time by train $\omega_a^e \in \mathbb{R}$ and an expected travel time using alternative services $\omega_a^i \in \mathbb{R}$.

For each origin–destination pair $(o, d) \in \mathcal{N} \times \mathcal{N}$ in the network, there is a passenger demand ϕ_{od} willing to travel from origin $o \in \mathcal{N}$ to destination $d \in \mathcal{N}$. Furthermore, it is assumed that passengers consider k possible routes to travel between an origin and destination and always take the shortest path in time from the k considered paths from an origin o to a destination d. These paths are the shortest, the 2nd shortest, …, the kth shortest.

In addition to the above, given is a set of jobs $j \in \mathcal{J}$, in which a job j is characterized by a set of connected arcs in $\mathcal{A}$ and processing time $\pi_j \in \mathbb{N}$. Hereafter, jobs will represent the maintenance projects. For each job j, a start time has to be assigned to each job such that each job is finished before the end of the time horizon $\mathcal{T}$. Scheduling a job on an arc $a \in \mathcal{A}$ results in an increased travel time ω_a^i on that arc until the job is completed. This is because the railway tracks are unavailable in that time period and alternative services are used to transfer passengers between the adjacent stations of that arc. These alternative services always serve exactly the same stops as the train that is canceled.

It should be considered that not all combinations of jobs are permitted to be scheduled in the same time period. The corridor book of ProRail describes agreements and guidelines to plan train-free periods and indicates which combinations of train-free periods are not permitted. Hence, a set C is defined containing combinations $c = (a_1, a_2, \dots) \subset \mathcal{A}$ that are forbidden to be unavailable simultaneously.

Furthermore, event requests are represented as a set of track segments in the networks. In these event request areas, the capacity must be sufficient to satisfy the passenger demand at that time period. It is assumed that train capacity is sufficient under normal circumstances, but it is limited in case of maintenance due to the lowered capacity that alternative services can provide per hour.

In event request areas, there is an increased passenger demand mostly before and after the event. When demand peaks on that day due to the event, the capacity should be sufficient within the event request area to prevent overcrowding. It is assumed that this is always the case if trains are operational, but capacity is limited if there is maintenance and alternative services replace train traffic. This capacity limit is represented by Λ_{st} for all track segments $s \in \mathcal{A}$ and time periods $t \in \mathcal{T}$. Set $\mathcal{E}_t$ represents the track segments of all active event requests at time t. Note that a track segment $s \in \mathcal{E}_t$ affects multiple train connections $s \subset \mathcal{A}$, since the closure of a track segment causes all relevant train connections to be replaced by alternative services.

The objective of the model is to minimize the passenger delays. In this case, the passenger delays are defined as the total increase in travel time of passengers. Additional assumptions made for the proposed model are listed below:

Table 2
Nomenclature.

Sets	**Description**
$\mathcal{N}$	Set of stations
$\mathcal{A}$	Set of direct train connections
$\mathcal{A}_j$	Set of direct train connections for a maintenance job $j \in \mathcal{J}$
$\mathcal{J}$	Set of maintenance jobs
$\mathcal{J}_a$	Set of maintenance jobs on arc $a \in \mathcal{A}$
$\mathcal{R}_{od}$	Set of k routes to travel from origin o to destination d
$\mathcal{C}$	Set of arc combinations that may not be unavailable simultaneously for rail traffic
$\mathcal{T}$	Set of discrete time periods
$\mathcal{E}_t$	Set of track segments (each track segment is expressed as a set of arcs)that are in an event request in time period t
Parameters	**Description**
T	Length of discrete time horizon
ϕ_{odt}	Passenger demand traveling from origin o to destination d in time period t
β_{odt}	Share of daily passenger demand traveling in the peak momentin time period t between o and d
Ω_{od}	Average travel time between origin o and destination d
ω^e_a	Average travel time by train on arc a
ω^i_a	Average travel time by alternative services on arc a
π_j	Processing time in days of job j
τ	Minimum time interval in days between maintenance jobs on an arc $a \in A$
Λ_{st}	Capacity of alternative services in time period t for track segment $s \subset A$
k	Number of routes considered to travel from an origin o to a destination d
Variables	**Description**
y_{jt}	Binary variable indicating the starting time of job j
x_{at}	Binary variable indicating the availability of arc a in time period t
h^k_{odt}	Binary variable indicating if route option k is used by passengers travelingfrom origin o to destination d
w_{at}	Travel time traversing arc a in time period t
v_{odt}	Travel time from origin o to destination d in time period t.

- There are no precedence relations between jobs, i.e. it holds for all pairs of jobs $(j_1, j_2) \in J \times J$ that j_1 can be scheduled before j_2 or the other way around.
- The urgency of maintenance jobs is not considered, hence each job has an equal urgency.
- Jobs cannot be interrupted.
- Maintenance contractors do not have a preference on the scheduling of jobs.
- All passengers traveling between an (o, d)-pair at time t chose the same route, which is the path with the minimum travel time.
- During off-peak hours or outside the event requests areas, bus capacity will always be sufficient.

An integer programming formulation is formally defined for the problem, which will be described in the next section.

4. Mathematical model

4.1. Objective function

The goal of the model is to minimize the total passenger delays. Delays in this study are defined as the increase in travel time compared to the average travel time between an origin and a destination and is shown in Eq. (1):

$$\sum_{(o,d)\in N\times N} \sum_{t=1}^{T} \phi_{odt}(v_{odt} - \Omega_{od}) \tag{1}$$

Eq. (1) determines the increase in travel time for passengers traveling between an origin o to a destination d for a single time period t. To account for all passengers, the increase in travel time is multiplied by the passenger demand ϕ_{odt}. The total delay is the sum over all considered time periods.

4.2. Constraints

Constraints of the mathematical model can be divided in four categories: Passenger routing behavior, maintenance scheduling, event request restrictions, and further restrictions on scheduling which depend on the scenario where the model is applied. These are analyzed in detail in the following sub-sections.

4.2.1. Maintenance scheduling

A feasible solution of the mathematical model finds a schedule such that that all jobs are scheduled. The variable y_{jt} is a binary variable representing the starting time, $y_{jt} = 1$ if job j starts at time t and $y_{jt} = 0$ otherwise. It is assumed that the maintenance job can only start once and cannot be interrupted. Also, delay or other factors that might affect the job are not taken into account. The basic restriction is that the job should be started and be completed within the considered time horizon T. This is modeled by Eq. (2):

$$\sum_{t=0}^{T-\pi_j+1} y_{jt} = 1 \qquad (\forall j \in \mathcal{J}) \tag{2}$$

During the time that the job is processed, the job occupies tracks which are therefore unavailable for train traffic. Boland et al. (2013) formulated a constraint to model the availability of an arc and the constraint behaves similarly if applied in this model. The availability of a train connection a in time period t is modeled by the binary variable x_{at}, which takes the value one if a is available, and zero otherwise. The value of x_{at} can be determined by Eq. (3):

$$x_{at} + \sum_{t'=t-\pi_j+1}^{t} y_{jt'} \le 1 \qquad (\forall a \in A, t \in \mathcal{T}, j \in \mathcal{J}_a) \tag{3}$$

Finally, alternative services will replace the canceled train connections. This comes with an increase in the travel time on one of the arcs in the network. This behavior is shown in Eq. (4):

$$w_{at} = x_{at}\omega^e_a + (1 - x_{at})\omega^i_a \qquad (\forall a \in A, t \in \mathcal{T}) \tag{4}$$

The variables x_{at} are only bounded by Eq. (3). An optional constraint is added to the model in order to ensure correct arc travel times for free variables. This equals the full considered time period minus the sum of the processing times of the jobs on the concerned arc as in Eq. (5):

$$\sum_{t\in\mathcal{T}} x_{at} = |\mathcal{T}| - \sum_{j\in\mathcal{J}_a} \pi_j \qquad (\forall a \in \mathcal{A}) \tag{5}$$

4.2.2. Restrictions on maintenance scheduling

Restrictions on maintenance scheduling are summarized in the corridor book explained in Section 3. All restrictions can be expressed as sets of track segments that cannot be unavailable simultaneously, except for the minimum time interval between jobs on an arc. Eq. (6) therefore models the compliance to the restrictions:

$$\sum_{a\in c}(1-x_{at})\leq 1 \qquad (\forall c\in C_j, t\in\mathcal{T}, j\in\mathcal{J}) \tag{6}$$

Without loss of generality, it can be stated that jobs on a single arc in the network cannot overlap. This prevents that two jobs, at the same location, will be scheduled at the same time. The constraint to prevent that jobs cannot overlap is given by Eq. (7):

$$\sum_{j\in\mathcal{J}_a}\sum_{t'=t-\pi_j-\tau}^{T} y_{jt}\leq 1 \qquad \forall t\in\mathcal{T}, a\in\mathcal{A} \tag{7}$$

We hereby note that an arc can undergo several maintenance actions, i.e., be part of multiple projects.

4.2.3. Inclusion of event requests

For a track segments s included in an event request, the passenger flow in the peak hour of the considered time period may not exceed the capacity provided by alternative services. This capacity is virtually unlimited when a track segment is available for train traffic and is limited when a job is scheduled. This is due to the alternative services used, which cannot meet the same capacity offered under normal circumstances. Eq. (8) models this behavior:

$$\sum_{a\in s}\sum_{(o,d)\in\mathcal{N}\times\mathcal{N}}\sum_{i=1|a\in R_{odi}}^{k} h^i_{odt}\beta_{odt}\phi_{odt}\leq\Lambda_{st}+M\sum_{a\in s}x_{at} \qquad (\forall s\in\mathcal{E}_t, t\in\mathcal{T}) \tag{8}$$

The left-hand side of the equation indicates the flow over that segment by combining the flows of the different train connection that are using the concerned track segment. The first term of the right-hand-side of the equation is the capacity when alternative services replace the train connection. The second term is zero if a track segment is closed due to maintenance and practically unlimited if the track is available. We note that M is a large positive number, where $M>\sum_{odt}\phi_{odt}$. This ensures the difference in capacity between using trains or alternative services.

4.2.4. Passenger route choice and evaluation

In general, most passengers try to minimize their travel time and therefore often choose the shortest route. The shortest path in a network can be computed by, for example, the use of flow conservation constraints used in the integer programming formulation of Boland et al. (2013). This is a heavy computational burden and a reduction in the complexity can be made by considering a selection of k preferred routes for each (o,d)-pair as input for the model. For example, the algorithm of Yen (1971) can be used to find the k preferred routes of each (o,d)-pair. Then, it is possible to determine the shortest path from the selection of routes at a time period t by computing travel times for each route in the selection by a summation over the arc weights.

$$\sum_{i=1}^{k} h^i_{odt}=1 \qquad \forall(o,d)\in N\times N, t\in\mathcal{T} \tag{9}$$

$$v_{odt}\geq\sum_{a\in R_{od_k}} w_{at}-M(1-h^k_{odt}) \qquad \forall i\in[k],(o,d)\in N\times N, t\in[T] \tag{10}$$

$$v_{odt}\leq\sum_{a\in R_{od_k}} w_{at} \qquad \forall i\in[k],(o,d)\in N\times N, t\in[T] \tag{11}$$

The first constraint, Eq. (9), ensures that the passenger flow from origin o to destination d is served by exactly one of the k selected routes. This path is the shortest path in time and is modeled by Eqs. (10) and (11). Eq. (10) provides a lower bound for the travel time from an origin o to a destination d at time t, where the travel time should be at least as big as the travel time of the chosen route. An upper bound is provided by Eq. (11) as the travel time should be smaller or equal than all the travel time of the k considered routes at time t.

A complete overview of the scheduling model considering passenger hindrance and event requests is provided in Appendix A.

4.3. Solution space pruning

Because of its mixed-integer nature, the proposed mathematical program is hard-to-solve in large scale scenarios. To rectify this, three methods are considered that might help reducing the solution space. The first method analyses values of parameters to observe if there are some infeasible scheduling combinations that could already be discarded from the solution pool. The second method concerns the application of valid inequalities, which are inequalities that tighten the solution space without removing any feasible solution. Finally, a meta-heuristic is considered to find a reasonably good solution as a starting point (initial solution guess). This initial solution guess is used to accelerate the solution search of branch-and-bound methods that can return a globally optimal solution. The effectiveness of the latter two methods are tested in Section 5.

4.3.1. Parameter analysis

Analysis of parameter values could sometimes lead to a reduction of the solution space. In some cases, variables could take a value that will never be included in any feasible solution. We exclude such cases to reduce the computational costs of the model.

The first case concerns arcs that are not included in any of the jobs. These arcs are always available to be used by train traffic and therefore their status (available/unavailable) and their travel times on the concerned arc can be considered as fixed:

$$x_{at}=1 \qquad \forall a\in\{a\in A|J_a=\emptyset\},\forall t\in\mathcal{T} \tag{12}$$

$$w_{at}=\omega^e_{at} \qquad \forall a\in\{a\in A|J_a=\emptyset\},\forall t\in\mathcal{T} \tag{13}$$

Now, consider the k possible shortest routes to travel between any (o,d)-pair. Suppose that an arc a is included in an event request in time period t. If an arc a is contained in each of the k routes and $\beta\phi_{o,d,t}>\lambda_e$ for a time period t, then the track segment corresponding to arc a should be available, yielding an infeasible solution otherwise. This is incorporated in Eq. (14):

$$\begin{aligned} x_{at} &= 1 \quad \forall(a,t)\in\{(a,t)\in\mathcal{A}\times\mathcal{T}|\beta_{odt}\phi_{odt}>\Lambda_{st},\\ a &\in R_{odi}\forall i\in\{1,\dots,k\},\exists s\in\mathcal{E}_t \text{ s.t. } a\in s\}\end{aligned} \tag{14}$$

Note that if there exists an (o,d) pair that exceeds this threshold each time there is an event request at the track segment s, this would result in an infeasible solution.

4.3.2. Valid inequalities

A set of valid inequalities is also provided in Appendix B.

4.3.3. Initial solutions

Meta-heuristics are able to improve the computational times of mixed-integer problems by finding in a relatively short amount of time a good solution. However, the use of meta-heuristics does not guarantee the convergence to a globally optimal solution. In this study, we use a meta-heuristic as a starting point for generating an initial solution which is (hopefully) close to a globally optimal solution within a limited time. Then, we use the initial solution provided by the meta-heuristic as a starting point in a branch-and-bound algorithm which guarantees the finding of a globally optimal solution for mixed-integer linear programs. The computational time of the branch-and-bound algorithm can be considerably improved following this approach since the initial solution guess may cut off a large amount of solutions by providing an upper bound on the objective function value. We note that the approach of providing a feasible solution to the Branch and Bound algorithm, and thus offering a warm start, dates back to the 1990s (Gondzio, 1998).

The meta-heuristic used in this study is simulated annealing. Simulated annealing is applied to find the starting times S_j of all jobs $j\in J$. An initial solution S_0 is used as input in the simulated annealing algorithm and the control value c should determine at what rate deteriorating solutions are accepted. Each iteration, a neighboring solution $N(S)$ is examined and accepted if this solution improves the performance of the system compared to the current solution. If not, there is still a probability that the neighboring solution is accepted. Best performing solutions are always stored in case the stopping criterion is met when evaluating worse solutions. A pseudo-code for the simulated annealing algorithm applied to our problem is shown in Algorithm 1.

For the computational instances, the initial solution S_0 of the simulated annealing is obtained via a construction heuristic. Here, job scheduling is considered per arc. Since no restrictions are imposed, this always leads to a feasible solution. In the case study, a solution is randomly generated until the generation of a feasible solution S_0. This solution is generated such that it does not cause conflicts and can therefore be used as initial solution for all tested thresholds.

The neighbor generation of the simulated annealing algorithm is performed randomly, although it restricts jobs on the overlapping track segments to be scheduled simultaneously. Other violations of constraints are treated as a large penalty (relatively infinity) and a feasible solution is then guaranteed, as the initial solution has no violated constraints.

The original annealing scheme of simulated annealing proposed by Kirkpatrick et al. (1983) is used in this study:

$$T_{t+1} = cT_t \qquad t = 1, 2, \dots \tag{15}$$

The constant c typically lies between 0.8 and 0.99. In this study, the value 0.99 is used.

```
Data: $S_0$ (initial set of job starting times, $c_0$ (initial control value)
Result: $S$
$S := S_0$  $c := c_0$  while stopcriterion do
    for $l := 1$ to $L$ do
        $S_{new} = N(S)$  if $f(S_{new}) \le f(S)$ then
            $S \rightarrow S_{new}$
        else
            $S \rightarrow S_{new}$ with probability $\exp\left(\frac{f(S) - f(S_{new})}{c}\right)$
        end
    end
    Update c
end
```

Algorithm 1: Simulated annealing meta-heuristic applied to find an improved schedule of the starting times of maintenance jobs.

We note that the output S of the simulated annealing algorithm presented in Algorithm 1 is used as an initial solution guess to the branch-and-bound algorithm in order to improve the speed of finding a globally optimal solution.

4.4. Solution strategies

The model is implemented in Python 3.7.4 using Gurobi 9.1.0, a solver for optimization problems. A laptop with Intel®Core™ i5-7200U processor and 8 GB RAM is used to run the models. Gurobi solves MILPs based on a branch-and-bound algorithm that includes several other techniques that improve optimization, of which the biggest contributors are pre-solving methods, cutting planes, heuristics and parallelism.

In total two solution strategies will be tested:

- Solution strategy 1 represents the branch-and-bound algorithm of Gurobi with the only addition that the proposed simulated annealing metaheuristic is used to generate an initial solution guess and provide an upper bound for the branch-and-bound algorithm.
- Solution strategy 2 extends solution strategy 1 by including the valid inequalities proposed in this study and the following cutting plane generators: Boolean quadric polytope cuts (Padberg, 1989), Clique cuts (Padberg, 1973), Cover cuts (Weismantel, 1997), Flow cover cuts (Gu et al., 1999), Flow path cuts (van Roy and Wolsey, 1987), Gomori cuts (Balas et al., 1996), GUB cover cuts (Nemhauser and Vance, 1994), Implied bound cuts (Padberg, 2001), Lift-and-project cuts (Balas and Perregaard, 2002), MIR cuts (Günlük and Pochet, 2001), Mod-k cuts (Caprara et al., 2000), Network cuts (Balas, 1971), Relax-and-lift cuts (Bonami, 2011), Strong-CG cuts (Chvátal, 1973), $\{0, \frac{1}{2}\}$ cuts (Caprara and Fischetti, 1996).

Table 3
Cardinality of sets for each toy network.

ID	$\|\mathcal{N}\|$	$\|\mathcal{A}\|$	$\|\mathcal{J}\|$	$\|\mathcal{T}\|$	$\|\mathcal{E}\|$	$\|\mathcal{C}\|$	k
1	10	45	10	10	0	0	3
2	10	45	10	50	0	0	3
3	10	45	10	100	0	0	3
4	10	45	40	10	0	0	3
5	10	45	40	50	0	0	3
6	10	45	40	100	0	0	3
7	10	45	80	10	0	0	3
8	10	45	80	50	0	0	3
9	10	45	80	100	0	0	3
10	20	190	10	10	0	0	3
11	20	190	10	50	0	0	3
12	20	190	10	100	0	0	3
13	20	190	40	10	0	0	3
14	20	190	40	50	0	0	3
15	20	190	40	100	0	0	3
16	20	190	80	10	0	0	3
17	20	190	80	50	0	0	3
18	20	190	80	100	0	0	3
19	40	780	10	10	0	0	3
20	40	780	10	50	0	0	3
21	40	780	10	100	0	0	3
22	40	780	40	10	0	0	3
23	40	780	40	50	0	0	3
24	40	780	40	100	0	0	3
25	40	780	80	10	0	0	3
26	40	780	80	50	0	0	3
27	40	780	80	100	0	0	3

That is, solution strategy 2 employs a branch-and-bound algorithm with valid inequalities and cutting planes, whereas solution strategy 1 employs a branch-and-bound algorithm without valid inequalities. Both solution strategies use the solution of the simulated annealing meta-heuristic as an initial solution guess. These solution strategies will be later compared against the as-is branch-and-bound implementation of Gurobi which we call "Model 0" and does not use the simulated annealing meta-heuristic to obtain a favorable initial solution guess.

5. Computational study on benchmark toy instances

Toy instances are used to gain insights into the performance of the valid inequalities and the meta-heuristic. The characteristics of the toy instances are discussed first. Then, the computational results from the applications of the solution strategies to the toy instances follow. Finally, limitations of this computational study and further testing directions are presented.

5.1. Description of benchmark instances

A total of 27 benchmark instances have been generated to test the computational performance of the model. These instances are characterized by the size of the network, number of maintenance jobs, and the length of the time horizon. The nodes in the network are evenly spaced on a unit circle. The weight of the arcs in the network is the Euclidean distance. In case an arc is under maintenance, the weight is multiplied by a factor 1.35, meaning $\omega_a^i = 1.35\omega_a^e$ for all $a \in A$. Parameters related to passenger demand are also randomly generated. The remaining sets $\mathcal{C}$ and $\mathcal{E}$ are not included in the computational study.

Table 4
Computational results for Solution Strategy 1 tested on the toy instances.

Instance ID	MIP Gap (%)	Runtime	Runtime	Runtime	Nodes explored
		(total)	meta-heuristic	branch and bound	
1	0.00	16.67	13.70	2.44	517
2	0.00	77.29	65.26	9.46	25
3	0.00	172.63	131.99	35.48	100
4	27.48	7216.98	16.41	7200.01	1 368 164
5	3.48	7281.20	77.16	7200.74	153 436
6	125.52	7343.21	135.34	7202.08	85 213
7	564.36	7228.68	28.04	7200.02	1027013
8	150.80	7286.01	83.06	7200.04	139 374
9	200.46	7355.95	149.74	7200.23	85 621
10	0.00	113.09	79.06	31.45	1337
11	0.00	524.29	493.03	15.67	3
12	0.01	1032.19	837.12	168.75	32
13	0.00	96.40	73.22	20.85	539
14	19.04	7586.66	374.95	7200.10	54 381
15	23.33	7957.17	732.79	7200.22	19 463
16	2.74	7297.72	95.11	7200.03	219 454
17	113.36	7590.56	378.51	7200.09	45 059
18	68.73	8060.58	833.65	7200.11	10 896
19	0.00	594.26	561.32	18.69	3
20	0.00	2093.12	1806.32	213.68	3
21	0.00	2968.47	1811.03	1006.50	22
22	0.00	720.67	544.45	162.34	528
23	0.01	2445.58	1806.44	572.30	22
24	1.74	9172.22	1812.76	7203.36	1022
25	12.16	7780.92	566.62	7200.09	36 863
26	39.69	9084.62	1808.09	7201.34	2155
27	909.46	9176.80	1815.88	7202.66	138

The toy instances are structurally generated. For each of the included sets (network size, number of jobs and time horizon), three values were chosen resulting in a total of 27 unique combinations. The network size varies between 10, 20 and 40; the number of jobs is 10, 40 or 80 and the length of the time horizon is 10, 50 or 100. The intervals between the values are chosen to be relatively large such that differences between set sizes might be easier to observe.

An overview of the toy instances is shown in Table 3. The table presents the sizes of all sets.

5.2. Computational times for the toy instances

The impact of set sizes on computational times is analyzed in Tables 4 and 5. Each table shows for a specific solution strategy the computational times, the MIP gap and the number of explored nodes during the branch-and-bound algorithm as indicators of the performance of the model.

The computational times are subdivided into a running time of the meta-heuristic and the branch-and-bound algorithm. The total runtime is the total running time of the solution strategy, i.e. the meta-heuristic time, the branch-and-bound time and some remaining time to compute some small sets that are not stored in the dataset of the instance.

The MIP gap is a percentage indicating the objective function value differences between the solution of the relaxation of the model and the best found solution. The Gurobi solver declares global optimality if the integrality gap is smaller than a predefined threshold, which is set to the default setting of 0.01%. That is, solution with a MIP gap less than 0.01% are considered to be globally optimal.

In order to compare the performance of the solution strategies, the computational time threshold is set to 7200 s (2 h) for the branch and bound algorithm. The simulated annealing algorithm used in the solution strategies terminates if the time limit of 1800 s is reached or if the temperature reaches a value lower than 1. The initial temperature is set to 500,000.

Tables 4 and 5 present the results for Solution Strategy 1 and Solution Strategy 2, respectively. The first observation is that a solution has been found for all instances by both models, although the solutions are not necessarily optimal. This is a logical consequence of the computational time threshold of 2 h imposed to the branch-and-bound algorithm.

Solution strategies 1 and 2 achieve roughly the same results with respect to the optimality gap (MIP Gap). There is, however, one exception. Interestingly, the results on instance 18 are significantly different. Solution strategy 1 reached a gap of 68.73% and solution strategy 2 a gap of 513.60%. This difference could be caused by the initial meta-heuristic. Solution strategy 1 started with a worse initial solution for the MIP than solution strategy 2 due to the stochastic nature of heuristics and probably affected the performance of the branch-and-bound algorithm.

On the instances that are solved optimally, solution strategy 1 has a better computational time on almost all instances compared to solution strategy 2. This is surprising as one might have expected that the addition of tight valid inequalities and cutting planes can reduce computational times. On the other hand, adding more inequalities and cutting planes increases the computational burden because the model has significantly more constraints (becomes more complex). That is, the additional model complexity of incorporating valid inequalities and cutting planes seems to have outweigh the possible gains from the tightening of the solution space.

From the instances that are not solved optimally by both solution strategies, it can be seen that the number of nodes explored in the rooted tree of the branch-and-bound algorithm is significantly lower for solution strategy 2. The results on the MIP gap are roughly equal; hence, the iterations of solution strategy 2 are more effective. As previously described, this comes at a cost because a single iteration of solution strategy 2 takes more time due to the increased number of constraints.

Fig. 1 shows the MIP gap development for the two solution strategies when applied on toy instance 5. In this figure, we also present the MIP gap when using the Gurobi optimizer without using simulated annealing to derive an initial feasible solution guess. We call this application of Gurobi without the use of simulated annealing to derive an initial solution guess “Model 0”. Model 0 represents the as-is implementation of Gurobi to solve our MILP. For brevity, we will also call Solution Strategy 1 “Model 1” and Solution Strategy 2 “Model 2” in the remaining figures. Fig. 1 makes explicit that our solution strategies

Table 5
Computational results for Solution Strategy 2 tested on the toy instances.

Instance ID	MIP gap (%)	Runtime (total)	Runtime meta-heuristic	Runtime branch and bound	Nodes explored
1	0.00	17.35	16.20	0.51	3
2	0.00	91.12	74.78	12.76	26
3	0.00	219.80	137.76	72.41	155
4	35.11	7221.43	20.64	7200.02	1086659
5	3.89	7271.03	65.13	7200.07	107 196
6	127.98	7382.72	155.33	7200.11	19 947
7	583.99	7235.58	34.73	7200.02	850 028
8	150.53	7319.31	105.51	7200.13	27 461
9	185.31	7454.53	192.68	7200.41	2876
10	0.00	307.52	213.13	85.35	1348
11	0.00	450.62	422.88	14.05	3
12	0.01	1955.99	1549.93	333.63	32
13	0.00	90.93	82.31	5.95	48
14	19.03	7630.17	414.79	7200.14	46 717
15	23.81	8062.17	820.92	7200.19	9927
16	3.24	7446.41	241.47	7200.03	149 405
17	114.38	7650.51	432.36	7200.09	29 068
18	513.60	9794.44	1730.67	7983.47	2553
19	0.00	928.55	762.69	147.58	517
20	0.00	3028.50	1806.08	1127.79	517
21	0.00	2986.37	1814.20	973.86	22
22	0.00	794.29	738.32	37.66	21
23	0.01	2450.69	1805.03	577.81	22
24	1.81	9178.53	1815.19	7203.49	1022
25	12.52	7965.01	746.90	7200.09	28 879
26	39.70	9084.82	1805.72	7200.50	2241
27	912.85	9229.46	1811.04	7241.88	1

1 and 2 offer a significant improvement to the computational times compared Model 0 that does not apply a meta-heuristic to obtain an initial solution guess. This is evident because they converge much faster (within less than 300 s).

In more detail, after 140 s of running the branch-and-bound algorithm (which is 220 s including the meta-heuristic time) solution strategy 1 reached a gap of 5% and has a significantly better performance compared to Model 0 within this time period. Solution strategy 2 reached the gap of 5% after 276 s of running the branch-and-bound algorithm and also shows a significantly better performance compared to Model 0.

Generally, Fig. 1 shows characteristics that were observed in all other instances. For all cases that were solved optimally, the branch and bound algorithm for both solution strategy 1 and solution strategy 2 found the optimal solution within 1200 s. The MIP gaps initially start with a MIP gap value higher than 300% and then make a sudden drop at around 140 s of execution towards a value close to zero. After that, the optimal value is reached quickly. The remaining cases exhibit the same behavior in the development of the MIP gap, although the drops are generally later in time and reach a lower bound quickly after the drop. We have an exception only on instance 18, where solution strategy 1 shows that there is a drop in the MIP gap after 3200 s of execution from 512% to 68%, but the branch and bound in solution strategy 2 does not improve the initial solution and remains at the gap of 513.6%.

To conclude this section, the following observations are made based on the considered toy instances:

- The proposed meta-heuristic (simulated annealing) has a significant impact on the computational costs for the medium-sized instances.
- The valid inequalities and cutting planes in Solution Strategy 2 tighten the solution space. As a result, the branch-and-bound algorithm explores significantly less nodes to achieve the same results in terms of convergence. However, this does not result in better solutions within the considered time period because each iteration of solution strategy 2 takes considerably more time because of the added complexity due to the high number of additional constraints.

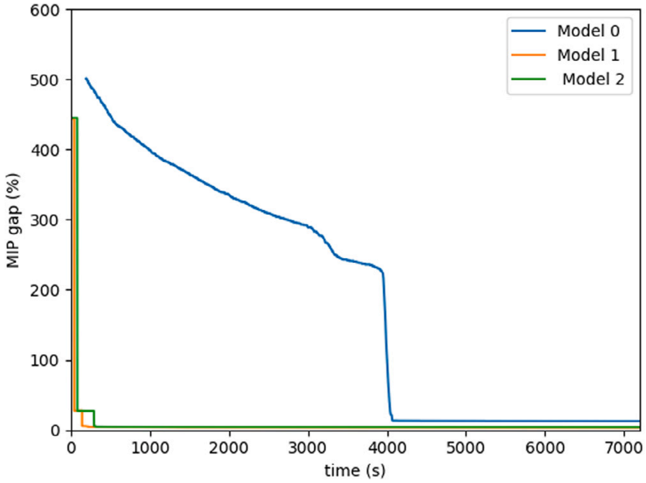

Fig. 1. MIP gap development for the as-is branch-and-bound algorithm of Gurobi (Model 0), and the Solution Strategies 1 and 2 on toy instance 5.

- Within 2 h, optimal solutions are only found for relatively small instances. This indicates that, in practical applications, one should apply sub-optimal solutions.

5.3. Limitations and further testing directions

The computational time limit was set to 7200 s and different time limits could reveal new information. For some instances, as for example instance 5, it can be seen that the optimality gap quickly decreases and only minor improvements are made over the remaining period. Other instances need more time and 7200 s of computational time do not suffice. Note that the allowed computational time can be extended since the scheduling is on a tactical level. Theoretically, computations can be allowed to run for many days.

Another interesting direction is to focus on a node limit instead of a time limit as stopping criterion of the branch-and-bound algorithm.

The time of processing one node in the branch-and-bound algorithm is a variable implying that within the considered two hours the number of explored nodes differs between solution strategies. This could give more information regarding the trade-off between the tightness of the solution space by adding more constraints and the computational time.

Note that the use of instances derived from complete graphs gives a quadratic relation between $|\mathcal{N}|$ and $|\mathcal{A}|$. Therefore, it cannot directly be stated that there is a significant difference on the impact of the model. In complete graphs, all options from the set of possible arcs are considered to construct paths and this implies that it is likely that a single arc is contained in relatively few paths. In sparse, connected graphs, single arcs might be included in more paths while the cardinality of $\mathcal{N}$ does not change and could result in a significant decrease of the number of variables. This is, however, a direction that should be further examined.

6. Case study

6.1. Case study area description

The Dutch rail infra manager ProRail publishes annually a schedule for the maintenance projects on the railway network. The construction of this maintenance schedule involves multiple stages and is started more than 2 years before the implementation of the schedule. Twenty-four months before the start of a schedule, all maintenance works larger than 7 days are published. This continues with a publication moment 12 months prior to the start of the annual schedule in which all projects with a duration longer than 48 h are published. In the year before the start of the new schedule, the train timetables and projects with a duration shorter than 48 h are scheduled. This study develops a tool to help the scheduling process before publication at 12 months. In this stage there are many projects to be scheduled, and this causes the arising of complex scheduling problems.

During the scheduling process, the rail infrastructure manager limits the expected passenger delays caused by projects by defining a set of rules that apply to the scheduling. These are described in the corridor book (Bekke, 2021). There are, for example, limits on the number of projects per year on predefined routes in the network, and rules to prevent that the alternative route will be blocked if the preferred route is also blocked. Furthermore, freight trains should have an alternative path if the original route is closed due to maintenance. Generally, all these constraints are location-specific and can be summarized in pairs of track segments that cannot be under maintenance simultaneously.

Our case study considers the scheduling of maintenance projects around the central station of Utrecht. Utrecht is a city located at the center of the Netherlands and, due to this location, the main station of Utrecht (Utrecht Centraal) is one of the most important stations of the country. In 2019, around 207,000 passengers used this station as their station of origin or destination on a daily basis. Another 65,000 passenger used this station as a transfer station on a daily basis. This makes it the station with the most daily travelers in the Netherlands, followed by Amsterdam Centraal (NS, 2019).

The infrastructure for the case study is shown in Fig. 2. The case study area contains 109 stations and a total of 290 travel connections. The case study area is bounded by important nodes in the railway network as the station of Zwolle, Eindhoven or Amsterdam Centraal.

A time period of three months, from the beginning of April 2023 to the end of June 2023, is considered. The scheduling happens on a daily level, i.e. maintenance projects are considered that have a duration of at least a day. This leads to a total number of five projects in that period that also directly reduce the accessibility of Utrecht. Maintenance projects are shown in Table 8 along with their initially proposed date by ProRail and their duration.

Furthermore, there are a total of 19 submitted event requests in the case study. An overview with the locations, dates and the expected number of visitors is shown in Table 7. The events are not necessarily hosted in Utrecht, but events are selected if a significant increase in the passenger demand is expected at Utrecht Centraal and could take place anywhere in the Netherlands. The event requests are considered in the planning process of ProRail as a hard scheduling rule, although the selected jobs have a conflict with event 52. A feasible solution for the model is obtained when the capacity of the alternative bus service is not exceeded on maintained tracks within the event request areas.

Table 6
Cardinality of input sets.

Set	Cardinality
$\mathcal{N}$	165
$\mathcal{A}$	420
$\mathcal{J}$	5
$\mathcal{C}$	446
$\mathcal{T}$	90

Other scheduling rules applied in the Netherlands are also considered. First, there are combinations of track segments that cannot be unavailable simultaneously. This is done, for example, to keep destinations for freight trains available or prevent multiple maintenance projects affecting the routes of the largest passenger streams. Between two maintenance projects on the same track segment there should be a minimum time interval of 25 days and maintenance can only be started in a weekend. The cardinality of all the aforementioned sets is presented in Table 6.

The model uses daily passenger demand predictions from the year 2022. These predictions are based on general trends, as seasonal and cyclical patterns. To include the passenger demand peaks during events, it is assumed that 30% of the event visitors will travel by train to reach the event. Due to the high variety of the nature of the events, the origins of passengers are equally distributed among all considered stations in the case study. Furthermore, the model considers $k = 3$ alternative travel routes for each origin–destination pair.

The goal of the case study is to find a range of schedules, each for a situation with a different bus capacity $\Lambda_{s,t}$. This shows how the model deals with different bus capacities and how schedules change based on the available bus capacity. This information can be crucial to prevent overcrowding, or passengers stranding.

6.2. Performance of the model with different capacity thresholds

The range of tested bus capacities is between 0 and 6900 passengers per hour with intervals of 100. This results in a total of 70 model runs. One final run is executed with unlimited bus capacity. Note that it is assumed that bus capacity is always sufficient outside the event requests and this results practically in the model without the consideration of event requests. On the other hand, the first run without any bus capacity practically means that no jobs could be scheduled inside event requests.

An optimal solution is found for all model runs with different bus capacity. The model run time was about two hours for each model. Table 9 shows a summary of the resulting objective values for the selected bus capacity values. From the results one can observe that the model runs with capacity $\Lambda_{st} \leq 1000$ resulted in exactly the same objective function value, as well as the model runs with $\Lambda_{st} \geq 1000$. The percentage difference in the two unique objective function values is 0.46%. This equals 17,500 min of passenger delay less, which is roughly 300 h, between the capacities below 1000 passengers per hour and above. This is a minor improvement, but most importantly the models show that there is a feasible solution for all capacities. The interesting part is that there are differences between solutions that achieve the same objective value. Fig. 3 shows key performance indicator (KPI) values for the schedules. These values indicate that there are differences in characteristics between schedules with the same objective function value. We have, for instances, minor differences in the delay, which are balanced by an increase in the number of passengers. It is worth

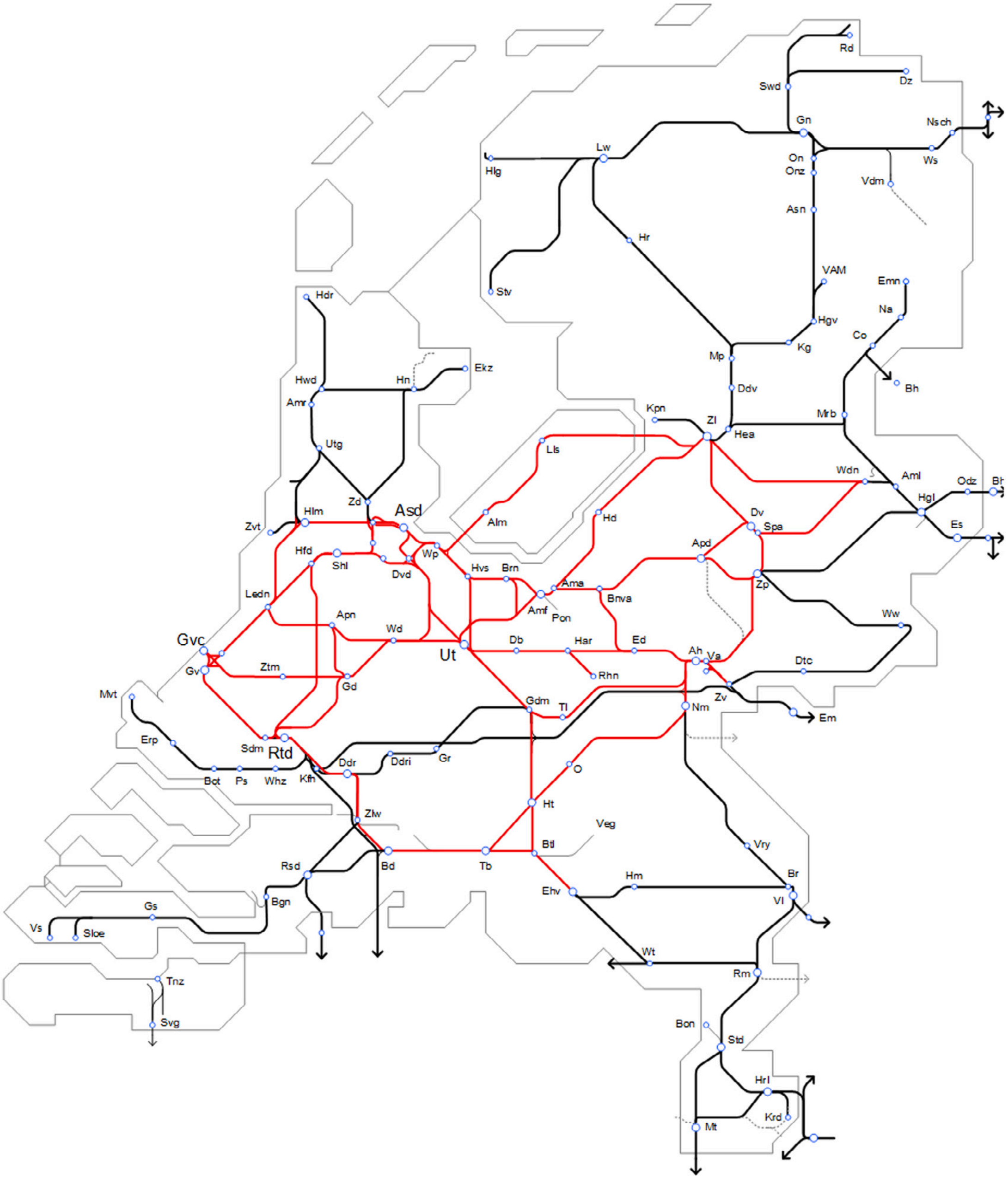

Fig. 2. Network for the case study.

noting that the schedules with $\Lambda_{st} \geq 1000$ show a conflict with an event request.

From the objective function combined with the KPIs, the following important observations are summarized:

- All model runs have a feasible solution. This implies that, even with a bus capacity of zero, there is a feasible schedule to the problem.
- Conflicts arise in the optimal solutions with a capacity above the capacity of 1100 passengers per hour. Before this point, no conflicts arise in the solution.
- Dependent on the capacity, there are multiple optimal schedules with different effects on passenger delays.
- Increased bus capacity may allow a job to be scheduled within an event request. The case study shows that this could lead to less passenger hindrance.

7. Conclusion

In this study, a railway maintenance scheduling problem is addressed with the goal to minimize passenger delays by considering event requests submitted by passenger operators. A MILP is developed that is able to find the optimal maintenance schedule that minimizes the passenger delays by considering event requests. This is applied in a case study on a part of the Dutch railway network.

The developed MILP in this study is able to construct optimal schedules for maintenance projects in order to minimize the passenger delays. To ensure that the optimal solution is found, practical limitations are included that represent the motivation of event requests instead of considering the event requests as hard constraints. That is, track segments included in event requests are only off-limits for maintenance projects if the capacity of the alternative services cannot satisfy the passenger demand in the peak hour after the end of the events. Furthermore, the model is able to handle other restrictions

Table 7
Events included in this case study.

Event ID	Event name	Start date	End date	Event station	Expected visitors/day
37	Marathon Rotterdam	April 2	April 2	Rotterdam Centraal	950 000
40	Paaspop	April 7	April 10	’s-Hertogenbosch	35 000
41	Paasraces Zandvoort	April 8	April 10	Amsterdam Centraal	105 000
42	Bloesemtocht Geldermalsen	April 15	April 15	Geldermalsen	35 000
50	Bekerfinale KNVB	April 30	April 30	Rotterdam Centraal	47 500
52	Libelle Zomerweek	May 9	May 21	Haarlem	11 500
53	Lakedance	May 13	May 14	Eindhoven	40 000
55	Marikenloop	May 21	May 21	Nijmegen	15 000
59	Emporium	May 27	May 29	Nijmegen	30 000
60	Pinksterraces Zandvoort	May 27	May 29	Haarlem	105 000
67	Concert Goffert I	June 9	June 9	Nijmegen	50 000
69	Concert Goffert II	June 10	June 10	Nijmegen	50 000
70	Parkpop Zuiderpark Den Haag	June 11	June 11	Den Haag Centraal	200 000
71	Concert Goffert III	June 11	June 11	Nijmegen	50 000
77	Concert Goffert III	June 11	June 11	Nijmegen	50 000
81	Veteranendag	June 24	June 24	Den Haag Centraal	75 000
82	Concert Goffert V	June 24	June 24	Nijmegen	50 000
84	Concert Goffert X	June 25	June 25	Nijmegen	50 000
88	Rijnweek	June 30	July 7	Ede-Wageningen	100 000

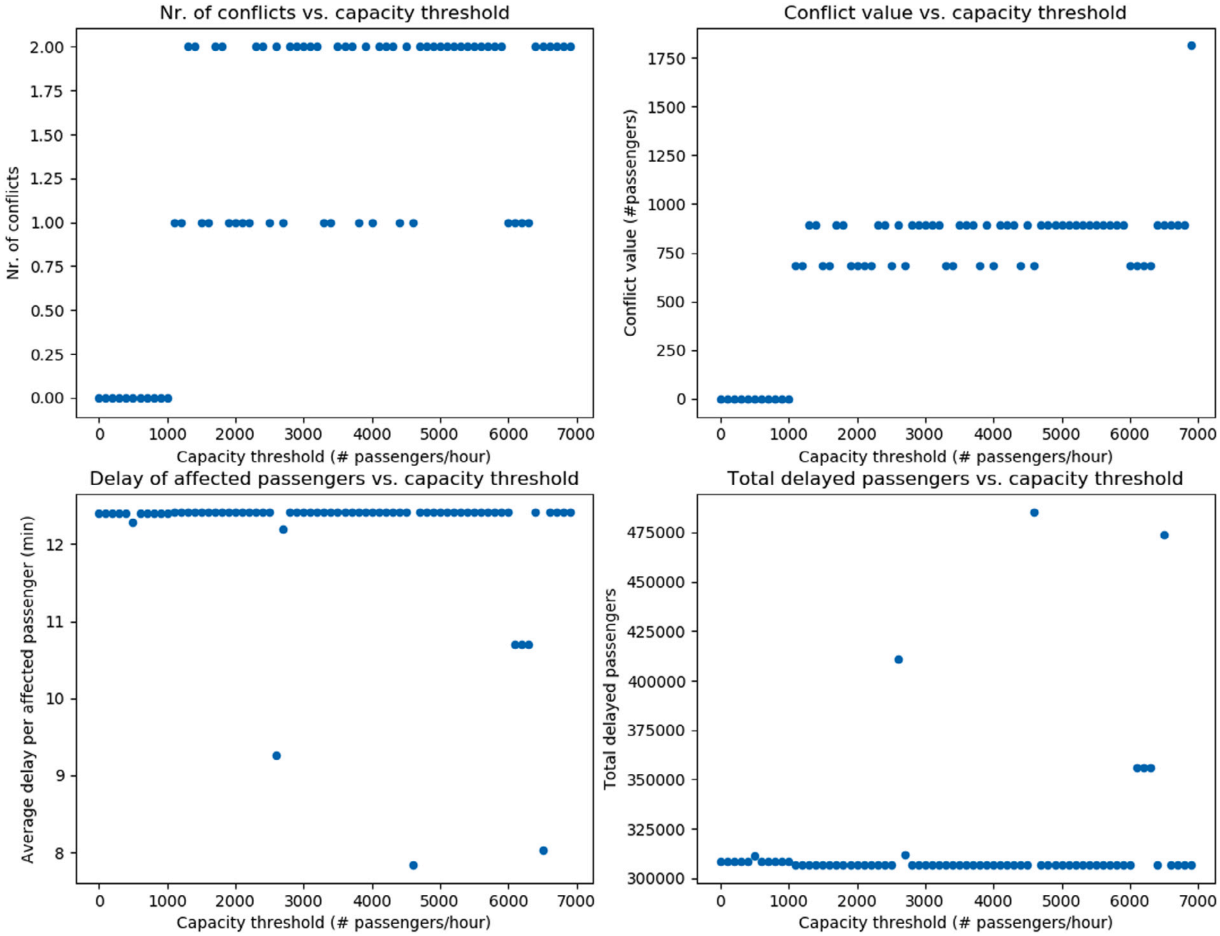

Fig. 3. Values for the KPIs of the model applied on the case study with increasing threshold values. Here the capacity threshold represents $\Lambda_{s,t}$.

that are imposed on maintenance scheduling to ensure that the resulting schedules are feasible for other rail operators, and not only for passenger operators.

Without the relaxation of event requests, we result in infeasible solutions. Practically this would mean that the approach followed in the past cannot be used due to the amount of event requests that are submitted each year. The proposed technique ensures that there is a way to provide better solutions for planners and reduces the amount of time spent on maintenance scheduling, which was done manually before.

The numerical experiments and the case study show that, due to the combinatorial nature of the problem, a globally optimal solution can be reached for instances with a limited size. Dependent on the capacity of the alternative services that can be provided, different optimal solutions can be obtained. The case study also shows that more flexibility in the scheduling may be obtained without increasing the passenger delays,

Table 8
Maintenance projects.

J	Start date	End date	Duration
249	April 7	April 8	2 days
289	April 9	April 10	2 days
358	May 13	May 13	1 day
360	May 13	May 14	2 days
466	June 17	June 18	2 days

Table 9
Maintenance projects.

Capacity	$\Lambda_{s,t} < 1000$	$1100 \leq \Lambda_{s,t} \leq 6900$
Total passenger delay	3, 822, 842	3, 805, 046

where decision makers are able to choose for solutions that reduce the total number of delayed passengers or the average delay time. These schedules could lead to more conflicts, although this should not necessarily lead to an increase in the passenger delay based on the results. It should be noted that are three players at stake in the maintenance scheduling problem: the infrastructure manager, the railway operators, and the passengers. Although our model uses passengers, a neutral entity, as arbiter of the conflicts, this probably will not be enough to solve them for all cases. For instance, a passenger operator may simply reject a scheduling proposal based on the importance/magnitude of the reason that led to the event request. Despite that, our model offers relaxed solutions that can form the basis for subsequent discussion and arbitration between the infrastructure manager and the passenger operators.

Our MILP was solved with the use of different solution strategies, including the implementation of a simulated annealing meta-heuristic to obtain an initial solution guess and the incorporation of valid inequalities or cutting planes. The use of the simulated annealing meta-heuristic had by far the biggest impact on the convergence of the branch-and-bound algorithm. Because the problem is computationally intractable, in future research there might be further experimentation with meta-heuristic approaches that can produce improved solutions for large instances. In addition, considering the fact that solution strategies 1 and 2 resulted in similar gaps on non-optimal solutions, it is highly recommended for future research that other metaheuristics will be tested in the hybridization of branch and bound inequalities constraints. Heuristics can also be devised to offer improved warm-start solutions. Finally, a more comprehensive scheduling model, considering aspects like opportunistic maintenance could also be a future research area.

Future directions could also involve passenger flow prediction for events or event venues. Predictions are hard due to the variety of events, but they are useful as they increase the accuracy of parameters used in the developed model. Currently, there are relatively few studies on predicting the travel behavior of passengers towards events. There are some methods for arrival predictions at stations near event venues (e.g. Rodrigues et al. (2017), Ni et al. (2017)) or short-term based methods to identify unexpected events (e.g. Li et al. (2017), Ni et al. (2017)). Applied on road networks, there are flow prediction methods for single events that study the impact of events on the road network traffic (e.g. Kwoczek et al. (2014, 2015), Tempelmeier et al. (2020), Di Martino et al. (2019)). There is still a gap in the prediction and analysis of passenger flows in railway networks though, and thus future directions could focus on developing new techniques for passenger flow prediction in railway networks during events. A last direction for future research could be the redirection of the model towards equity by having as objective the minimization of the maximum passenger delay, as excessive delays tend to be heavily penalized by legislation in many countries.

CRediT authorship contribution statement

Y.R. de Weert: Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **K. Gkiotsalitis:** Writing – review & editing, Validation, Supervision, Methodology, Conceptualization. **E.C. van Berkum:** Writing – review & editing, Validation, Supervision, Conceptualization.

Data availability

Data will be made available on request.

Appendix A. Scheduling model considering passenger delays and event requests

A complete overview of the mathematical model is presented by Eqs. (A.1a)–(A.1q).

$$\min \quad \sum_{(o,d)\in\mathcal{N}\times\mathcal{N}} \sum_{t=1}^{T} \phi_{odt}(v_{odt} - \Omega_{od}) \tag{A.1a}$$

$$\sum_{t=0}^{T-\pi_j+1} y_{jt} = 1 \qquad \forall j \in \mathcal{J} \tag{A.1b}$$

$$x_{at} + \sum_{t'=t-\pi_j+1}^{t} y_{jt'} \leq 1 \qquad \forall a \in \mathcal{A}, t \in \mathcal{T}, j \in \mathcal{J}_a \tag{A.1c}$$

$$\sum_{t\in\mathcal{T}} x_{at} = |\mathcal{T}| - \sum_{j\in\mathcal{J}_a} \pi_j \qquad \forall a \in \mathcal{A} \tag{A.1d}$$

$$w_{at} = x_{at}\omega_a^e + (1 - x_{at})\omega_a^i \qquad \forall a \in \mathcal{A}, t \in \mathcal{T} \tag{A.1e}$$

$$\sum_{a\in c} (1 - x_{at}) \leq 1 \qquad \forall c \in \mathcal{C}, t \in \mathcal{T}, j \in \mathcal{J} \tag{A.1f}$$

$$\sum_{j\in\mathcal{J}_a} \sum_{t'=t-\pi_j-\tau}^{T} y_{jt} \leq 1 \qquad \forall t \in \mathcal{T}, a \in \mathcal{A} \tag{A.1g}$$

$$\sum_{a\in s} \sum_{(o,d)\in\mathcal{N}\times\mathcal{N}} \sum_{i=1|a\in R_{odi}}^{k} h_{odt}^i \beta_{odt} \phi_{odt} \leq \Lambda_{st} + \sum_{a\in s} x_{at} M \qquad \forall s \in \mathcal{E}_t, t \in \mathcal{T} \tag{A.1h}$$

$$\sum_{i=1}^{k} h_{odt}^i = 1 \qquad \forall (o,d) \in \mathcal{N}\times\mathcal{N}, t \in \mathcal{T} \tag{A.1i}$$

$$v_{odt} \geq \sum_{a\in R_{od_k}} w_{at} - M(1 - h_{odt}^k) \qquad \forall i \in [k], (o,d) \in \mathcal{N}\times\mathcal{N}, t \in \mathcal{T} \tag{A.1j}$$

$$v_{odt} \leq \sum_{a\in R_{od_k}} w_{at} \qquad \forall i \in [k], (o,d) \in \mathcal{N}\times\mathcal{N}, t \in \mathcal{T} \tag{A.1k}$$

$$y_{jt} \in \{0,1\} \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \tag{A.1l}$$

$$x_{at} \in \{0,1\} \qquad \forall a \in \mathcal{A}, t \in \mathcal{T} \tag{A.1m}$$

$$z_{at} \in \{0,1\} \qquad \forall a \in \mathcal{A}_t, t \in \mathcal{T} \tag{A.1n}$$

$$h_{odt}^i \in \{0,1\} \qquad \forall i \in [k], (o,d) \in \mathcal{N}\times\mathcal{N}, t \in \mathcal{T} \tag{A.1o}$$

$$w_{at} \in \mathbb{R} \qquad \forall a \in \mathcal{A}, t \in \mathcal{T} \tag{A.1p}$$

$$v_{odt} \in \mathbb{R} \qquad \forall (o,d) \in \mathcal{N}\times\mathcal{N}, t \in \mathcal{T} \tag{A.1q}$$

The objective function in Eqs. (A.1a) minimizes the passenger delays. Eqs. (A.1b)–(A.1e) ensure that the maintenance jobs are scheduled exactly once in the allowed time periods and model effects on the network, i.e. the unavailability of arcs due to maintenance jobs. Eqs. (A.1f)–(A.1g) ensure that the corridor-book restrictions (combinations and minimum time interval) are respected, while Eq. (A.1h) models the passenger flow tolerance within event request areas. Eqs. (A.1i)–(A.1k) model the passenger route choice and the corresponding travel times from an origin to a destination. The remaining constraints in Eqs. (A.1l)–(A.1q) model the type of the considered variables in the model.

Appendix B. Valid inequalities

An inequality is valid if $\pi^T x \leq \pi_0$ holds for all feasible solutions x in a solution space, where $\pi \in \mathbb{R}^n$ and $\pi_0 \in \mathbb{R}$ (Wolsey, 1976). Useful valid inequalities could therefore reduce the solution space without cutting off feasible solutions. This may lead to improving computational times.

Valid inequalities can be derived from the Single Machine Scheduling problem. This is because there are similarities between the maintenance scheduling problem in this study and a general Single Machine Scheduling problem in which a set of jobs need to be scheduled on a single machine within a time horizon without overlapping. Now consider one arc in a railway network. A set of jobs need to be scheduled on that arc within the considered time period and overlap is not allowed. This basically implies that a single machine scheduling problem should be solved for each arc a with corresponding jobs $j \in J_a$ with additional restrictions emerging from the rules of the corridor-book.

Sousa and Wolsey (1992) formulated valid inequalities for the time-discretized single machine scheduling problem. The main proposed valid inequality is:

$$\sum_{s \in Q_j} y_{jt} + \sum_{l \neq j} \sum_{s \in Q'_l} y_{lt} \leq 1 \qquad \forall a \in \mathcal{A}, j \in \mathcal{J}_a, \forall t \in \mathcal{T}, \forall \Delta \in \{2, \dots, \bar{\pi}\}. \quad \text{(B.1)}$$

where $Q_j = [t - \pi_j + 1, t + \Delta - 1]$, $Q'_l = [t - \pi_l + \Delta, t]$ and $\bar{\pi} = \max_{l \neq j} p_l$. The valid inequality in Eq. (B.1) is facet-defining under the condition that $T \geq \sum_{j \in J_a} \pi_j + 3\bar{\pi}$ and $\pi_l + \bar{\pi} < t \leq T - \bar{\pi}$ for a job l and time $t \in \mathcal{T}$. This means that the inequality under these circumstances is not redundant (see proof in Sousa and Wolsey (1992)).

Another valid inequality can directly be derived from the definition of a pair of jobs that cannot overlap:

$$y_{jt} + y_{j't'} \leq 1 \quad \forall a \in \mathcal{A}, (j,t), (j',t') \in \mathcal{J}_a \times \mathcal{T} \, : \, j \neq j, t' \in (t - \pi_{j'} - \tau, t + \pi_j + \tau) \quad \text{(B.2)}$$

This constraint explicitly mentions that at a time t at most one of the jobs j and j' can be scheduled. This directly implies that this inequality is valid for the problem in this study.

References

Albrecht, A., Panton, D., Lee, D., 2013. Rescheduling rail networks with maintenance disruptions using problem space search. Comput. Oper. Res. 40 (3), 703–712. http://dx.doi.org/10.1016/j.cor.2010.09.001, URL https://www.sciencedirect.com/science/article/pii/S0305054810001905, Transport Scheduling.

Balas, E., 1971. Intersection cuts-a new type of cutting planes for integer programming. Oper. Res. 19–39.

Balas, E., Ceria, S., Cornuéjols, G., Natraj, N., 1996. Gomory cuts revisited. Oper. Res. Lett. 19 (1), 1–9.

Balas, E., Perregaard, M., 2002. Lift-and-project for mixed 0–1 programming: recent progress. Discrete Appl. Math. 123 (1–3), 129–154.

Bekke, C., 2021. Corridorboek 2023.

Boland, N., Kalinowski, T., Kaur, S., 2016. Scheduling arc shut downs in a network to maximize flow over time with a bounded number of jobs per time period. J. Comb. Optim. 32 (3), 885–905.

Boland, N., Kalinowski, T., Waterer, H., Zheng, L., 2013. Mixed integer programming based maintenance scheduling for the Hunter Valley coal chain. J. Sched. 16 (6), 649–659.

Boland, N., Kalinowski, T., Waterer, H., Zheng, L., 2014. Scheduling arc maintenance jobs in a network to maximize total flow over time. Discrete Appl. Math. 163, 34–52. http://dx.doi.org/10.1016/j.dam.2012.05.027, URL https://www.sciencedirect.com/science/article/pii/S0166218X12002338, Matheuristics 2010.

Bonami, P., 2011. Lift-and-project cuts for mixed integer convex programs. In: International Conference on Integer Programming and Combinatorial Optimization. Springer, pp. 52–64.

Budai, G., Huisman, D., Dekker, R., 2006. Scheduling preventive railway maintenance activities. J. Oper. Res. Soc. 57 (9), 1035–1044, URL http://www.jstor.org/stable/4102318.

Budai-Balke, G., Dekker, R., Kaymak, U., 2009. Genetic and memetic algorithms for scheduling railway maintenance activities. Technical Report.

Buurman, B., Gkiotsalitis, K., Van Berkum, E., 2023. Railway maintenance reservation scheduling considering detouring delays and maintenance demand. J. Rail Transp. Plan. Manag. 25, 100359.

Caprara, A., Fischetti, M., 1996. {0, 1/2}-Chvátal-Gomory cuts. Math. Program. 74 (3), 221–235.

Caprara, A., Fischetti, M., Letchford, A.N., 2000. On the separation of maximally violated mod-k cuts. Math. Program. 87 (1), 37–56.

Cheung, B.S., Chow, K., Hui, L.C., Yong, A.M., 1999. Railway track possession assignment using constraint satisfaction. Eng. Appl. Artif. Intell. 12 (5), 599–611.

Chvátal, V., 1973. Edmonds polytopes and a hierarchy of combinatorial problems. Discrete Math. 4 (4), 305–337.

de Jonge, M., 2017. Railway maintenance reservation scheduling considering train traffic and maintenance demand. (Master's thesis). University of Twente.

de Weert, Y., 2022. Improving the scheduling of railway maintenance projects by considering passenger hindrance and event requests of passenger operators. (Master's thesis). University of Twente, Drienerlolaan 5, 7522 NB Enschede, Netherlands.

Di Martino, S., Kwoczek, S., Rossi, S., 2019. Predicting the spatial impact of planned special events. In: International Symposium on Web and Wireless Geographical Information Systems. Springer, pp. 102–117.

Forsgren, M., Aronsson, M., Gestrelius, S., 2013. Maintaining tracks and traffic flow at the same time. J. Rail Transp. Plan. Manag. 3 (3), 111–123. http://dx.doi.org/10.1016/j.jrtpm.2013.11.001, URL https://www.sciencedirect.com/science/article/pii/S2210970613000267, Robust Rescheduling and Capacity Use.

Gkiotsalitis, K., 2023. Public Transport Optimization. Springer Nature.

Gkonou, N., Nisyrios, E., Gkiotsalitis, K., 2023. Combined optimization of maintenance works and crews in railway networks. Appl. Sci. 13 (18), 10503.

Gondzio, J., 1998. Warm start of the primal-dual method applied in the cutting-plane scheme. Math. Program. 83 (1–3), 125–143.

Gu, Z., Nemhauser, G.L., Savelsbergh, M.W., 1999. Lifted flow cover inequalities for mixed 0-1 integer programs. Math. Program. 85 (3), 439–467.

Günlük, O., Pochet, Y., 2001. Mixing mixed-integer inequalities. Math. Program. 90 (3), 429–457.

Higgins, A., 1998. Scheduling of railway track maintenance activities and crews. J. Oper. Res. Soc. 49 (10), 1026–1033, URL http://www.jstor.org/stable/3010526.

IEA, 2019. The future of rail. URL https://www.iea.org/reports/the-future-of-rail, (Accessed: 2021-10-19).

Kirkpatrick, S., Gelatt, C., Vecchi, M., 1983. Optimization by simulated annealing. Science 220, 671–680. http://dx.doi.org/10.1126/science.220.4598.671.

Krueger, H., 1999. Parametric modeling in rail capacity planning. In: WSC'99. 1999 Winter Simulation Conference Proceedings. 'Simulation - A Bridge to the Future' (Cat. No.99CH37038). Vol. 2, pp. 1194–1200. http://dx.doi.org/10.1109/WSC.1999.816840.

Kwoczek, S., Di Martino, S., Nejdl, W., 2014. Predicting and visualizing traffic congestion in the presence of planned special events. J. Vis. Lang. Comput. 25 (6), 973–980. http://dx.doi.org/10.1016/j.jvlc.2014.10.028, URL https://www.sciencedirect.com/science/article/pii/S1045926X14001219, Distributed Multimedia Systems DMS2014 Part I.

Kwoczek, S., Di Martino, S., Nejdl, W., 2015. Stuck around the stadium? An approach to identify road segments affected by planned special events. In: 2015 IEEE 18th International Conference on Intelligent Transportation Systems. pp. 1255–1260. http://dx.doi.org/10.1109/ITSC.2015.206.

Li, Y., Wang, X., Sun, S., Ma, X., Lu, G., 2017. Forecasting short-term subway passenger flow under special events scenarios using multiscale radial basis function networks. Transp. Res. C 77, 306–328. http://dx.doi.org/10.1016/j.trc.2017.02.005, URL https://www.sciencedirect.com/science/article/pii/S0968090X17300451.

Lidén, T., 2015. Railway infrastructure maintenance - a survey of planning problems and conducted research. Transp. Res. Procedia 10, 574–583. http://dx.doi.org/10.1016/j.trpro.2015.09.011, URL https://www.sciencedirect.com/science/article/pii/S2352146515001982, 18th Euro Working Group on Transportation, EWGT 2015, 14-16 July 2015, Delft, The Netherlands.

Lidén, T., Joborn, M., 2017. An optimization model for integrated planning of railway traffic and network maintenance. Transp. Res. C 74, 327–347. http://dx.doi.org/10.1016/j.trc.2016.11.016, URL https://www.sciencedirect.com/science/article/pii/S0968090X16302340.

Lidén, T., Kalinowski, T., Waterer, H., 2018. Resource considerations for integrated planning of railway traffic and maintenance windows. J. Rail Transp. Plan. Manag. 8 (1), 1–15. http://dx.doi.org/10.1016/j.jrtpm.2018.02.001, URL https://www.sciencedirect.com/science/article/pii/S2210970617300860.

Macedo, R., Benmansour, R., Artiba, A., Mladenović, N., Urošević, D., 2017. Scheduling preventive railway maintenance activities with resource constraints. Electron. Notes Discrete Math. 58, 215–222.

Meng, L., Mu, C., Hong, X., Chen, R., Luan, X., Ma, T., 2017. Integrated optimization model on maintenance time window and train timetabling. In: International Conference on Electrical and Information Technologies for Rail Transportation. Springer, pp. 861–872.

Nemhauser, G.L., Vance, P.H., 1994. Lifted cover facets of the 0–1 knapsack polytope with GUB constraints. Oper. Res. Lett. 16 (5), 255–263. http://dx.doi.org/10.1016/0167-6377(94)90038-8, URL https://www.sciencedirect.com/science/article/pii/0167637794900388.

Ni, M., He, Q., Gao, J., 2017. Forecasting the subway passenger flow under event occurrences with social media. IEEE Trans. Intell. Transp. Syst. 18 (6), 1623–1632. http://dx.doi.org/10.1109/TITS.2016.2611644.

Nijland, F., Gkiotsalitis, K., van Berkum, E., 2021. Improving railway maintenance schedules by considering hindrance and capacity constraints. Transp. Res. C 126, 103108. http://dx.doi.org/10.1016/j.trc.2021.103108, URL https://www.sciencedirect.com/science/article/pii/S0968090X21001273.

NS, 2019. Capaciteitsverdeling. URL https://dashboards.nsjaarverslag.nl/reizigersgedrag/utrecht-centraal, (Accessed: 2022-3-6).

Padberg, M.W., 1973. On the facial structure of set packing polyhedra. Math. Program. 5 (1), 199–215.

Padberg, M., 1989. The boolean quadric polytope: some characteristics, facets and relatives. Math. Program. 45 (1), 139–172.

Padberg, M.W., 2001. Improving LP-representations of zero-one linear programs.

Pour, S.M., Drake, J.H., Ejlertsen, L.S., Rasmussen, K.M., Burke, E.K., 2018. A hybrid constraint programming/mixed integer programming framework for the preventive signaling maintenance crew scheduling problem. European J. Oper. Res. 269 (1), 341–352.

Pour, S.M., Marjani Rasmussen, K., Drake, J.H., Burke, E.K., 2019. A constructive framework for the preventive signalling maintenance crew scheduling problem in the Danish railway system. J. Oper. Res. Soc. 70 (11), 1965–1982.

ProRail, 2022. Capaciteitsverdeling. URL https://www.prorail.nl/samenwerken/vervoerders/spoorruimte-aanvragen, (Accessed: 2022-04-11).

Robbins, D., Dickinson, J., Calver, S., 2007. Planning transport for special events: A conceptual framework and future agenda for research. Int. J. Tourism Res. 9 (5), 303–314.

Rodrigues, F., Borysov, S.S., Ribeiro, B., Pereira, F.C., 2017. A Bayesian additive model for understanding public transport usage in special events. IEEE Trans. Pattern Anal. Mach. Intell. 39 (11), 2113–2126. http://dx.doi.org/10.1109/TPAMI.2016.2635136.

Schasfoort, B.B., Gkiotsalitis, K., Eikenbroek, O.A., Van Berkum, E.C., 2020. A dynamic model for real-time track assignment at railway yards. J. Rail Transp. Plan. Manag. 14, 100198.

Sousa, J.P., Wolsey, L.A., 1992. A time indexed formulation of non-preemptive single machine scheduling problems. Math. Program. 54 (1), 353–367.

Tempelmeier, N., Dietze, S., Demidova, E., 2020. Crosstown traffic-supervised prediction of impact of planned special events on urban traffic. GeoInformatica 24 (2), 339–370.

Trommelen, W., Gkiotsalitis, K., Van Berkum, E.C., 2022. Optimization model for rail line crossover design considering the cost of delay. Transp. Res. Rec. 2676 (4), 554–569.

van Roy, T.J., Wolsey, L.A., 1987. Solving mixed integer programming problems using automatic reformulation. Oper. Res. 45–57.

van Zante-de Fokkert, J., Hertog, D.d., van den Berg, F., Verhoeven, J., 2007. The Netherlands schedules track maintenance to improve track workers' safety. Interfaces 133–142.

Weismantel, R., 1997. On the 0/1 knapsack polytope. Math. Program. 77 (3), 49–68.

Wolsey, L.A., 1976. Facets and strong valid inequalities for integer programs. Oper. Res. 24 (2), 367–372.

Yen, J.Y., 1971. Finding the k shortest loopless paths in a network. Manag. Sci. 17 (11), 712–716.

Zhang, T., Andrews, J., Wang, R., 2013. Optimal scheduling of track maintenance on a railway network. Qual. Reliab. Eng. Int. 29 (2), 285–297.

Zhang, C., Gao, Y., Yang, L., Kumar, U., Gao, Z., 2019. Integrated optimization of train scheduling and maintenance planning on high-speed railway corridors. Omega 87, 86–104. http://dx.doi.org/10.1016/j.omega.2018.08.005, URL https://www.sciencedirect.com/science/article/pii/S0305048317310575.