Improving the scheduling of railway maintenance projects by considering passenger hindrance and event requests of passenger operators

Yoran de Weert

June 7, 2022

UNIVERSITY OF TWENTE.

ProRail

Contact information

Student number	s1693859
Email address	y.r.deweert@student.utwente.nl
	Yoran.deWeert@prorail.nl
Commissioned by:	ProRail - Department ”Capaciteit voor Beheer”
Internship:	November 2021 - June 2022
External supervisors:	H. Zandman BSc.
	Floris Nijland MSc.
Internal supervisors:	Prof. Dr. Ir. E.C. van Berkum
	Dr. K. Gkiotsalitis
University	University of Twente
Master	Civil Engineering & Management
Track	Transport Engineering & Management
	Transport & Logistics
Version	Final version

Enschede, June 7, 2022

I. Preface

Before you lies the thesis 'Improving the scheduling of railway maintenance projects by considering passenger hindrance and event requests of passenger operators'. This thesis has been written in the final phase of the master Civil Engineering & Management at the University of Twente. From November 2021 until June 2022, I have been engaged in the writing and researching of this thesis as an intern of ProRail.

This thesis could not have been written without the support of my supervisors from ProRail, Harmen Zandman and Floris Nijland. They initiated the research problem and guided me throughout the whole process with ideas, contacts and tips. I would to thank them for the great experience at ProRail and also the opportunity for me to pursue my thesis here.
From December until Februari, it was obligated to work from home due to the COVID-19 pandemic. Nevertheless, the online weekly meetings with colleagues of ProRail made me feel welcome within the organisation and the colleagues helped me with the thesis whenever it was needed. Therefore, I would like to thank them as well. In the remaining period, I could work in the main office of ProRail. It was a really nice experience to see and speak to my colleagues in person.

Furthermore, I would like to thank Eric van Berkum and Konstantinos Gkiotsalitis for the feedback they gave on my thesis, and on being part of the graduation committee. Discussions about the developed model and comments on the thesis really improved it.

A special thanks to my flatmates, friends and family. During my spare time, they provided me with some distraction from my thesis by playing games, sporting activities, just a nice little chat or other fun social activities.

During my internship, I have been offered a job at ProRail. The job concerns maintenance scheduling in the region Zee tot Zevenaar. An exciting challenge and I am really looking forward to it.

I hope you enjoy reading my thesis.

Yoran de Weert
June 7, 2022

II. Summary

In future scenarios, it is expected that passenger activities on railway networks will double by 2050. To handle the passenger demand in these scenarios, railway capacity planning needs to be adapted. One fundamental aspect of the railway capacity planning is the scheduling of large maintenance projects. These projects cause a significant amount of hindrance as track segments are unavailable for train traffic for some consecutive days.

To minimize the passenger hindrance induced by these projects, the main principle of the rail-infra manager ProRail is to avoid that these projects are scheduled during events. Therefore passenger operators can submit event request, i.e. a time period and location in which no project should be scheduled. Currently these event requests are considered as hard constraints and due to the number of event requests, the flexibility on maintenance scheduling decreases. Not being able to schedule the maintenance projects outside of the event request areas results in a conflict between ProRail and the passenger operator. This is solved by iterative negotiations, although quantitative methods may be able to provide better solutions or give more insight into the value of these conflicts.

In this thesis, the aforementioned problem is addressed in a quantitative manner. The focus is on scheduling maintenance projects to minimize passenger hindrance by considering these event requests. This is done by introducing a Mixed Integer Linear Program (MILP) that schedules maintenance projects to minimize the passenger hindrance. The MILP also considers capacity constraints based on the capacity of alternative services that can be provided within event request areas. Furthermore, methods are considered that might reduce the computational costs. A computational study shows that, compared to a naive branch-and-bound algorithm, the inclusion of a heuristic shows the best improvement on computational costs. Cutting planes and valid inequalities make the search more efficient, but do not reduce computational times.

A case study on the Dutch railway network tested a range of capacities for alternative services. The resulting schedules do not create any conflicts if a capacity equal or below 1000 passengers per hour is considered and one or two conflicts with events requests for higher capacity thresholds. The resulting schedules show minor improvements when the capacity is over 1000 and also shows an increase in the flexibility to schedule projects. This allows decision makers to choose from a set of optimal schedules with different characteristics on the passenger hindrance.

III. Samenvatting

In de toekomst wordt verwacht dat reizigersactiviteiten op het spoor zullen verdubbelen tegen 2050. Om de passagiersvraag in de toekomst aan te kunnen, moeten er veranderingen komen in het maken van een capaciteitsplanning. Een fundamenteel aspect van deze planning van de spoorwegcapaciteit is het plannen van onderhoudsprojecten. Deze projecten kunnen een aanzienlijke hoeveelheid hinder veroorzaken als spoorsegmenten voor een langere periode niet beschikbaar zijn voor het treinverkeer.

Om de veroorzaakte passagiershinder door deze projecten tot een minimum te beperken, is het principe van de spoor-infrabeheerder ProRail om te voorkomen dat deze projecten tijdens evenementen worden ingepland. Daarom kunnen reizigersvervoerders een evenement-aanvraag indienen, d.w.z. een tijdsperiode en locatie waarin geen projecten moeten worden gepland. Momenteel worden deze evenementen-aanvragen als harde beperkingen beschouwd en door het aantal evenement-aanvragen neemt het aantal mogelijkheden om een onderhoudsplanning te maken af. Als het niet lukt om buiten de tijden en gebieden van een evenemten aanvraag heen te planne, resulteert dit in een conflict tussen ProRail en de reizigersvervoerder. Dit wordt momenteel opgelost door onderhandelingen, hoewel kwantitatieve methoden een uitkomst bieden om mogelijk betere oplossingen te bieden of meer inzicht geven in de waarde van deze conflicten.

In dit proefschrift wordt het bovengenoemde probleem op een kwantitatieve manier aangepakt. De focus is om het plannen van onderhoudsprojecten zo te plannen dat de passagiershinder tot een minimum te beperkt wordt, terwijl evenementen-aanvragen in overweging worden genomen. Dit wordt gedaan door een Mixed Integer Linear Program (MILP) te introduceren dat onderhoudsprojecten plant met als doel de passagiershinder te minimaliseren. De MILP overweegt ook capaciteitsbeperkingen op basis van de capaciteit die alternatieve diensten kunnen leveren rondom het gebied van een evenement. Verder worden methodes overwogen die de rekenkosten kunnen reduceren. Een computationele studie toont aan dat, vergeleken met een simpel branch-and-bound algoritme, een meta-heuristiek het beste werkt om de rekenkosten te verminderen. Andere methodes als snijvlakken en geldige ongelijkheden maken het zoeken efficiënter, maar verkorten de computationale tijd niet.

In een casus op het Nederlandse spoorwegnet werden een reeks capaciteiten voor alternatieve diensten getest. De resulterende schema's veroorzaken geen conflicten als de capaciteit gelijk aan of lager is dan 1000 passagiers per uur en voor hogere waardes voor de capaciteit zullen er in een optimaal schema een of twee conflicten voorkomen met evenementen-aanvragen. De resulterende schema's tonen kleine verbeteringen wanneer de capaciteit groter is dan 1000 en toont ook een toename van de flexibiliteit om projecten te plannen. Dit stelt beleidsbepalers in staat om te kiezen uit een reeks optimale schema's, elk met verschillende kenmerken rondom passagiershinder.

Contents

List of Tables

List of Figures

Improving the scheduling of railway maintenance projects by considering passenger hindrance and event requests of passenger operators

Y.R. de Weert*

Abstract

In future scenarios, it is expected that passenger activities on railway networks will double by 2050. To handle the passenger demand in these scenarios, railway capacity planning needs to be adapted. One fundamental aspect of the railway capacity planning is the scheduling of large maintenance projects. It should be avoided that these projects are scheduled during events and therefore passenger operators can submit event request, i.e. a time period and location in which no project should be scheduled. Currently these event requests are considered as hard constraints and due to the number of event requests, the flexibility in maintenance scheduling decreases resulting in conflicts. In this thesis, the focus is on scheduling maintenance projects to minimize passenger hindrance by considering these event requests. This problem is addressed by introducing a Mixed Integer Linear Program (MILP) that minimizes passenger hindrance while scheduling maintenance projects, which includes capacity constraints for alternative services in event request areas during maintenance projects. A case study on the Dutch railway network shows minor improvements when event requests are not considered as hard constraints and also shows an increase in the flexibility to schedule projects. This allows decision makers to choose from a set of optimal schedules with different characteristics.

Keywords: railway; maintenance projects; scheduling; events; passenger hindrance.

1. Introduction

Railway infrastructure plays an important role in our society. It is important for the transportation of people and freight and becomes even more important in the future. 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. Therefore, the need for additional capacity in railway networks is rising.

The capacity of a railway line is by Krueger (1999) defined 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 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) The first category concerns inspections and small reparations as inspection

*Corresponding author
Email address: `y.r.deweert@student.utwente.nl` (Y.R. de Weert)

of rails, switch, level crossing, overhead wire, signalling 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 (2021) conducted studies that concern optimization of scheduling small maintenance routine works in the nightly train-free periods.

Maintenance projects 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) as for example ballast cleaning and tamping. As these projects cannot only be scheduled during nightly train-free periods, train traffic is cancelled or diverted 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 cancelled 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)

In the Netherlands, the rail-infra 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 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. Quantitative methods are able to evaluate such situations and provide optimal solutions to minimize hindrance for passengers.

In this thesis, the aforementioned problem is addressed to construct an optimal schedule for maintenance projects such that the passenger hindrance is 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 new information about events for example, are solved ad hoc.(ProRail, 2022) The scope of this thesis is on the construction of the annual schedule. In particular, the focus is on the development of a method that optimally schedules maintenance projects to minimize passenger hindrance by considering the event requests of the passenger operator.

The thesis is structured as follows. Chapter 2 provides a literature review on maintenance scheduling and the research goals. Chapter 3 contains the problem description and is followed by the methodology in Chapter 4. In Chapter 5 computational costs are examined and also contains the results of a case study. Chapter 6 concludes the research and the thesis ends with a discussion in Chapter 7.

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 thesis 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 thesis 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 the 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 since the problem size and complexity makes the computational burden high.

Cheung et al. (1999) proposes a method to assign maintenance jobs to track segments on 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 is 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 for example Budai-Balke et al. (2009), extending the research by proposing more genetic and memoric heuristics for the PMSP, and more recently, Macedo et al. (2017) solve the PSMP with resource constraints using a Variable Neighborhood Search (VNS) algorithm.

To improve 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 STGs (single track grids), sets of working zones on a railway corridor that can be blocked simultaneously, are closed to trains exactly once to perform small routine maintenance works. The MIP minimizes on 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, they divide small routine maintenance works in three components due to differences in the hindrance. They can find exact solutions using their developed MILP, although for bigger instances they proposed a combination of an exact method and meta-heuristics to reduce the computational burden.

Buurman (2021) optimizes the nightly maintenance schedules considering the hindrance for train operators and the flexibility for contractors. The question is addressed of how often, when and where nightly maintenance slots need to be reserved. A multi-objective optimization model is proposed and solved using heuristics. The result are Pareto optimal solutions, giving the decision maker multiple options to choose from based on the value of their objectives.

The work of de Jonge (2017) focuses 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 result show 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 project under the weight of event requests. A weight was assigned to each event and weekend to maximize the sum over the violated event requests.

Zhang et al. (2013) consider 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 are monitored and this study proposes a method to create a maintenance schedule based on the information from the monitoring procedure. The problem is 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) proposes an integer programming formulation for the maximum total flow in a network with flexible arc outages. Here, the goal is to schedule all given jobs and maximize the total flow over the planning time horizon. The problem is NP-hard and for practical purposes, they propose heuristics to solve the problems. This model is made for a coal export supply chain, and Boland et al. (2013) continues on this problem and adds more problem-specific constraints. Boland et al. (2016) expands their original MaxTFFAO 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 is conducted by Albrecht et al. (2013). The goal is 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) present a MIP model optimizing both train timetables and preventive maintenance simultaneously. In their model, they allow 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 can solve the proposed MILP optimally within one hour on a laptop computer. Lidén et al. (2018) extends the research by explicitly focusing on crew resource constraints, meaning that there is a maximum working hours and a a minimum rest time between working days. Furthermore, they present a MILP for this problem.

Zhang et al. (2019) also integrated train timetabling and maintenance scheduling, but for high-speed railway corridors with SDSA-trains (sunset departure, sunset arrival 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) addresses the integration of train timetabling and maintenance time windows scheduling, but consider speed restrictions on trains. They solve their problem also optimally by solving a MILP developed for their problem.

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). Table 1 provides a summary of the reviewed studies and corresponding topics and objectives they address.

2.2. Research gap

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

The 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) focus on a combination of routine works and project and minimizes possession and maintenance costs. The work of de Jonge (2017) maximizes the number of possessions. Possession scheduling, however, has a significant impact on the quality of train timetables or other impact on train traffic, although these factors are barely included in these studies.

Some studies do focus on dynamic networks. (Boland et al., 2013) has a dynamic arc capacities, but does not deal with passenger flows and are able to choose where the products flows. Passengers, however, are free to choose their route and should be modelled differently. The study of Buurman (2021) models train routes in a network, but passenger routes are never considered. The integration of train timetabling and maintenance scheduling also models train routes, but do not consider passenger routing or the hindrance on passengers. Passenger hindrance is related to hindrance on trains, but objectives as minimizing train delays or minimizing the possession time are not completely equivalent.

Furthermore, there is relatively little attention to abnormal circumstances that could, together with a maintenance schedule, heavily increase passenger hindrance. Most objectives in

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)

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

the construction maintenance schedules are based on the minimization of possession time, train traffic hindrance or based on track segment quality. External factors as for example events or holidays, are not considered in these studies. The work of de Jonge (2017) is the only reviewed work that also schedules maintenance projects while considering events in the optimization by minimizing the number of projects during an event. Passenger hindrance is, however, not considered in that study.

2.3. Research questions

The aim of this study is to find an optimal maintenance schedule that minimizes passenger hindrance considering the impact of the 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 hindrance and event requests. The aim is translated into the following research question:

How can maintenance projects optimally be scheduled when minimizing passenger hindrance considering event requests?

A set of sub-questions are formulated to support the research question. The first sub-question is included to understand the event requests submitted by the operator. There is a lack of knowledge about the motives behind the set of event requests that should be accounted for in the maintenance schedule.

1. What are factors that the passenger operator considers when selecting track segments that are included in the event requests, and how can these factors be included in the model?

Furthermore, the event requests are meant to represent track segments that have abnormally large passenger flows due to the event. Significant differences, however, might be noticeable between affected areas indicated from the event requests and the real situation. This gives insight into the usefulness of these event requests in general. The sub-question is formulated as follows:

2. What is the difference in the maintenance schedule between a model that includes event requests and a model that does not include event requests?

2.4. Contribution

The thesis contributes to the topic of maintenance scheduling by presenting a framework to determine the optimal maintenance schedule such that passenger hindrance is minimized. It is closest related to 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 hindrance by examining and modelling passenger route choices.

The model that is developed is a MILP and solved with exact methods, showing that optimal solutions can be achieved for relatively small to medium sized instances. The application in this thesis 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

The Dutch rail infra manager ProRail constructs an annual schedule for maintenance projects as part of the capacity distribution on the railway network. This is a large process involving multiple stages, which starts by identifying maintenance requirements and the time needed for singular maintenance activities on track segments in regional meetings. The outcome of these meetings results in maintenance projects, and the dates to start the project are discussed in a national meeting, which results in the maintenance project schedule. This is an iterative process throughout the year as new information about maintenance works or external factors as events keeps arriving. A year before the schedule goes into effect, the schedule is published. Problems with the schedule are solved ad hoc after publishing, although this is not in the scope of the thesis.

The relevant stakeholders for this thesis in these national meetings are the rail infra manager and the passenger operators. The role of other stakeholders as maintenance contractors or freight operators are not considered in this thesis.

3.1. Rail infra manager

The rail infra manager has the priority to keep the conditions of the track in a sufficient state and while achieving this, also minimizing the hindrance for passengers, passenger operators and other stakeholders. In this thesis the focus is particularly on the minimization of passenger hindrance, while normally other hindrance factors as hindrance on freight trains or parked stock are also considered.

The maintenance project scheduling happens according to a set of rules that are defined in the corridor book. These rules are made to ensure that hindrance caused by a maintenance project is minimized. This concerns all involved parties; for example, detours should be available for both freight and passenger operators, maintenance facilities for the railway network should still be accessible and maintenance projects should not be scheduled around the location and time of events.

Focusing on events, there are some guidelines in the corridor book to schedule maintenance projects around events. First, some events in the Netherlands as King's Day or Liberation Day are periodically and the rail infra manager already has a list with recurrent events. These events are categorized according to Table 2 which indicates a set of rules that should be followed for that event.

Table 2: Event categorization for maintenance scheduling implications. Note: Maintenance works are allowed if the train paths are not affected.

Category	**Restriction on maintenance works**	**Allowed maintenance works in affected area**	**Nr. travellers**
E1: National	National	No	>150.000
E2: Interregional	2nd transfer station indicated track segments	1st transfer station: No 2nd transfer station if delay < 30 min Indicated track segments if alternative available	> 10.000
E3: Regional	1st transfer station	1st transfer station if delay < 30 min	> 3.000
E4: Special trains	-	Delay < 30 min	-
E5: Special events	-	-	-

Furthermore, a set of commonly used event venues for relatively large sized events is included in the corridorbook that indicates the region around these venues where no maintenance can be scheduled when events take place. They have a standard framework for regular sized events and for some venues an extended framework is presented including some additional track segments for larger sized events. Figure 1 shows an example of the framework for the events that take place in Utrecht. Finally, as mentioned in the introduction, train operators can submit event requests for events in the concerned year. This happens in a system that contains an overview of entities related to possessions on maintenance tracks. For an event request, they submit the set of track segments, one of the categories and a reason why the track segment is included. There are three options. The first option is the primary goal of the event requests, which is to indicate track segments with high passenger peaks due to the event. The second reason is logistical, meaning that maintenance scheduled increases hindrance for the planning of trains and carriers. The last reason is political and is used for extraordinary reasons.

From experience of the rail infra manager, most issues arise from the event requests. 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. As a result, the best schedule should be found where all demand for maintenance is met, and where possibly some event requests are violated. Then, key is to find a schedule that minimizes the passenger hindrance.

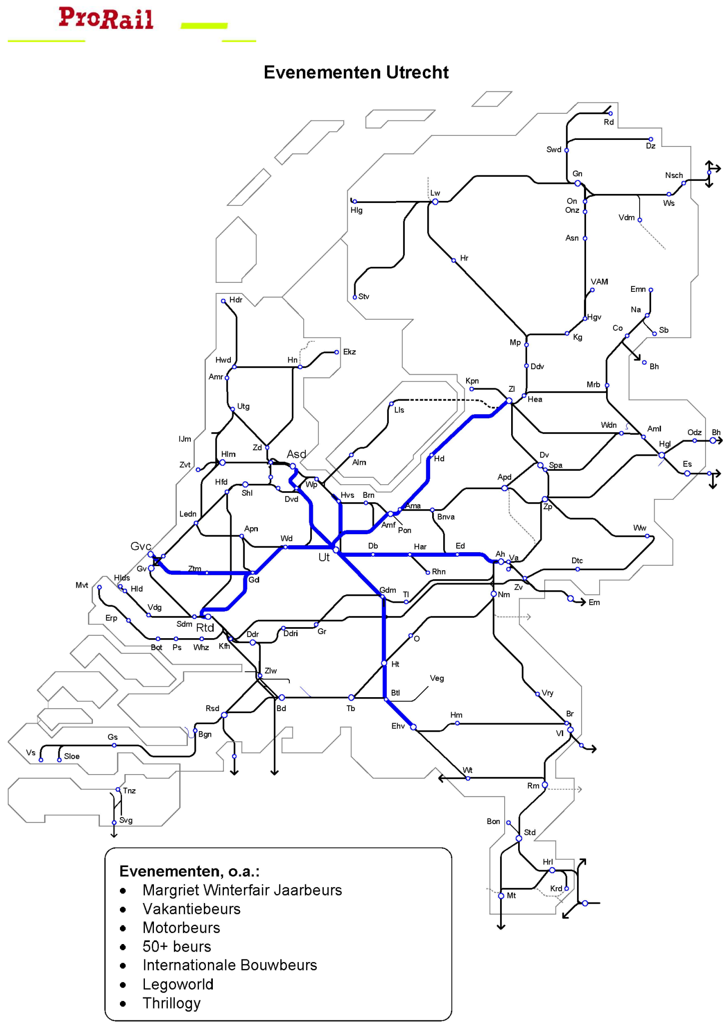

Figure 1: Standard set of track segments that should be available for train traffic during events in Utrecht. (Bekke, 2021)

3.2. Passenger operators

Passenger operators have the goal to convey passengers from a origin to a destination without inconveniences to achieve a high service quality. Event requests play an important role to achieve this goal. Event requests are submitted to prevent overcrowding and dangerous situations at stations and in trains before and after events due to scheduled maintenance works.

Passenger operators have a detailed insight in the passenger demand. Using this data, the passenger demand induced by an event and main directions of those passengers are predicted and serves as a base for these event requests. This is manually converted to a set of track segments of which it is expected that a significant amount of passengers will use trains that run over these tracks. If needed, additional track segments are included in an event request to reduce hindrance on the logistics or parked stock.

In general, scheduled maintenance works on the tracks causes inconveniences for passengers as increasing travel times, additional transfers, overcrowded trains or traveling with alternative services. Under a normal passenger demand, these alternative services suffice to ensure that passengers can reach their destination without too many inconveniences. The problem with events are the additional passenger demand peaks. If this predicted amount of passengers at the station is too high, alternative services cannot provide a seat to all passengers within a certain amount of time. That results in two problems. If this happens after the ending of an event in the evening, remaining passengers might not be able to continue their journey as the train timetable ended for that day. In the other case, remaining passengers could cause nuisance on the stations possibly leading to dangerous situations. Therefore, it is important that the capacity of the alternative services is high enough to handle the outflow of visitors that travel by train.

3.3. Model implications

From the interests of the passenger operator and the rail-infra manager, the following aspects are considered in the development of the model:

- The overall passenger hindrance should be minimized, but that should not lead to an infeasible maintenance schedule considering other rail traffic.
- To guarantee feasibility for other rail traffic, corridorbook restrictions are interpreted as hard constraints
- Within the event requests, the maximum capacity of alternative services should be sufficient to able to handle the outflow of event visitors.
- Regarding the consideration of events in the scheduling of maintenance projects, only event requests are included as these form a bottleneck. Other restrictions concerning events are not included in this model.

3.4. Nomenclature

The nomenclature used is presented in Table 3.

3.5. Model description

The proposed model to address the problem is adapted from Boland et al. (2014) and an explanation of their developed model can be found in Appendix A.

Consider a railway network over a finite, discrete time horizon $\mathcal{T} = \{1, ..., T\}$ with a node set $\mathcal{N}$ representing the stations and an arc set $\mathcal{A}$ representing direct train connections. That is, if an arc exists between two stations $n_1 \in \mathcal{N}$ and $n_2 \in \mathcal{N}$, a train goes directly from station n_1 to n_2 without a stop on other stations. Also, each direct connection $a \in \mathcal{A}$ 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.

Also, given is a set of maintenance jobs $j \in \mathcal{J}$, where a job j characterized by the associated arc $a_j \in \mathcal{A}$ and processing time $p_j \in \mathbb{N}$. In this thesis, jobs represent the maintenance projects. For each maintenance job j, a start time has to be assigned 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 for 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 cancelled.

Table 3: Nomenclature

Sets	**Description**
$\mathcal{N}$	Set of stations
$\mathcal{A}$	Set of direct train connections
$\mathcal{J}$	Set of maintenance jobs
$\mathcal{J}_a$	Set of maintenance jobs on arc $a \in A$
$\mathcal{R}_{o,d}$	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
$\mathcal{T}_f$	Set of discrete time periods in which no job $j \in J$ can be started
$\mathcal{E}_t$	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
$\phi_{o,d,t}$	Passenger demand travelling from origin o to destination d in time period t
$\beta_{o,d,t}$	Share of daily passenger demand traveling in the peak moment in time period t between o and d
$\Omega_{o,d}$	Average travel time between origin o and destination d
ω_a^e	Average travel time by train on arc a
ω_a^i	Average travel time by alternative services on arc a
π_j	Processing time of job j
τ	Minimum time interval between maintenance jobs on an arc $a \in A$
$\Lambda_{s,t}$	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**
$y_{j,t}$	Binary variable indicating the starting time of job j
$x_{a,t}$	Binary variable indicating the availability of arc a in time period t
$h_{o,d,t}^k$	Binary variable indicating if route option k is used by passengers travelling from origin o to destination d
$w_{a,t}$	Travel time traversing arc a in time period t
$v_{o,d,t}$	Travel time from origin o to destination d in time period t.

It should be considered that not all combinations of maintenance 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 $\mathcal{C}$ is defined containing combinations $c = (a_1, a_2, ...) \subset \mathcal{A}$ that are forbidden to be unavailable simultaneously.

Furthermore, when a railway operator submits an event request, the operator requests a set of track segments on the railway network to be available. In these event request areas, the capacity must be sufficient to satisfy the passenger demand at that time period. It assumed that train capacity is always sufficient, but in case a maintenance job is scheduled this capacity is reduced to the maximum capacity that alternative services can provide. Set $\mathcal{E}_t$ represent the track segments at time t that are included in some event request. A track segment $s \in \mathcal{E}_t$ consists of multiple train connections $s \subset A$, since the closure of a track segment causes all relevant train connections to be replaced by alternative services. Additional assumptions made for the proposed model are listed below:

- 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 travelling between an (o, d)-pair at time t chose the same route, which is the path with the minimum travel time.

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

4. Mathematical model

The developed mathematical model will be explained in this section. Furthermore, approaches to tighten the solution space are discussed and the solution strategies are presented. The section ends with a validation of the implementation.

4.1. Objective function

The goal of the model is to minimize the total passenger hindrance. The definition of hindrance in this thesis is the increase in travel time compared to the average travel time between an origin and a destination and is shown in Equation (1):

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

Equation (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 with the passenger demand $\phi_{o,d,t}$. The total hindrance is the sum over all considered time periods.

4.2. Constraints

Constraints of the mathematical model can be divided in four categories: Passenger routing behaviour, maintenance scheduling, event request restrictions, and further restrictions on scheduling which depend on the scenario where the model is applied.

4.2.1. Maintenance scheduling

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

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

Furthermore, it is assumed that jobs cannot start in some periods, however jobs may continue in time periods that are not included in the set $\mathcal{T}_f$. Equation (3) ensures that jobs cannot start on a predefined set of time periods:

$$y_{j,t} = 0 \qquad (\forall j \in \mathcal{J}, t \in \mathcal{T}_f) \tag{3}$$

The implications of scheduling a maintenance project on the network are not included in Equation (2). Boland et al. (2013) formulated a constraint to model the availability of the arc and this constraint is adapted for this specific model. If a maintenance project is scheduled, the corresponding track segments of the project are unavailable for railway traffic. This means that a set of direct travel connections are not available if the travel connection uses the track segment. The availability of a direct travel connection a in time period t is modeled by the variable $x_{a,t}$ and its availability can be determined by Equation (4):

$$x_{at} + \sum_{t'=t-\pi_j+1}^{t} y_{jt'} \leq 1 \qquad (\forall a \in A, t \in \mathcal{T}, j \in J_a) \tag{4}$$

Due to the scheduled maintenance works, trains on unavailable track segments are replaced by alternative services impacting the travel time between the two concerned stations. These travel times are updated accordingly by Equation (5):

$$w_{at} = x_{at}\omega_a^e + (1 - x_{at})\omega_a^i \qquad (\forall a \in A, t \in \mathcal{T}) \tag{5}$$

The variables x_{at} are only bounded by Equation (4). An additional 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 Equation (6) :

$$\sum_{t \in \mathcal{T}} x_{at} = |\mathcal{T}| - \sum_{j \in J_a} \pi_j \qquad (\forall a \in \mathcal{A}) \tag{6}$$

4.2.2. Restrictions on maintenance scheduling

Dependent on the location, a set of rules is imposed on maintenance project scheduling. These rules are meant to reduce the hindrance for passenger operators, freight train operators or other stakeholders. For example, it should be prevented that two maintenance projects are scheduled simultaneously on an important passenger corridor causing that passengers should transfer to alternative buses twice on their trip. All rules, except for the time period between two maintenance projects, can be summarized in pairs of track segments that cannot both have maintenance projects scheduled simultaneously. All these combinations of arcs that are not allowed to be unavailable simultaneously are described by Equation (7).

$$\sum_{a \in c} (1 - x_{a,t}) \leq 1 \qquad (\forall c \in C_j, t \in \mathcal{T}, j \in J) \tag{7}$$

In general job scheduling problems, it is often assumed that jobs cannot overlap when processed on the same machine. This is also assumed for this thesis, although it is not necessarily true. Jobs may be combined in railway scheduling, but without loss of generality this can be observed as one job. Then, the constraint to prevent that jobs cannot overlap is given by Equation:

$$\sum_{j \in J} \sum_{t'=t-p_j-\tau}^{T} y_{j,t} \leq 1 \forall t \in T \tag{8}$$

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 cannot exceed the capacity provided by alternative services. This capacity is virtually unlimited in case no jobs are scheduled on a track segment and is limited when a job scheduled. This is due to the alternative services used, which cannot meet the same capacity offered under normal circumstances by train. Equation 9 models this behaviour:

$$\sum_{a\in s}\sum_{(o,d)\in\mathcal{N}\times\mathcal{N}}\sum_{i=1|a\in R_{o,d,i}}^{k} h^{i}_{o,d,t}\beta_{o,d,t}\phi_{o,d,t} \leq \Lambda_{s,t} + M\sum_{a\in s} x_{a,t} \qquad (\forall s\in\mathcal{E}_t, t\in\mathcal{T}) \tag{9}$$

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 right-hand side implies that there is practically no limit if no job is scheduled and there is a limit $\Lambda_{s,t}$ if a job is scheduled.

4.2.4. Passenger route choice and evaluation

In general, most passenger try to minimize their travel time and therefore often choose the shortest route regarding travel time. The shortest path in a network can be computed by for example flow conservation constraints used in the integer programming formulation of (Boland et al., 2013). A reduction in the complexity can be made by pre-computing k alternative routes for each (o, d)-pair instead of computing the shortest path for each (o, d)-pair for each considered time period t. The k alternative routes are considered to be the k shortest routes possible. For example, the algorithm of (Yen, 1971) can be used to find these shortest paths.

$$\sum_{i=1}^{k} h^{i}_{od,t} = 1 \qquad (\forall (o,d)\in N\times N, t\in\mathcal{T}) \tag{10}$$

$$v_{o,d,t} \geq \sum_{a\in R_{od_k}} w_{at} - M(1-h^{k}_{od,t}) \qquad \forall i\in[k], (o,d)\in N\times N, t\in[T] \tag{11}$$

$$v_{o,d,t} \leq \sum_{a\in R_{od_k}} w_{at} \qquad \forall i\in[k], (o,d)\in N\times N, t\in[T] \tag{12}$$

The first constraint, Equation (10), ensures that all passengers travelling from an origin o to a destination d in time period t chose exactly one and the same path. This path is the shortest path and is modeled by Equations (11) and (12). Equation (11) 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 Equation (12) as the travel time should be smaller or equal than all the travel time of the k considered paths.
The danger is that the actual shortest path is not included in one of the k alternatives, however this is not very likely as most public transport networks do not have that many effective routes that could be considered as a shortest path.

4.3. Scheduling model considering passenger hindrance and event requests

A complete overview of the mathematical model is presented by the Equations 13a - 13s.

$$\min \quad \sum_{(o,d)\in\mathcal{N}\times\mathcal{N}} \sum_{t=1}^{T} \phi_{o,d,t}(v_{o,d,t} - \Omega_{o,d}) \tag{13a}$$

$$\sum_{t=0}^{T-\pi_j+1} y_{j,t} = 1 \qquad (\forall j \in \mathcal{J}) \tag{13b}$$

$$y_{j,t} = 0 \qquad (\forall j \in \mathcal{J}, t \in \mathcal{T}_f) \tag{13c}$$

$$x_{at} + \sum_{t'=t-\pi_j+1}^{t} y_{jt'} \leq 1 \qquad (\forall a \in \mathcal{A}, t \in \mathcal{T}, j \in \mathcal{J}_a) \tag{13d}$$

$$\sum_{t\in\mathcal{T}} x_{at} = |\mathcal{T}| - \sum_{j\in J_a} \pi_j \qquad (\forall a \in \mathcal{A}) \tag{13e}$$

$$w_{at} = x_{at}\omega_a^e + (1 - x_{at})\omega_a^i \qquad (\forall a \in \mathcal{A}, t \in \mathcal{T}) \tag{13f}$$

$$\sum_{a\in c}(1 - x_{a,t}) \leq 1 \qquad (\forall c \in \mathcal{C}, t \in \mathcal{T}, j \in \mathcal{J}) \tag{13g}$$

$$\sum_{j\in J} \sum_{t'=t-p_j-\tau}^{T} y_{j,t} \leq 1 \forall t \in T \tag{13h}$$

$$\sum_{a\in s} \sum_{(o,d)\in\mathcal{N}\times\mathcal{N}} \sum_{i=1|a\in R_{o,d,i}}^{k} h^i_{o,d,t}\beta_{o,d,t}\phi_{o,d,t} \leq \Lambda_{s,t} + \sum_{a\in s} x_{a,t}M \qquad (\forall s \in \mathcal{E}_t, t \in \mathcal{T}) \tag{13i}$$

$$\sum_{i=1}^{k} h^i_{od,t} = 1 \qquad (\forall (o,d) \in \mathcal{N}\times\mathcal{N}, t \in \mathcal{T}) \tag{13j}$$

$$v_t^{od} \geq \sum_{a\in R_{od_k}} w_{at} - M(1 - h^k_{od,t}) \qquad (\forall i \in [k], (o,d) \in \mathcal{N}\times\mathcal{N}, t \in \mathcal{T}) \tag{13k}$$

$$v_t^{od} \leq \sum_{a\in R_{od_k}} w_{at} \qquad (\forall i \in [k], (o,d) \in \mathcal{N}\times\mathcal{N}, t \in \mathcal{T}) \tag{13l}$$

$$y_{j,t} \in \{0,1\} \qquad (\forall j \in \mathcal{J}, t \in \mathcal{T}) \tag{13m}$$

$$x_{a,t} \in \{0,1\} \qquad (\forall a \in \mathcal{A}, t \in \mathcal{T}) \tag{13n}$$

$$z_{a,t} \in \{0,1\} \qquad (\forall a \in \mathcal{A}_t, t \in \mathcal{T}) \tag{13o}$$

$$\tag{13p}$$

$$h^i_{o,d,t} \in \{0,1\} \qquad (\forall i \in [k], (o,d) \in \mathcal{N}\times\mathcal{N}, t \in \mathcal{T}) \tag{13q}$$

$$w_{a,t} \in \mathbb{R} \qquad (\forall a \in \mathcal{A}, t \in \mathcal{T}) \tag{13r}$$

$$v_{o,d,t} \in \mathbb{R} \qquad (\forall (o,d) \in \mathcal{N}\times\mathcal{N}, t \in \mathcal{T}) \tag{13s}$$

The objective function in Equation (13a) minimizes the passenger delays and the conflict value. Equation (13b) - Equation (13f) ensures that the jobs are scheduled exactly once, in the allowed time periods and models effects on the network, i.e. the unavailability of arcs and travel times due to maintenance jobs. Equation (13g) and Equation (13h) ensure that the corridor-book restrictions (combinations and minimum time interval) are respected, while Equation (13i) models the passenger flow tolerance within event request areas. Equations (13j) - (13l) model the passenger route choice and corresponding travel times from an origin to a destination. The remaining constraints (Equations (13m) - (13s)) model the type of the considered variables in the model.

4.4. Solution space pruning

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 losing any feasible solution. Finally, a meta-heuristic is considered to find a reasonably good solutions as starting point for branch-and-bound methods. The effectiveness of the latter two methods are tested in Section 5.1.

4.4.1. Parameter analysis

Analysis of parameter values could sometimes lead to reduction of the solution space. In some cases, variables could take a certain value that will never be included in any feasible solution. These cases will be excluded to change 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 implies that the status (available/unavailable) and the travel time on the concerned arc are already fixed:

$$x_{a,t} = 1 \qquad \forall a \in \{a \in A | J_a = \emptyset\}, \forall t \in \mathcal{T} \tag{14}$$

$$w_{a,t} = \omega^e_{a,t} \qquad \forall a \in \{a \in A | J_a = \emptyset\}, \forall t \in \mathcal{T} \tag{15}$$

Consider the k possible shortest routes to 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 a infeasible solution otherwise. This is contained in Equation 16:

$$x_{a,t} = 1 \quad \forall (a,t) \in \{(a,t) \in \mathcal{A} \times \mathcal{T} | \beta_{o,d,t}\phi_{o,d,t} > \Lambda_{s,t}, a \in R_{o,d,i} \forall i \in \{1, ..., k\}, \exists s \in \mathcal{E}_t \text{ s.t. } a \in s\} \tag{16}$$

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

Furthermore, the generation of k- shortest paths for an (o, d) - pair could still lead to redundant options in route choice. In this model, the maximum travel time of a single path is if all jobs related to that path are processed simultaneously and the minimum travel time of a path is logically achieved when no jobs are scheduled. Now consider a path p_1, if the minimum travel time of that path is higher than the maximum travel time of path p_2, path p_1 will obviously never be chosen. Since the shortest path is always chosen by passengers, selecting path p_1 would yield an infeasible solution. This is mathematically described in Equation 17:

$$h_{o,d,t,i} = 0 \ \forall (o,d,i) \in \{(o,d,i) \in \mathcal{N} \times \mathcal{N} \times k | \min_{i \in \{1,...,k\}} \{\Omega_{o,d,i}\} > \max_{j \neq i \in \{1,...,k\}} \{\Omega_{o,d,j}\} \forall (o,d) \in \mathcal{N} \times \mathcal{N}\} \tag{17}$$

4.4.2. 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 effectively describe a solution space and 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 scheduling problem in this thesis 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 overlap of jobs. 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 as rules of the corridorbook.

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

$$\sum_{s \in Q_t} y_{j,t} + \sum_{l \neq j} \sum_{s \in Q_l} y_{l,t} \leq 1 \qquad \forall a \in \mathcal{A}, j \in \mathcal{J}_a, \forall t \in \mathcal{T}. \tag{18}$$

The valid inequality in Equation 18 is facet-defining under the condition that $T \geq \sum_{j \in J_a} p_j + 3\bar{p}$ and $p_l + \bar{p} < t \leq T - \bar{p}$ for a job l and time $t \in \mathcal{T}$. This means that the inequality is under these circumstances not redundant. For the proofs of both the valid inequality and the facet-defining property, the reader is referred to Sousa and Wolsey (1992).

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

$$y_{j,t} + y_{j',t'} \leq 1 \qquad \forall a \in \mathcal{A}, (j,t), (j',t') \in \mathcal{J}_a \times \mathcal{T} : j \neq j, t' \in (t - \psi_{j'} - \tau, t + \psi_j + \tau) \tag{19}$$

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 thesis.

4.4.3. Initial solutions

Meta-heuristics are able to improve the computational costs by finding in a relatively short amount of time a good solution. First, it can be used as a starting point for generating an initial solution and second, this helps exact methods as branch-and-bound algorithms to cut off solutions by providing an upper bound on the objective function value. For this, a constructive heuristic is proposed to find a feasible solution. This solution is improved by means of a meta-heuristic, simulated annealing. The method is described in Appendix C.

4.5. 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^{TM} i5-7200U processor and 8GB RAM is used to run the models. The Gurobi solver utilizes advanced algorithms for Mixed Integer Linear Programs. The optimization algorithm is 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. The idea of a pre-solving method is to tighten the formulation and reduce the problem size. The cutting planes are used during the optimization process by removing undesirable fractional solutions. As it is not always possible to prove optimality, heuristics are used to create reasonably good feasible solutions and also to generate upper bounds for the branch-and-bound algorithm. Finally, parallelism is the idea of using multiple cores to process multiple nodes in a branch and bound tree simultaneously. (Gurobi, a)

In total three models will be tested. Model 1 represents the naive branch-and-bound algorithm of Gurobi. That is, all features as the generation of cutting planes, presolving methods and initial solutions are disabled. Model 2 is an extension of the first model with the only addition that the proposed simulated annealing algorithm is used to generate an initial solution and provide an upper bound for the branch-and-bound algorithm. The final model, Model 3, extends Model 2 by including the proposed valid inequalities and cutting plane generators of Gurobi. Gurobi is a private company and therefore, detailed information about cutting plane generators is not available. An overview of cutting plane generators is available (Gurobi, b) and shows the following identifiable cuts:

- 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)

Note that there are more cutting plane generators included in the software, but no literature could be found for these cuts and are not considered in this thesis.

4.6. Model validation

All models are implemented in Python 3.7.4 and solved with the Gurobi solver. An instance is created to validate the considered models. A visualization of the instance is shown in Figure 2. The code of the models and the data of the validation can be found in an online Mendeley dataset[1].

The network is shown as a directed graph to clarify that edges can be traversed in both directions and for this instance it holds that travel times are symmetric, i.e. equal in in both directions. Note that this is not necessarily the case when the model is applied for other directed networks.

A time horizon $T = 10$ is considered. Furthermore, for each (o, d)- pair, $k = 3$ optional paths are considered. The weight representing the travel time on an arc is independent of the travel direction on the concerned arc. For this purpose, conflicts are not allowed , which actually implies that the bus capacity in the event request areas is 0 and hence, $\Lambda_{s,t} = 0 \ \forall s \in \mathcal{E}_t, \ t \in \mathcal{T}$. An overview of other input parameters for the toy instance is shown in Tables 4 - 8.

[1]https://data.mendeley.com/datasets/5x9kjg6mxx/draft?a=3eb39bde-016d-4674-ba99-eb0de1510813

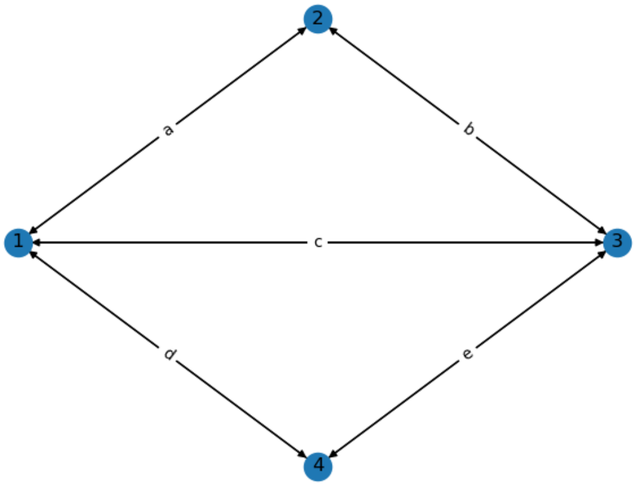

Figure 2: Visualization of the test network

Table 4: Arcs: Expected travel time and delayed travel time.

A	ω^e	ω^i
a	5	9
b	4	7
c	7	12
d	6	10
e	4	7

Table 5: Job locations and processing times.

J	A	p
1	a	3
2	c	2
3	e	2

Table 6: Event requests: Indicated arcs, start and end time.

E	A	T_{start}	T_{end}
1	$\{a, b\}$	2	5

Table 7: Restriction from the corridor book.

C	$\{c, e\}$
t_{int}	2

Table 8: Passenger demand, considered paths for each (o, d) - pair in the toy network. The last two columns represent the expected travel time and number of transfers without any delay due to maintenance.

OD	ϕ	$R_{o,d,1}$	$R_{o,d,2}$	$R_{o,d,3}$	$\Omega_{o,d}$
$1 \to 3$	50	$1 \to 3$	$1 \to 2 \to 3$	$1 \to 4 \to 3$	7
$3 \to 1$	50	$3 \to 1$	$3 \to 2 \to 1$	$3 \to 4 \to 1$	7
$1 \to 2$	50	$2 \to 1$	$1 \to 3 \to 2$	$1 \to 4 \to 3 \to 2$	5

When running the implemented model, it can be observed that the optimal solution is not unique. In fact, many different optimal solutions are possible and for this toy network the limit was set on five alternative solutions. Therefore, the first given solution is considered to show that this is solution is optimal for the instance.

The objective function value of the given solution is 1000 and the solution is shown in Tables 9 and 10. The purpose of the validation is to prove the optimality and correctness of the solution. The supporting variables $x_{a,t}$ and $w_{a,t}$ are not discussed as these are a result of the scheduling variable $y_{j,t}$.

First, feasibility implies that jobs are scheduled exactly once, the capacity threshold in event request areas are not exceeded and restrictions from the corridorbook are respected. The first requirement clearly suffices, Table 9 shows the resulting start and end times of the jobs. The second requirement implies that ith a bus capacity of $\Lambda_e = 0$ in event request areas, no passengers can travel via arc a or arc b when there is a job scheduled. This solely concerns job 1, which is scheduled to start at time 6, after the event is finished.

For each of the (o, d)-pairs, the path minimizing the travel time in the network is presented

Table 9: Start and end times as a result of the job scheduling variable y

J	Start	End
1	6	8
2	1	8
3	4	2

Table 10: Results of the validation instance. $k \in \{1,2,3\}$ shows the chosen route option and $v_{o,d,t}$ the corresponding travel time for each (o,d) pair at a time period t.

t	1		2		3		4		5	
OD	k	$v_{o,d,t}$	k	$v_{o,d,t}$	k	$v_{o,d,t}$	k	$v_{o,d,t}$	k	$v_{o,d,t}$
(1,3)	2	9	2	9	1	7	1	7	1	7
(3,1)	2	9	2	9	1	7	1	7	1	7
(1,2)	1	5	1	5	1	5	1	5	1	5

t	6		7		8		9		10	
OD	k	$v_{o,d,t}$	k	$v_{o,d,t}$	k	$v_{o,d,t}$	k	$v_{o,d,t}$	k	$v_{o,d,t}$
(1,3)	1	7	1	12	1	12	1	7	1	7
(3,1)	1	7	1	12	1	12	1	7	1	7
(1,2)	1	9	1	9	1	9	1	5	1	5

by sh_1, so naturally this path will be chosen by passengers without any maintenance jobs on this route. Table 10 shows that this is true for all (o,d) pairs.

It remains to show what path passengers choose if maintenance is scheduled on their preferred route. Consider the passengers travelling between nodes 1 and 3, they choose option 1 as their preferred route and during the maintenance period on arc $(1,3)$, travel time increased from 7 to 12 leading to a change in their preferred path. This is now path option 2 that goes via node 2 with a corresponding travel time of 9.

Passengers travelling from node 1 to node 2 always choose their first option, even during maintenance periods. This can be explained by the fact that the increased travel time their preferred route, which is the arc $(1,2)$, is 9 and has still a lower travel time than the other considered alternative paths. The alternatives 2 and 3 have a travel time of 11 and 14, respectively.

Finally, the result of the objective function is the sum of all travel time delay for all passengers. For the (o,d)-pair $(1,3)$, there are 2 time periods in which maintenance is carried out on the shortest path, resulting in a delay of 2 time units per passenger per time period. The total hindrance calculated for this (o,d)-pair is $2 \cdot 50 \cdot 2 = 200$ Since travel times do not differ when traversing the edge in the other direction, this calculation is equivalent for $(o,d) = (3,1)$. The remaining (o,d)-pair is $(1,2)$. The hindrance over this time horizon equals $3 \cdot 50 \cdot 4 = 600$, because of a delay of 4 time units in three time periods times the passenger demand in these time periods, resulting in a total hindrance of 600 time units. This leads to an optimal objective function value of $200 + 200 + 600 = 1000$, equal to the result of objective function value of the model.

5. Numerical experiments

This section consists of two parts. The first part is a computational study on toy instances to evaluate the performance of the model. The second part of this chapter is a case study in which the model will be applied to a part of the Dutch railway network to test the model on practical instances.

5.1. Computational study on toy instances

Toy instances are used to gain insight into the performance of the valid inequalities and heuristic. The characteristics of the toy instances are discussed first and then the computational results from the models on the toy instances follows. Finally, limitations of this computational study and further testing directions are presented. The models, toy instances and results of the models applied on the toy instances can be found in the Mendely dataset.

5.1.1. Description of toy instances

A total of 27 toy instances have been generated to test the computational performance of the model. These instances are characterised by the size of the network, number of jobs and the length of the time horizon. The other sets and parameters are randomly generated or not included in the computational study. These sets are related to restrictions on maintenance scheduling and do not add significant value to increasing problem sizes as these are problem-specific. The full explanation of the toy instance generation is explained in Appendix E. 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 big such that differences between set sizes might be easier to observe.

An overview of the toy instances is shown in Table 11. The table shows the set sizes of all sets.

5.1.2. Computational times for the toy instances

The impact of set sizes on computational times is analyzed using the reference model, Model 1. The results of Model 1 on the toy instances are shown in Table 12.

At first sight, it seems that there is no clear relation in the results for different network sizes. For a fixed network size, there is a clear increase in the gap or in the run time only if the number of jobs or the time horizon is increased. This indicates that the latter two sets have more impact on the run times. The run times, however, are slightly increasing for a fixed number of jobs and time horizon. This can primarily be seen for the instances that are optimally solved. This means that there is a minor impact on the computational times of the model.

Now consider a fixed network size. The general trend in the table is that the model is harder to solve with an increase in the number of jobs or the length of the time horizon. Instances with a small number of jobs or short time horizon are generally solved to optimality and increasing one of the set sizes shows memory errors or solutions that are not proven to be optimal.

The instances with a network size of 10 (instance 1 -9) illustrate this trend perfectly. If the number of jobs increases from 10 to 40, there is an explosion in the run time of the model. With 10 jobs, it takes around 100 seconds to run a model, while this takes goes up to at least 5300 seconds with 40 or more jobs.

To observe the effect of the time horizon, triples of instances can be considered that have a fixed number of jobs and a fixed network size (e.g. instances 1, 2 and 3 are one triple). It shows that there is an increase in the run time or in the gap if the run time limit has been reached. This is a clear trend and the differences within the triples get bigger if the network size and the number of jobs increase.

The findings on changes in set sizes for Model 1 are consistent with the results of Model 2 and Model 3, which are shown in Tables 13 and 14. These models also give more information

Table 11: Cardinality of sets for each toy network

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

on the computational performance, as these models were able to generate a feasible solution for each toy instance.

The additional information from the models indicates that an increasing network size might lead to an easier problem to solve. There are three instances solved optimally with a network of 10 nodes, four instances solved optimally with 20 nodes and five instances with an optimal solution for instances 5. Even when there is no optimal solution, the gap is in most cases smaller when the network size increases.

Furthermore, there are some inconsistencies that do not hold on to the trends explained before. This might be explained by the generation of the instances. The average number of jobs on a single arc decreases if the network size gets bigger as there are more arcs to locate a job. It increases the probability that two jobs are independent, decreasing the complexity of the problem.

5.1.3. Computational differences between the models

Tables 12 - 14 show the results of Model 1, Model 2 and Model 3 applied on the toy instances. Each table shows for a specific model the computational times , 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 runtime of the meta-heuristic and the branch-and-bound algorithm. The total runtime is the total running time of the script, i.e. the heuristic time, the branch-and-bound time and some remaining time to compute some small sets that are not in saved in the dataset of the instance.

The MIP gap is a percentual value indicating the differences in objective value between the solution of the relaxation of the model and the best found solution. The Gurobi solver declares optimality if the integrality gap is smaller than a predefined threshold, which is set to the default setting of 0.01%.

In order to compare the performance of the models, the computational time is set to 7200 seconds for the branch and bound algorithm. The simulated annealing algorithm used in Models 2 and 3 terminates if the the time limit of 1800 seconds is reached or if the temperature reaches a value lower than 1. The initial temperature is set to 500000. The progress of the branch-and-bound method is stored in log files and can be found in the online database.

Table 12: Computational results for model 1 tested on the toy instances

Instance ID	**Gap**	**Runtime (total)**	**Runtime Heuristic**	**Runtime Branch and bound**	**Nodes explored**
1	0.00	2.35	0	1.24	3
2	0.00	48.30	0	42.59	22
3	0.00	112.67	0	104.86	69
4	0.01	5346.79	0	5345.36	410484
5	12.57	7202.77	0	7200.04	190348
6	-	-	0	7200.03	19964
7	338.80	7201.00	0	7200.01	1048096
8	-	-	0	7200.04	18008
9	-	-	0	7200.02	5906
10	0.00	17.54	0	13.89	36
11	0.00	131.04	0	116.42	517
12	0.00	711.07	0	664.22	3
13	0.00	27.23	0	24.35	539
14	20.81	7213.58	0	7200.13	46094
15	292.19	7226.68	0	7200.14	19593
16	0.37	7203.66	0	7200.06	111822
17	616.88	7222.70	0	7200.07	42705
18	-	-	0	7200.34	370
19	0.00	229.73	0	202.32	3
20	0.00	1629.12	0	1503.09	3
21	-	-	0	737.16*	0
22	0.00	249.66	0	225.24	16
23	0.00	3224.38	0	3153.30	1679
24	-	-	0	5724.02*	0
25	0.00	3766.18	0	3741.59	5142
26	-	-	0	8182.17	0
27	-	-	0	7200.89	0

* branch and bound terminated due to memory error

Table 12 shows the results of Model 1. The model is able to find a solution for most instances in the 7200 seconds. In most cases, the gap is close to zero and implies that an optimal solution has been found within the predefined tolerance. In the remaining cases, the gap is over the 100 % indicating that no there are significant differences with the solution from the relaxation of the model. This does not necessarily imply that the optimal solution has not been found, but it is not proven to be optimal.

There are still some instances that could not be solved by the model, even no feasible solution has been found. Several causes could explain this. The first one is that the branch-and-bound algorithm did not yield any feasible solution within 7200 seconds. This is the case when there are more than 40 jobs included in the model. Another cause is the size of the instance. Instances 21 and 24 did not yield any solution due to a memory error. This means that the instance is relatively large and therefore, the computational burden in terms of memory is high. Note that instances 26 and 27 contains larger sets than instances 21 and 24 and the model did not produce a memory error. The procedure of the branch-and-bound algorithm might have a big impact on this.

Table 13: Computational results for model 2 tested on the toy instances

Instance ID	**Gap**	**Runtime (total)**	**Runtime Heuristic**	**Runtime Branch and bound**	**Nodes explored**
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	1368164
5	3.48	7281.20	77.16	7200.74	153436
6	125.52	7343.21	135.34	7202.08	85213
7	564.36	7228.68	28.04	7200.02	1027013
8	150.80	7286.01	83.06	7200.04	139374
9	200.46	7355.95	149.74	7200.23	85621
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	54381
15	23.33	7957.17	732.79	7200.22	19463
16	2.74	7297.72	95.11	7200.03	219454
17	113.36	7590.56	378.51	7200.09	45059
18	68.73	8060.58	833.65	7200.11	10896
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	36863
26	39.69	9084.62	1808.09	7201.34	2155
27	909.46	9176.80	1815.88	7202.66	138

Tables 13 and 14 present the results for Model 2 and Model 3, 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 model type. Both models use an initial heuristic to obtain a relatively good, feasible solution as input for the branch-and-bound method.

Model 2 and Model 3 achieve roughly the same results on the gap. There is, however, one exception. Interestingly, the results on instance 18 are significantly different. Model 2 reached a gap of 68.73% and Model 3 a gap of 513.60 %. This differences and as well the smaller differences in the gap between the model could be caused by the initial heuristic. The heuristic is partially random and could therefore result in different solution qualities.

On the instances that are solved optimally, Model 2 has a better computational time on almost all instances compared to Model 3. This is surprising as the addition of tight valid inequalities should reduce computational times. On the other hand, adding more inequalities increases the computational burden and that might be the case here.

From the instances that are not solved optimally by Models 2 and 3, it can be seen that the number of nodes explored is significantly lower for Model 3. The results on the gap are roughly equal and hence, iterations using Model 3 are more effective with the valid inequalities. This comes at a cost, because a single iteration takes more time.

Table 14: Computational results for model 3 tested on the toy instances

Instance ID	**Gap**	**Runtime (total)**	**Runtime 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	107196
6	127.98	7382.72	155.33	7200.11	19947
7	583.99	7235.58	34.73	7200.02	850028
8	150.53	7319.31	105.51	7200.13	27461
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	46717
15	23.81	8062.17	820.92	7200.19	9927
16	3.24	7446.41	241.47	7200.03	149405
17	114.38	7650.51	432.36	7200.09	29068
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	28879
26	39.70	9084.82	1805.72	7200.50	2241
27	912.85	9229.46	1811.04	7241.88	1

Generally, results show that Model 2 and Model 3 outperform Model 1. The gap is generally reduces with Model 2 and 3 and if optimally solved, the run time is generally lower as well compared to Model 1. There are two exceptions. On instance 7, Models 2 and 3 both perform worse than Model 1. Interestingly, the gap difference is almost 200 % and the only difference between Model 2 and Model 1 is the initial heuristic. Therefore, the expectation is that Model 2 would always outperform or achieves similar results as Model 1. The other exception is that Model 1 is able to solve instance 4 optimally, while the 2 other models cannot solve the problem optimally and reach a gap around the 30 %. Again, this is not expected as Model 2 is an extension of Model 1.

On relatively big instances, the used meta-heuristic is able to reduce the computational burden. It is able to prune the solution space, causing the branch-and-bound algorithm to be more effective. The heuristic, however, does not seem necessary for the smaller instances. For instances 1, 2 and 3 for example, the total run time is lower for Model 1 compared to the other two models.

The speed of the heuristic also decreases for the bigger instances. For all instances except instances 7-10, the heuristic was terminated because the temperature reached a value below 1. For the other instances, the time limit was reached. This might impact the quality of the heuristic solution as there is relatively a lot of time spent on the exploration of the solution space.

It could be stated that the additional heuristic time (that is not included in the 7200 second) caused the difference in the gap between Model 1 and the other two models. This is because Models 2 and 3 had more time to find better solutions for the instances that are not optimally solved. This can be supported by displaying the development of the gap. Figure 3 shows this MIP gap development for the three models applied on toy instance 1.

After 140 seconds in the branch-and-bound algorithm (which is 220 seconds including the heuristic time) Model 2 reached a gap of 5% and shows a better performance than Model 1 within this time period. Model 3 reached the gap of 5 % after 276 seconds in the branch and bound algorithm also showing a better performance than Model 1. The development of the MIP gap for the models on the remaining instances show equivalent results and figures can be found in the online Mendeley database.

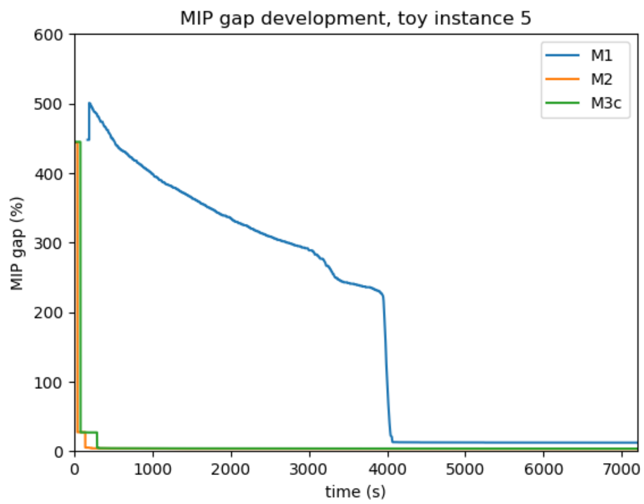

Figure 3: MIP gap development for the model variants on toy instance 1

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

- The heuristic has a significant impact on the computational costs for the medium-sized instances.
- The valid inequalities and cutting planes tighten the solution space. As a result, the branch-and-bound explores less nodes to achieve the same results, although it did not result in better solutions within the considered time period.
- Optimal solutions are only found for relatively small instances within 7200 seconds. This indicates that, for practical applications, exact methods do not suffice.

5.1.4. Limitations and further testing directions

Further testing directions should examine the parameter settings and local search methods of the simulated annealing algorithm used in the Models 2 & 3. The cooling scheme and the initial temperature are based on commonly used settings and not optimized for the purpose of this thesis. The optimization of the cooling scheme and initial temperature could lead to a better objective value and therefore reduce the size of the tree generated in the branch-and-bound algorithm.
Also, it could be tested if the heuristic could be effective for smaller instances. With the current settings, the heuristic reduces the performance on instances 2,3 and 7. This could possibly changed by reducing the initial temperature or time limit.

The run time limit was set to 7200 seconds and different time limits could reveal new information. For some instances as for example instance 5, it can be seen that the gap quickly decreases and only minor improvements are made over the remaining period. Other instances need more time and 7200 seconds of runtime do not suffice. Note that the run time can be extended since the scheduling is on a tactical level.
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 variable implying that within the considered two hours the number of explored nodes differs between models. This could give more information regarding the trade-off between the tightness of the solution space by adding more constraints and the run 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 if 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 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.

5.2. Case study

In the remainder of this section, a case study is presented on a part of the Dutch railway network. First, the case study area will be explained in detail, then results of the model will be presented and analyzed.

5.2.1. Case study area description

The model will be tested in a case study around the area of Utrecht, the Netherlands. Utrecht is a city located the centre of the Netherlands and due to this central location, the main station of Utrecht, Utrecht Centraal, is one of the most important of stations of the

Netherlands. In 2019, on a daily bases around 207.000 passengers have the station Utrecht Centraal as origin or destination for their journey. Another 65.000 passenger use this station as a transfer station. This makes it the station with the most daily travellers in the Netherlands, followed by Amsterdam Centraal. (NS, 2019)
The scheduling of maintenance projects is examined that affect the track segments around the station Utrecht Centraal in the year of 2023. The considered time period is limited to the second quartile of that year, from April 1st to June 30th. In this section, the railway network for the case study is described, a description of the maintenance project schedule in the considered time period is given and the event requests in that period are described.

Infrastructure

The infrastructure for the case study is shown in Figure 4. This infrastructure shows the track segments that are the underlying structure for the direct travel connections of trains. The case study area contains 109 stations and a total of 290 direct travel connections. The case study area is bounded by important nodes in the railway network as the station of Zwolle, Eindhoven or Amsterdam Centraal. In this area, the only passenger operator on these tracks is NS. There are some international trains passing through the case study area, but these are not included in the case study.

Planned Special Events

A total of 19 unique event requests are submitted by the passenger operator for the considered period and should be taken into account while scheduling of the projects around Utrecht Centraal. The events corresponding to the event requests may be hosted at any place in the Netherlands and are included in the case study if the event request contains track segments that are directly linked to the station Utrecht Centraal. An overview of all events with the relevant characteristics are shown in Table 15.

Table 15: Events included in this case study

Event ID	Event name	Startdate	Enddate	Event Station	Expected visitors/day
37	Marathon Rotterdam	April 2	April 2	Rotterdam Centraal	950000
40	Paaspop	April 7	April 10	's-Hertogenbosch	35000
41	Paasraces Zandvoort	April 8	April 10	Amsterdam Centraal	105000
42	Bloesemtocht Geldermalsen	April 15	April 15	Geldermalsen	35000
50	Bekerfinale KNVB	April 30	April 30	Rotterdam Centraal	47500
52	Libelle Zomerweek	May 9	May 21	Haarlem	11500
53	Lakedance	May 13	May 14	Eindhoven	40000
55	Marikenloop	May 21	May 21	Nijmegen	15000
59	Emporium	May 27	May 29	Nijmegen	30000
60	Pinksterraces Zandvoort	May 27	May 29	Haarlem	105000
67	Concert Goffert I	June 9	June 9	Nijmegen	50000
69	Concert Goffert II	June 10	June 10	Nijmegen	50000
70	Parkpop Zuiderpark Den Haag	June 11	June 11	Den Haag Centraal	200000
71	Concert Goffert III	June 11	June 11	Nijmegen	50000
77	Concert Goffert III	June 11	June 11	Nijmegen	50000
81	Veteranendag	June 24	June 24	Den Haag Centraal	75000
82	Concert Goffert V	June 24	June 24	Nijmegen	50000
84	Concert Goffert X	June 25	June 25	Nijmegen	50000
88	Rijnweek	June 30	July 7	Ede-Wageningen	100000

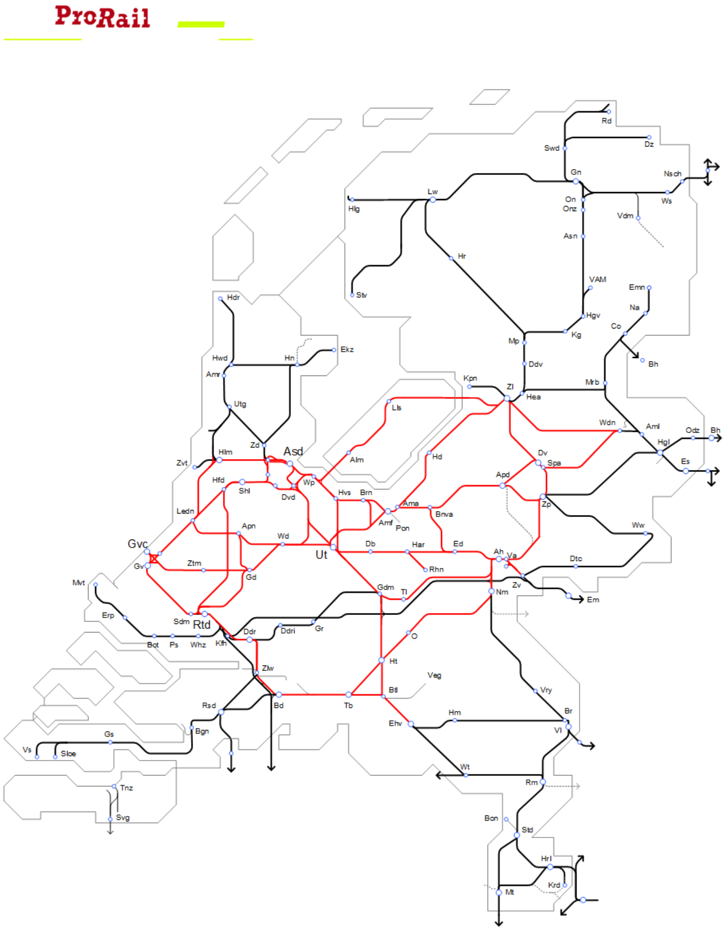

Figure 4: Network for the case study

Current maintenance schedule

A total of 5 maintenance projects are scheduled in the period between April 1 to June 30 that affect one or more track segments adjacent to the central station of Utrecht. The duration of the projects differs in duration with a maximum of 52:00 hours in this dataset. This equals two days including an additionally night, but as it does not affect the schedule of the next day, this duration is rounded to the nearest number of days. An overview of the maintenance projects is shown in Table 15. There is one conflict in the current schedule, which is at May 13. At that day, the Libelle Zomerweek is ongoing and an operator submitted an event request for this event, but project 358 is scheduled at a subset of the track segments that was included in the event request.

5.2.2. Model input

The model has $|\mathcal{N}| = 109$ stations and $|\mathcal{A}| = 209$ direct travel connections. A period of $|T| = 90$ is considered. Furthermore, there are $k = 3$ shortest paths included for each (o, d)-pair. These are determined using the modified Yen's algorithm described in Appendix B. Table 17 summarizes the cardinality of all main sets in this thesis.

Table 16: Maintenance projects

J	**Startdate**	**Enddate**	**Duration (hours)**
249	April 7	April 8	52:00
289	April 9	April 10	52:00
358	May 13	May 13	28:00
360	May 13	May 14	52:00
466	June 17	June 18	52:00

Table 17: Cardinality of input sets.

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

5.2.3. Spatial and temporal characteristics

The stations are represented by nodes and the arcs between nodes represent the direct travel connections by train. These arcs are directed to ensure that passengers flows are also directed. If undirected arcs are used in the case study, combining passenger flows in both directions would result in false violations of the capacity constraint (Equation (13i)). Each arc is characterised by the adjacent nodes, travel time by train and the travel time using alternative services.

The time periods are considered to be days, where the first day ($t = 1$) represents April 1, 2023. Furthermore, no jobs can be started on weekdays, although they might continue on weekdays. These dates are contained in the set T_f.

Network entities

Maintenance projects and event requests are the network entities for the case study. Both maintenance projects and event requests are characterised by a set of arcs in the network.

The maintenance projects scheduled in second quartile of 2023 presented in Table 16 and the requests for the events in Table 15 are used as input for the model.

For both maintenance projects and event requests, the data contains a set of stations that is affected by the entity. This can be converted to train connections by the line connections that exist in the database for each pair of stations. Specifically for the event request, the bus capacity is not known in case there is maintenance and this depends on the location, date and time. Since there are no concrete values for $\Lambda_{s,t}$ is constant for all $s \in \mathcal{E}_t$, $\forall t \in \mathcal{T}$, the range is set from 0 to 10000.

Passenger characteristics

Daily passenger demand data is located in a dataset that contains predictions of the passenger demand from the year 2020 to 2022. The dataset contains pairs of stations as (o, d)-pairs with an average daily passenger demand travelling from the origin to the destination.

This daily average for an (o, d)-pair can be used to find an estimate of the passenger demand for a certain hour on a certain date by multiplying the demand with a factor at a specific day and hour. The factor is between 0 and 1 and represents the share of the daily passenger demand on the concerned hour. The sum of factors per hour on a day equals one. It should be noted that this factor is not location or (o, d)-pair specific, i.e. this factor is the same for all OD pairs for a given day and hour.

This data shows estimates of the regular flows on a hourly basis, but due to the location-independent factor the additional passenger demand induced by an event cannot be recognized from this data set. It is assumed that the induced additional passenger demand by the event is 30 % of the total visitors. The visitors are distributed over the station based on the average number of daily travellers using this station for boarding, independent of the distance. This is because most events attract visitors based on their interest and distance becomes less important. The distribution is modelled by means of a gravity model and is described in Appendix D.

For each origin-destination pair, a total of $k = 3$ alternative travel routes are considered. The routing of the passenger is determined via a k-shortest path algorithm adjusted for the considered network. The algorithm to determine the shortest paths is described in detail in Appendix B.

Finally, the model considers every pair of stations as an (o, d) pair and results in a total of 11772 pairs. As this is a large number of (o, d) pairs, this cannot be solved to optimality by the model within reasonable time. To limit this, a threshold was set of 100 on the daily passenger demand. Every (o, d) pair with a threshold less than that value is excluded from the model. This resulted in a total of 534 pairs considered in the model. This results in a minor decrease of the accuracy of the results, although this is acceptable as perfect passenger demand predictions for events are nearly impossible due to the high variety of events and visitors.

Corridorbook restrictions

There are some specific restrictions in the Netherlands to schedule maintenance projects. These ensure that maintenance facilities and diverting routes for freight trains and passenger operators are accessible on the tracks. (Bekke, 2021) These restrictions are expressed in pairs of subcorridors (a set of adjacent track segments) that can not be blocked simultaneously. These combinations are converted to combinations of direct travel connections as input for the model. This is eventually all combined into one matrix that contains all conflicts.

Furthermore, the minimum time interval is 25 days to have maintenance projects on a junction or track segment.

5.2.4. Key Performance Indicators

Key Performance Indicators (KPIs) are defined to evaluate the quality of maintenance project schedules as a result of the model. The focus is especially on event requests, where it is important to observe which conflicts arise given a schedule and the corresponding conflict value. This is represented with two KPIs, the number of conflicts and the conflict value. The definition of a conflict is when a job is scheduled in an event request area, which can happen at most one time per event. The conflict value represents the number of passengers that should travel with alternative service on a track segment included in an event request.

Furthermore, analysis of the distribution of passenger hindrance provides insight in distribution of hindrance among passengers. Therefore, the total number of delayed passengers and the average delay of these passengers is included.
Finally, the desire while scheduling maintenance projects is to keep the additional travel time below 30 minutes and therefore this will be used as a KPI.

The KPIs are summarized below and if needed, a mathematical formulation is given:

1. Number of conflicts
2. Conflict value:

   $\sum_{(o,d,t)\in\mathcal{Z}} \beta_{o,d,t}\phi_{o,d,t}$
3. Mean additional travel time per affected passenger:

   $$\frac{\sum_{(o,d)\in\mathcal{N}\times\mathcal{N}}\sum_{t\in\mathcal{T}}\phi_{o,d,t}(v_{o,d,t}-\Omega_{o,d})}{\sum_{(o,d,t)\in\{\mathcal{N}\times\mathcal{N}\times\mathcal{T}|v_{o,d,t}-\Omega_{o,d}>0\}}\phi_{o,d,t}}$$
4. Total amount of affected passengers:

   $\sum_{(o,d,t)\in\{\mathcal{N}\times\mathcal{N}\times\mathcal{T}|v_{o,d,t}-\Omega_{o,d}>0\}}\phi_{o,d,t}$
5. Share of affected passenger that have an additional travel time higher than 30 minutes:

   $$\frac{\sum_{(o,d,t)\in\{\mathcal{N}\times\mathcal{N}\times\mathcal{T}|v_{o,d,t}-\Omega_{o,d}>30\}}\phi_{o,d,t}}{\sum_{(o,d,t)\in\{\mathcal{N}\times\mathcal{N}\times\mathcal{T}|v_{o,d,t}-\Omega_{o,d}>0\}}\phi_{o,d,t}}$$

Note that KPI 2 uses the set $\mathcal{Z}$. $\mathcal{Z}$ is the set of origin-destination pairs that use a route with an arc causing a conflict in a time period t.

5.2.5. Performance of the model with different capacity thresholds

The effect of the event requests will be tested by executing the model for different capacity thresholds. The model that includes the event requests as hard constraint ($\Lambda_{s,t} = 0$) is set as the reference model. This is the main principle that the rail-infra manager uses currently. That is, no conflicts with event requests should occur in the schedule for the maintenance projects.

In total, 70 capacity thresholds are tested. The range starts at 0, representing the reference model, and goes up to 6900 with constant intervals of 100. One last model run is executed without threshold capacity, meaning that event requests are not considered. An optimal solution is found for all executions with different capacity thresholds. Figure 5 shows the resulting objective of the model for different thresholds of the capacity.

Compared to the reference model ($\Lambda_{s,t} = 0$), the resulting schedules do not show a change in the objective function value for $\Lambda_{s,t} \leq 1000$. Optimal schedules for $\Lambda_{s,t} \geq 1000$ show a decrease of the passenger hindrance. This value stays constant if the capacity threshold is raised, even if there is an unlimited capacity of the alternative services. This shows that for this specific case, there is an important capacity threshold in the passenger hindrance for a capacity threshold around 1000 passengers. This indicates that there is a track segment within an event request with a passenger flow around the 1000 passengers in the peak hour. Interestingly, it appears to be optimal to schedule a job on that track segment.

Considering the quantitative differences between the objective values, the percentage differences are is 0.46%. In absolute number, this 17500 minutes and is almost 300 hours of passenger delay less between the thresholds below 1000 and above. The percentual difference is not significant, but analysis of KPIs should give insight into the value of the 300 hours reduction in passenger delay.

The values for the KPIs are shown in Figure 6. The last KPI is left out of the results since there are no passengers with a travel time delay higher than 30 minutes.

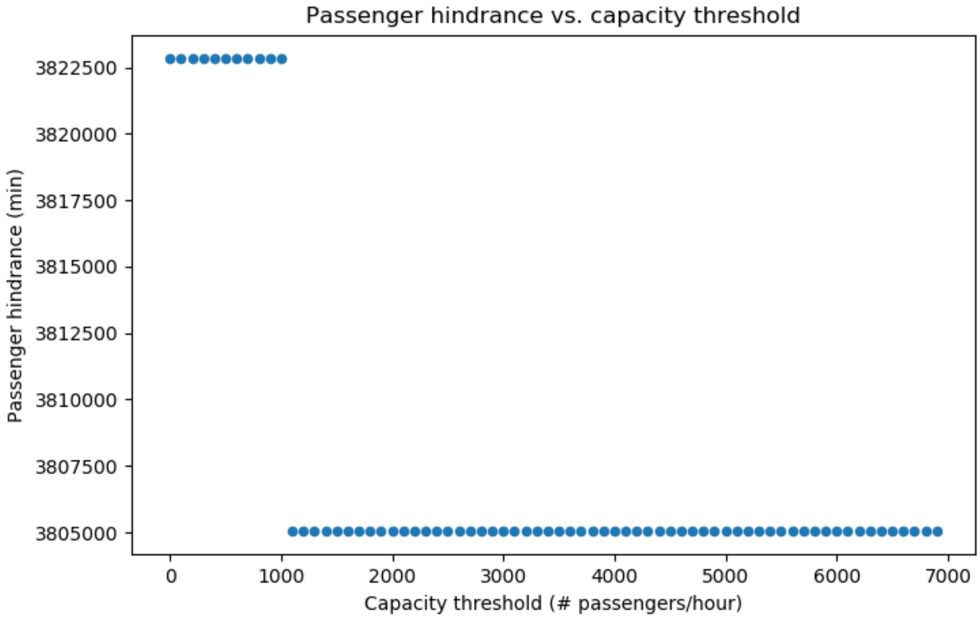

Figure 5: Objective of the model for different capacity thresholds

Similar to the results of the objective function, the capacity threshold of 1000 is again an important point in the number of conflicts. The schedules corresponding to models with a threshold higher than 1000 always result in at least one conflict and below that threshold, conflicts never occur. This confirms the suspicion that causing a conflict with an event request could even reduce the passenger hindrance. Due to the capacity threshold the conflict was not allowed and the conflict is allowed if the threshold is higher than 1000.

The conflict value, i.e. the number of passengers that should travel via alternative services shows a correlation with the number of conflicts. Logically, the conflict value is 0 if no passengers are affected in event request areas. This is the case for all solutions with a threshold lower or equal than 1000 since the maintenance schedule does not result in conflicts. Above the capacity threshold of 1000, the behaviour of the conflict value is similar to the number of conflicts in a solution. This may indicate that the conflicts are always the same combinations of job and events that conflict. This could also explain that per conflict always the same set of passengers is affected.

The passenger delay does not show clear differences between the thresholds. First observations between the lower two figures show a clear inverse correlation between the average delay per affected passenger and the number of affected passengers. Since these KPIs are a direct implication of the objective function, this makes sense. Otherwise this would result in a different value for the objective function.

Furthermore, the delay is minimally lower below the capacity of 1000 than the delay with capacity thresholds above 1000. On the other hand, the number of delayed passengers increases below this threshold. This is an interesting observation, as one would expect that the number of delayed passengers would increase if a conflict with an event occurs. An event causes high passenger demand peaks and therefore it was expected that a violation would result in more passengers with on average less delay.

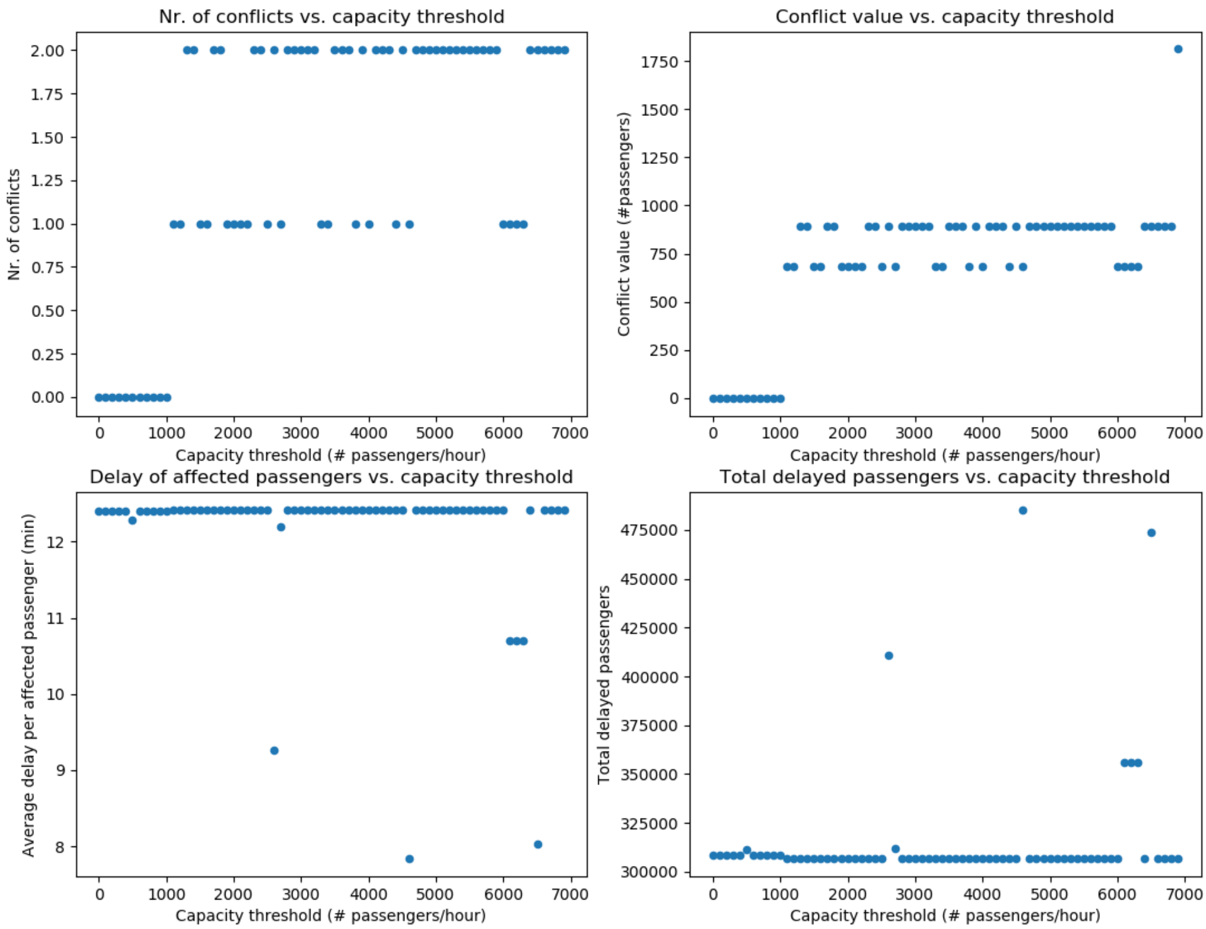

Figure 6: values for the KPIs of the model applied on the case study with increasing threshold values

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

- Above the capacity threshold of 1000 conflicts arise in the optimal solution. Before this point, no conflicts arise in the solution. After this threshold the objective improves, but causes conflicts with event requests.
- Dependent on the capacity, there are multiple optimal schedules with different effects on the passenger hindrance.

In the remainder of this section, aspects of the resulting maintenance schedules will be analyzed.

5.2.6. Analysis of maintenance schedules

The model runs resulted in 70 maintenance schedules, each schedule optimal for one of the considered capacity thresholds. Analysis of 70 schedules shows that there are 44 unique schedules. Considering the solutions with conflicts between event requests and maintenance projects, 34 unique solutions remain. This high amount of different optimal solutions is not completely unexpected, since the passenger demand $\phi_{o,d,t}$ is constant over time if there is no event in a time period t.

Focusing on the 34 unique solutions, the first question is if all 34 solutions are feasible for all $\Lambda_{s,t} > 1000$. All solutions are tested and validated on the model with a threshold of 1100 and the result is that the schedules are all feasible.

Another question is how the differences in the resulting maintenance schedules still lead to the same objective value. Every maintenance schedule has at least one conflict. Consistencies can be found in the conflicts that arise from these schedules. If the resulting maintenance schedule has 1 conflict, this is always with event 52, which is an event with relatively few daily visitors. The jobs that cause the conflict are jobs 360, 289 and 249. In all other cases, the schedule creates 2 conflicts. This concerns a conflict with 42 and 52, or a conflict with 52 and 42. This is also with the same three aforementioned jobs.

The explanation for the consistencies of the conflicting jobs is simple. The affected arcs by these jobs are illustrated in Appendix F and these show that all three jobs are located on the same track segments. Since these jobs are equivalent, the number of unique solutions is reduced by only 2. This can be explained by the fact that there are a significant amount of options to schedule the remaining two jobs. These two remaining jobs are mostly scheduled at time periods without event requests and since the demand is equal for these time periods, such scheduling solutions are trivial. If those schedules are removed as well, 8 unique schedules remain. This is exactly equal to the number of solutions with deviating KPI values for the models with $\Lambda_{s,t} \leq 1000$. There are 7 deviating schedules with a different number of delayed passengers and the last solution has a conflict value of 1750.

For the 10 remaining solutions with $\Lambda_{s,t} \leq 1000$, interesting observations can be made. In all schedules, except one schedule, exactly one of the jobs 466 and 358 is scheduled in a time period with the presence of an event. The other schedule has both jobs located to a time period with a schedule, which is exactly the solution with a different number of delayed passengers in the lower right graph of Figure 6. The other three jobs that cover the same track segments do not have that much flexibility in the optimal solutions. Always one job is scheduled on time period 1, the second job in time period 35 or 36, and the third job in time period 70, 77 or 78.
The increase of the capacity shows that passenger hindrance could be improved and more scheduling options could be considered. The first observation is case study specific and although the improvement in passenger hindrance is minor, the results shows that there are possibilities to improve the passenger hindrance while increasing the number of conflicts. This is an interesting outcome as the expectation was that scheduling around events would result in an increase of the passenger hindrance.
The second observation makes sense as each feasible solution remains feasible if the capacity is increased. Interestingly, there are counter-intuitive options where the number of affected passengers is lower in schedules with conflicts than schedules without conflict, although this difference is again minor.

6. Conclusion

In this thesis, a railway maintenance scheduling problem is addressed with the goal to minimize passenger hindrance by considering events requests submitted by passenger operators. A MILP is developed that is able to find optimal schedule that minimizes the passenger hindrance by considering event requests. This is applied in a case study on a part of the Dutch railway network.
Before addressing the main research question of this thesis, the subquestions will be answered as these are included to help answering the main question:

1. What are factors that the passenger operator considers when selecting track segments that are included in the event requests, and how can these factors be included in the model?

The passenger operator indicates that there are three reasons to decide whether certain track segments should be included in an event request. The main reason is to prevent overcrowded stations after events, which could lead to dangerous circumstances. This is included in the developed model as a constraint indicating the maximum capacity of alternative services that can be provided on track segments where maintenance is carried out.

The other two reasons are logistical and political. These latter two reasons are not included as a contributing factor in the model as there is no insight in the logistics of the passenger operator. Political reasons are not considered because trivially no maintenance projects can be scheduled and therefore does not contribute to the model.

2. What is the difference in the maintenance schedule between a model that includes event requests and a model that does not include event requests?

By means of a case study, insights are obtained regarding the differences in optimal schedules that regard event requests as hard constraint or optimal schedules that completely ignore event requests. The case study mainly shows that relaxing the event request constraint creates more scheduling options without necessarily increasing the passenger hindrance, and it might even reduce the passenger hindrance.

These findings are used to answer the main research question:

How can maintenance projects optimally be scheduled when minimizing passenger hindrance considering event requests?

The developed MILP in this thesis is able to construct optimal schedules for maintenance projects that minimizes the passenger hindrance. To ensure that the optimal solution is found for the passenger, practical limitations are included that represent the motivation of event requests instead of considering the event request as hard constraint. 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 ending of events. Furthermore, the model is able to handle other restrictions that are imposed on maintenance scheduling to ensure that resulting schedules are as well feasible for other rail operators than only passenger operators.

The case study shows that an optimal solution can be reached for limited sized studies. Dependent on the capacity of the alternative services that can be provided, different unique optimal solutions can be obtained. A low capacity threshold results in a higher total higher passenger hindrance than a higher capacity threshold, although the improvement is minor for this case study. The case study also shows that more flexibility in the scheduling may obtained without increasing the passenger hindrance, 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 hindrance based on the results.

7. Discussion

In this section, limitations of this study are discussed. This results in options for further research and implications of the research.

One major limitation of this study concerns the limited information in the data sets. This is probably one factor contributing to the amount of unique, optimal solutions. The data concering passenger flows is constant over the time periods because the data is generalized. If there where minor differences in the passenger demand on a daily level, this would result in one optimal capacity threshold with a corresponding maintenance scheduling. It could, however, be argued that the number of unique solutions, as in this case study, might be even more useful in practical situations than exactly one optimal solution. Highly precise passenger demand predictions might show huge differences compared to the real situation and degrades the quality of a schedule that is based on the wrongly predicted demand.

This also holds for the inclusion of event visitors. The data on passenger demand did not contain the number of passengers visiting certain events. Predicting passenger demand for events is a complex issue due to the wide variety of characteristics in events. Even if events are recurrent, behaviour of visitors might be completely different. In this case, a standard gravity model is used with weights according to the number of boardings on a station. Other passenger prediction models could also be tested, but is not very effective as verification is not possible with the current data.

No waiting times are considered in the model or k-shortest path algorithm due to the fact that these could not be clearly retrieved from the data. Transfers are not always optimal and could therefore result in a significant increase in the travel times times, changing the shortest paths from an origin to a destination. This implicates that the route choices of passengers may differ from the choices that are made in real situations.
Furthermore, if alternative services replace train services, waiting times could deviate from the regular waiting times between those connections. This might also explain why passenger had a delay exceeding 30 minutes. Future research could focus on the inclusion of waiting times and the deviation of these waiting times in the model. One approach is to increase the detail of the network such that it includes arcs on a station level representing transfer connections between different arcs.

There is also further research possible on the method for passenger route choices. Since the model uses k pre-defined paths, it could occur that the actual path is not contained in the k shortest paths and if the resulting deviation in the travel time is significant, optimal solutions might be affected by this. In this case study, $k = 3$ is used and k could be increased to reduce the probability that the actual shortest path is not contained in the set of routes, but this comes at a computational cost. There are options to combine this problem with the shortest path problem to guarantee the shortest path, although the resulting model will not remain linear and other solution methods should be applied.
Furthermore it is assumed that passenger route choices are based on solely the travel time. However, the behaviour of passengers selecting a route might be dependent in cases of maintenance based on the number of transfers and the type transfers.

The capacity of alternative services, $\Lambda_{s,t}$, is assumed to be constant over the track segments and time periods. However, this differs per station and per time period. To be able to apply the model in real situations, these capacity thresholds should be examined. Assuming that buses are used as transportation mode for the alternative services, the number of buses could be determined by considering the maximum number of buses available and the round trip time of the buses.
Furthermore, the corridorbook generally provides a guideline to create feasible schedules, although in practice it is not able to satisfy all rules. In this study, some rules of the corridorbook were not included. These rules are meant for international trains and (e.g. international trains, freight trains) are not accounted for in the maintenance schedule. The impact on these trains should as well be examined.

In future research, options could be considered to improve the current solution method. First, the parameters used in the heuristic solution are not optimized. The simulated annealing algorithm creates generally good solutions as a starting point for the branch-and-bound. This can be improvement by optimizing the initial temperature and the cooling scheme for the problem addressed in thesis, because these impact the exploration and exploitation of solutions in the solution space. Also, improving local search methods to generate neighbor solutions could improve the exploration of solutions. The local search for neighboring solutions is now solely based on single-job movements within the time period, but swap-moves or other moves might be examined.

Second, optimality can only be reached for limited sized instances. In the case study, the model is applied on a part of the Dutch railway over a period of three months. To apply the model on a bigger scale, as for example the entire Dutch railway network over a yearly period, development of (meta-)heuristics for this specific model could be considered as an alternative for the exact method used in this thesis. The results of meta-heuristics might not be optimal solution, but this is not necessarily needed. Since the construction of the schedule should be based on a predicted passenger demand, deviations are not rare and could affect the optimality of solutions in practical situations.

Future directions could also involve passenger flow prediction for events or event venues. Predictions are hard due to high variety of events, but are useful to this thesis (and useful in general) as it increases the accuracy of parameters used in the developed model. Currently, there are relatively few studies on predicting the travel behaviour 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)) short-term based methods to identify unexpected events (e.g. (Li et al., 2017), (Ni et al., 2017)). Applied on road networks, flow predictions for single events or event venues and the impact on the road network of the events (e.g. (Kwoczek et al., 2014) (Kwoczek et al., 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 and therefore, future directions could focus on developing new techniques for passenger flow prediction in railway networks.

Finally, the application of the model is not limited to the construction of schedules. As stated in the problem description, the construction is an iterative process. Therefore some projects could already have a fixed date at some stage in the process. The model can be used to optimally schedule the remaining with the fixed projects included.

Also, the model could be used to provide decision makers with scheduling options. The schedules provide quantitative insight and is a support for the negotiation process. Although there is already a significant amount of rules included in the model to ensure feasibility, negotiations are still needed due to the complexity of creating a railway maintenance schedule in which a lot of stakeholders are involved.

References

Albrecht, A., Panton, D., Lee, D., 2013. Rescheduling rail networks with maintenance disruptions using problem space search. Computers Operations Research 40, 703–712. URL: `https://www.sciencedirect.com/science/article/pii/S0305054810001905`, doi:`https://doi.org/10.1016/j.cor.2010.09.001`. transport Scheduling.

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

Balas, E., Ceria, S., Cornuéjols, G., Natraj, N., 1996. Gomory cuts revisited. Operations Research Letters 19, 1–9.

Balas, E., Perregaard, M., 2002. Lift-and-project for mixed 0–1 programming: recent progress. Discrete Applied Mathematics 123, 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. Journal of Combinatorial Optimization 32, 885–905.

Boland, N., Kalinowski, T., Waterer, H., Zheng, L., 2013. Mixed integer programming based maintenance scheduling for the hunter valley coal chain. Journal of Scheduling 16, 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 Applied Mathematics 163, 34–52. URL: https://www.sciencedirect.com/science/article/pii/S0166218X12002338, doi:https://doi.org/10.1016/j.dam.2012.05.027. 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. The Journal of the Operational Research Society 57, 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., 2021. Railway Maintenance Reservation Scheduling Considering Train Traffic and Maintenance Demand. Master's thesis. University of Twente.

Caprara, A., Fischetti, M., 1996. {0, 1/2}-chvátal-gomory cuts. Mathematical Programming 74, 221–235.

Caprara, A., Fischetti, M., Letchford, A.N., 2000. On the separation of maximally violated mod-k cuts. Mathematical Programming 87, 37–56.

Cheung, B.S., Chow, K., Hui, L.C., Yong, A.M., 1999. Railway track possession assignment using constraint satisfaction. Engineering Applications of Artificial Intelligence 12, 599–611.

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

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.

Dijkstra, E.W., et al., 1959. A note on two problems in connexion with graphs. Numerische mathematik 1, 269–271.

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.

Forsgren, M., Aronsson, M., Gestrelius, S., 2013. Maintaining tracks and traffic flow at the same time. Journal of Rail Transport Planning Management 3, 111–123. URL: https://www.sciencedirect.com/science/article/pii/S2210970613000267, doi:https://doi.org/10.1016/j.jrtpm.2013.11.001. robust Rescheduling and Capacity Use.

Gu, Z., Nemhauser, G.L., Savelsbergh, M.W., 1999. Lifted flow cover inequalities for mixed 0-1 integer programs. Mathematical Programming 85, 439–467.

Günlük, O., Pochet, Y., 2001. Mixing mixed-integer inequalities. Mathematical Programming 90, 429–457.

Gurobi, a. Mixed-integer programming (mip) – a primer on the basics. URL: https://www.gurobi.com/resource/mip-basics/. accessed: 2022-2-9.

Gurobi, b. Parameter descriptions. URL: https://www.gurobi.com/documentation/9.5/refman/parameter_descriptions.html. accessed: 2022-4-17.

Higgins, A., 1998. Scheduling of railway track maintenance activities and crews. The Journal of the Operational Research Society 49, 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.

de Jonge, M., 2017. Railway Maintenance Reservation Scheduling Considering Train Traffic and Maintenance Demand. Master's thesis. University of Twente.

Kirkpatrick, S., Gelatt, C., Vecchi, M., 1983. Optimization by simulated annealing. Science (New York, N.Y.) 220, 671–80. doi: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), pp. 1194–1200 vol.2. doi: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. Journal of Visual Languages Computing 25, 973–980. URL: https://www.sciencedirect.com/science/article/pii/S1045926X14001219, doi:https://doi.org/10.1016/j.jvlc.2014.10.028. 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. doi: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. Transportation Research Part C: Emerging Technologies 77, 306–328. URL: https://www.sciencedirect.com/science/article/pii/S0968090X17300451, doi:https://doi.org/10.1016/j.trc.2017.02.005.

Lidén, T., 2015. Railway infrastructure maintenance - a survey of planning problems and conducted research. Transportation Research Procedia 10, 574–583. URL: https://www.sciencedirect.com/science/article/pii/S2352146515001982, doi:https://doi.org/10.1016/j.trpro.2015.09.011. 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. Transportation Research Part C: Emerging Technologies 74, 327–347. URL: https://www.sciencedirect.com/science/article/pii/S0968090X16302340, doi:https://doi.org/10.1016/j.trc.2016.11.016.

Lidén, T., Kalinowski, T., Waterer, H., 2018. Resource considerations for integrated planning of railway traffic and maintenance windows. Journal of Rail Transport Planning Management 8, 1–15. URL: https://www.sciencedirect.com/science/article/pii/S2210970617300860, doi:https://doi.org/10.1016/j.jrtpm.2018.02.001.

Macedo, R., Benmansour, R., Artiba, A., Mladenović, N., Urošević, D., 2017. Scheduling preventive railway maintenance activities with resource constraints. Electronic Notes in Discrete Mathematics 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. Operations Research Letters 16, 255–263. URL: https://www.sciencedirect.com/science/article/pii/0167637794900388, doi:https://doi.org/10.1016/0167-6377(94)90038-8.

Ni, M., He, Q., Gao, J., 2017. Forecasting the subway passenger flow under event occurrences with social media. IEEE Transactions on Intelligent Transportation Systems 18, 1623–1632. doi:10.1109/TITS.2016.2611644.

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

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

Padberg, M., 1989. The boolean quadric polytope: some characteristics, facets and relatives. Mathematical programming 45, 139–172.

Padberg, M.W., 1973. On the facial structure of set packing polyhedra. Mathematical programming 5, 199–215.

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

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. International Journal of Tourism Research 9, 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 Transactions on Pattern Analysis and Machine Intelligence 39, 2113–2126. doi:10.1109/TPAMI.2016.2635136.

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

Sousa, J.P., Wolsey, L.A., 1992. A time indexed formulation of non-preemptive single machine scheduling problems. Mathematical programming 54, 353–367.

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

Weismantel, R., 1997. On the 0/1 knapsack polytope. Mathematical Programming 77, 49–68.

Wolsey, L.A., 1976. Facets and strong valid inequalities for integer programs. Operations research 24, 367–372.

Yen, J.Y., 1971. Finding the k shortest loopless paths in a network. management Science 17, 712–716.

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. URL: https://www.sciencedirect.com/science/article/pii/S0305048317310575, doi:https://doi.org/10.1016/j.omega.2018.08.005.

Zhang, T., Andrews, J., Wang, R., 2013. Optimal scheduling of track maintenance on a railway network. Quality and Reliability Engineering International 29, 285–297.

Appendix A. Base model

Boland et al. (2014) addresses the maximum total flow problem with flexible arc outages. Consider a railway network over a finite time horizon $[T]$ with a node set N, arc set A, source s and sink s', and capacities $u_a \in \mathbb{R}$ for each arc $a \in A$ in the network. Also, they are given a set of maintenance jobs J where a job $j \in J$ is characterized by the associated arc a_j, processing time p_j, release time r_j and deadline d_j. The problem they try to solve is to schedule the set of maintenance jobs such that the total throughput from the source to the sink is maximized over the considered time period. They define three variables to model their problem: $\phi_{at} \in \mathbb{R}$ is the flow on arc a over time interval t,$x_{at} \in \{0, 1\}$ indicates the availability of arc a at time t, and $y_{jt} \in \{0, 1\}$ indicates the start time t of a job j.

$$\max \quad \sum_{t=1}^{T} \sum_{a\in\delta^+(s)} \phi_{at} \tag{A.1a}$$

$$\text{s.t.} \quad \sum_{a\in\delta^-(v)} \phi_{at} - \sum_{a\in\delta^+(v)} \phi_{at} = 0 \qquad \forall v \in N \setminus \{s, s'\}, t \in [T] \tag{A.1b}$$

$$\phi_{at} \leq u_a x_{at} \qquad \forall a \in A, t \in [T] \tag{A.1c}$$

$$\sum_{t=r_j}^{d_j-p_j+1} y_{j,t} = 1 \qquad \forall j \in J \tag{A.1d}$$

$$x_{at} + \sum_{t'=\max\{r_j,t-p_j+1\}}^{\min\{t,d_j\}} y_{jt'} \leq 1 \qquad \forall a \in A, t \in [T], j \in J_a \tag{A.1e}$$

$$\phi_{at} \in \mathbb{R}_{>0} \qquad \forall a \in A, t \in [T] \tag{A.1f}$$

$$y_{jt} \in \{0, 1\} \qquad \forall j \in J, t \in [r_j, d_j - p_j + 1] \tag{A.1g}$$

$$x_{at} \in \{0, 1\} \qquad \forall a \in A, t \in [T] \tag{A.1h}$$

The objective function A.1a maximizes the total throughput over the considered time horizon. Constraints (A.1b) and (A.1c) model flow conservation and capacity constraints. Constraint (A.1d) ensures that each job starts exactly once and constraint (A.1e) ensures that an arc is unavailable while a job is being processed at that arc. Constraints (A.1f)-(A.1h) model the variable types. For more detailed information concerning the model, the reader is referred to the paper of (Boland et al., 2014).

Appendix B. Modified Yen's algorithm

A common method to find the k shortest paths in a given network is Yen's algorithm. (Yen, 1971) The principle of this is algorithm is to use a single shortest path algorithm, to find the shortest path and iteratively blocking edges by temporarily setting the edge weight to infinity. In this version of Yen's algorithm, the algorithm of Dijkstra et al. (1959) is used to find shortest paths. The modification applied here is an alteration in method that decides which edges should be blocked each iteration. A pseudo code for the modified Yen's algorithm is shown in Algorithm 1.

Data: N (nodes), o (origin), d (destination), w (arc weights), K (number of shortest paths), B (blocked track segment combinations)
Result: $A = \{A^1, ..., A^K\}$
$A \leftarrow \emptyset$;
$A^1 \leftarrow Dijkstra(o, d, N, w)$;
$A \leftarrow A \cup \{A^1\}$;
$k \leftarrow 2$;
while $k \leq K$ **do**
  for $i \in A^{k-1}$ **do**
    $R^k_{o\to i} \leftarrow A^{k-1}_{o\to i}$;
    $W \leftarrow w$;
    for $P \in A$ **do**
      if $P_{a\to i} = R^k_{o\to i}$ **then**
        $W(B(P_i, P_{i+1})_j) \leftarrow \infty \quad \forall j \in B$
      end
    end
    $V^k_{i\to d} \leftarrow Dijkstra(i, d, S, W)$;
    $A^k_i \leftarrow R^k_{a\to i} + V^k_{i\to d}$;
  end
  $A \leftarrow A \cup \{A^k\}$ where $A^k \leftarrow \min_i \{A^k_i\}$;
  $k \leftarrow k + 1$;
end

Algorithm 1: Modified Yen's algorithm for K-shortest paths

The set B contains for each segment all tracks that will be (partly) blocked if maintenance works are scheduled on that arc. The idea behind the use of this set is that maintenance on a track segment affects all train routes that use the concerned track segment. The skip-stop property of some train routes yield a database in which a track segment a might be contained in a bigger track segment b implying that if maintenance works are carried out at the contained track a, the track segment b cannot be used as well at the time of the maintenance works. This also works the other way around, all track segments that are contained in a track segment that is maintained are unavailable as well.

Appendix C. Simulated annealing procedure for initial feasible solutions

An initial solution as input for the branch-and-cut method of the Gurobi solver can be found by means of simulated annealing. Simulated annealing is a meta-heuristic and is for combinatorial optimization problems a method to find reasonably good solutions in a relatively short amount of time.

Simulated annealing algorithm

Simulated annealing is applied to find the starting times $S_j \forall j \in J$. An initial solution is used as input and the control value c should to determine at what rate deteriorating solutions are accepted. Each iteration, a neighboring solution $N(S)$ is examined and accepted if this solution improves 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 2.

```
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) ≤ f(S) then
            S → S_new
        else
            S → S_new with probability exp((f(S) − f(S_new)) / c)
        end
    end
    Update c;
end
```

Algorithm 2: Simulated annealing

Evaluation of the objective function

To evaluate the passenger hindrance of a given solution, each path taken by passengers for each (o, d) pair at each time period should be determined. This can be reduced to the time periods in which jobs are scheduled, because in time periods without maintenance no passengers experience hindrance.

Verifying feasibility

Note that that each solution is required to be feasible, because it should be used as input for the branch-and-cut algorithm of Gurobi, which should start at a point in the solution space. Feasibility is guaranteed if the following constraints (Eq.C.1 - C.4) are met:

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

$$\sum_{a \in c}(1 - x_{a,t}) \leq 1 \qquad (\forall c \in \mathcal{C}, t \in \mathcal{T}, j \in \mathcal{J}) \tag{C.2}$$

$$y_{j,t} + y_{j',t'} \leq 1 (\forall a \in \mathcal{A}, (j,t), (j',t') \in \mathcal{J}_a \times \mathcal{T} : j \neq j, t' \in (t - \psi_{j'} - \tau, t + \psi_j + \tau)) \tag{C.3}$$

$$\sum_{a \in s} \sum_{(o,d) \in \mathcal{N} \times \mathcal{N}} \sum_{i=1|a \in R_{o,d,i}}^{k} h^i_{o,d,t} \beta_{o,d,t} \phi_{o,d,t} \leq \Lambda_{s,t} + \sum_{a \in s} x_{a,t} M \qquad (\forall s \in \mathcal{E}_t, t \in \mathcal{T}) \tag{C.4}$$

Constructing an initial solution

For the computational instances, an initial solution is obtained via a construction heuristic. Here, job scheduling is considered per arc and are then located. 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. This solution is generated such that it does not cause conflicts and can therefore be used as initial solution for all tested thresholds.

Neighboring solutions

Neighboring solutions are randomly generated. First, an arc is selected that corresponds to one or more jobs. One of the jobs is randomly selected and a new starting time is proposed. The only restriction on the starting times that is given is that jobs may not overlap. If that is possible, then it will choose a new starting time and otherwise no neighboring solution will be chosen and a new iteration starts.
Other restrictions are not considered. However, a penalty value is added to the objective function if a restriction is violated. The advantage is that the heuristic does not get stuck in a certain point and if the penalty is large enough, the best solution will always be feasible since the initial solution was feasible.

Cooling scheme

The cooling scheme defines the duration of the algorithm. It consists of an initial temperature T and an annealing scheme. The initial temperature should be an estimate of the largest difference between two neighboring solutions. This causes that that every solution has a chance to be accepted and enforces exploration of the solution space. When the temperature reaches a point below T_s, the desired energy level is reached and the algorithm finishes The annealing of the temperature follows a scheme. The original annealing scheme of simulated annealing is used in this thesis, which is proposed by Kirkpatrick et al. (1983):

$$T_{t+1} = c * T_t \qquad t = 1, 2, ... \tag{C.5}$$

The constant c typically lies between 0.8 and 0.99. For all model executions, the value 0.99 is used.

Appendix D. Passenger demand induced by events

A destination gravity model will be used to model the additional passenger demand that is induced by events. To model the passenger streams on the network, the gravity model is used. Destination-constrained model for trips from origin i to destination j, T_{ij}:

$$T_{ij} = \frac{O_i f(c_{ij})}{\sum_{i=1}^{z} O_i f(c_{ij})} D_j \tag{D.1}$$

Here, D_j is the number of visitors of an event (attracted trips by only considering the event), O_i the number of generated trips in total and $f(c_{ij})$ the distance decay function for a travel cost c_{ij} from origin i to destination d.

For now, the distance decay function that is used is:

$$f(c_{ij}) = 1 \tag{D.2}$$

Since the problem deals with events and an assumption is made that should hold for all events, the distance decay function is irrelevant. It is assumed that visitors of an event do not consider distance as a limiting factor.

However, the outflow of visitors generally induces higher peaks than the inflow, which is mostly more spread over the hours before the event. This leads to 2 considerations. First, the assumption that all induced trips by public transport will go home within one hour after the event has finished. Second, the fact that the focus is on the number of trips from station corresponding to the event venue towards the home locations of the visitors and hence, trips are reversed to model the outflow instead of the inflow.

Therefore, the assumption is as well that the peak hour is the hour after the event is finished and consists of the recurrent passenger demand at that hour, which can be derived from the dataset and is a factor 0.1 of the daily demand, plus the additional passenger demand of the events, which are all visitors travelling by train.

Appendix E. Generation of toy instances

Generated instances are derived from n-complete graphs, denoted as K_n, graphs with n nodes and with the characteristic that each node is adjacent to all other nodes in the graph. This characteristic is useful as running the model with a disconnected graph is the same as running the model for each disconnected component of the graph. For the network, the number of nodes is predefined and those nodes are equally spread over the circumference of a unit circle. The weight or expected travel time, ω_a^e, of an arc a equals the euclidean distance between a pair of nodes and the increased travel time, ω_a^i, is the euclidean distance times a factor of 1.3. Other characteristics of an instance are partially randomly generated. For each $\phi_{o,d,t}$, the passenger demand between two nodes o and d at time t, a random integer value is assigned between 0 and 2000. A predefined of number of paths, k, are computed for each (o, d)-pair in the network. To construct a path from o to d, random stations are iteratively chosen until the destination d is selected. The only condition in which a path is rejected is when the path is exactly the same as one of the other predefined paths. Note that it is not required to compute the shortest paths as this is only a test and any randomly generated path will do for this goal. The corresponding expected travel time any two nodes o and d, $\Omega_{o,d}$, is the travel time corresponding to the shortest path of the k predefined paths between an origin an destination. The characteristics of the jobs are randomly generated. This concerns the location of the job and the processing time. The location of the job consist of a single arc and it is possible that multiple jobs are assigned to the same arc. It is assumed that each job only occupies one arc while being processed, but this is not necessarily the case in practical situations. The processing time of a job, p_j, is a random integer in the interval $[1, ..., \frac{T}{2}]$. the sum of processing times on a single arc should not exceed the time horizon to guarantee a feasible solution.

Event requests and corridor book restrictions, except the minimum time interval between jobs on a single arc, are not considered in the toy instances. The goal of evaluating toy instances is to observe the effect of increasing instance sizes on the computational times. Including event requests and corridor book restrictions could imply that the difficulty of finding a feasible solution increases, but should reduce the computational times the as the number of feasible solutions is reduced.

The code for instance generation and all generated instances used for testing the computational times can also be found in the online data set.

Appendix F. Track segments corresponding to maintenance projects

Maintenance projects

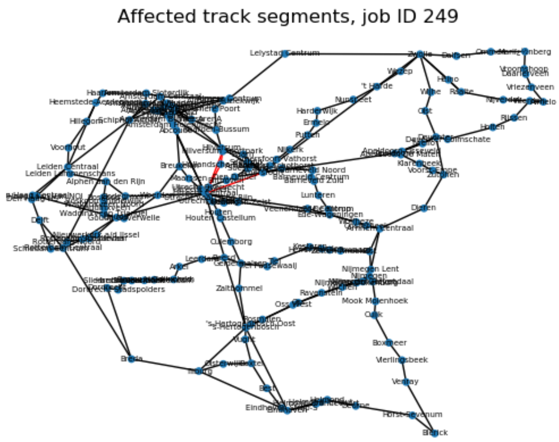

Figure F.7: Unavailable track segments during maintenance project 249

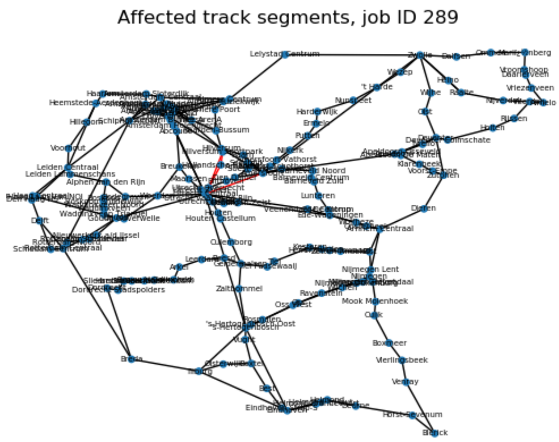

Figure F.8: Unavailable track segments during maintenance project 289

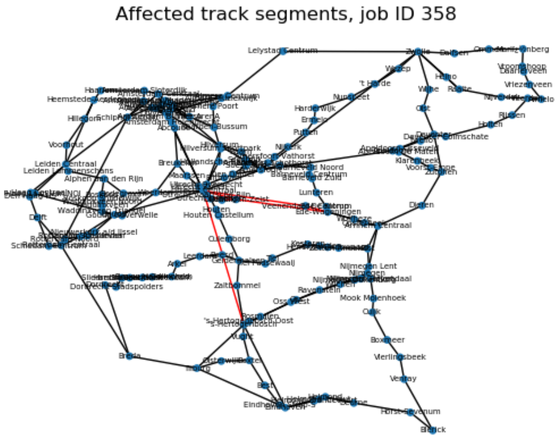

Figure F.9: Unavailable track segments during maintenance project 358

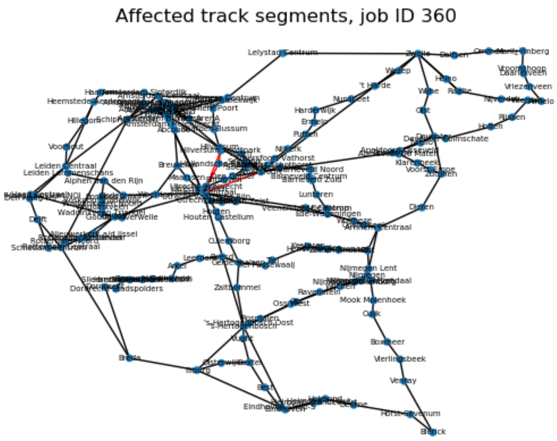

Figure F.10: Unavailable track segments during maintenance project 360

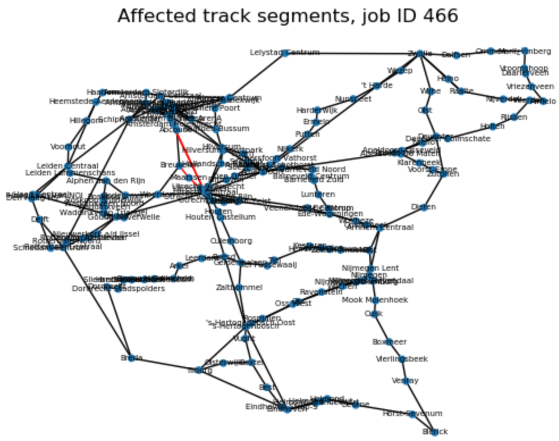

Figure F.11: Unavailable track segments during maintenance project 466