Application of an Iterative Framework for Real-time Railway Rescheduling [1]

Twan Dollevoet[2,6], Dennis Huisman[2,3,6], Leo G. Kroon[3,4,6],

Lucas P. Veelenturf[5], Joris C. Wagenaar[4,6]

[2]*Econometric Institute, Erasmus University Rotterdam,*
P.O. Box 1738, 3000 DR Rotterdam, The Netherlands
{dollevoet, huisman}@ese.eur.nl

[3]*Process quality and Innovation, Netherlands Railways,*
Utrecht, The Netherlands

[4]*Rotterdam School of Management, Erasmus University,*
P.O. Box 1738, 3000 DR Rotterdam, The Netherlands
{lkroon, jwagenaar}@rsm.nl

[5]*School of Industrial Engineering, Eindhoven University of Technology,*
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
l.p.veelenturf@tue.nl

[6]*Erasmus Center for Optimization in Public Transport,*
Rotterdam, The Netherlands

Abstract

Since disruptions in railway networks are inevitable, railway operators and infrastructure managers need reliable measures and tools for disruption management. Current literature on railway disruption management focuses most of the time on rescheduling one resource (timetable, rolling stock or crew) at the time. In this research, we describe the application of an iterative framework in which all these three resources are considered. The framework applies existing models and algorithms for rescheduling the individual resources. We extensively test our framework on instances from Netherlands Railways and show that schedules which are feasible for all three resources can be obtained within short computation times. This case study shows that the framework and the existing rescheduling approaches can be of great value in practice.

Keywords: Railway Operations, Disruption Management, Algorithmic Framework

1 Introduction

Railway transportation plays an important role in the lifes of many people. They travel by train to their work or school, or for leisure purposes. One of the most important criteria for passenger satisfaction is the reliability of the journeys. However, disruptions like accidents, malfunctioning infrastructure or rolling stock, or crew unavailability are inevitable in a railway system. As a consequence, passengers face cancelled, delayed or overcrowded train services. It is very important for railway operators to reduce the nuisance caused by these disruptions for the passengers as much as possible.

As stated in the overview paper by Cacchiani et al. (2014): *"the development of algorithmic real-time railway rescheduling methods is currently still mainly an academic field, where the research is still far ahead of what has been implemented in practice."* The models and algorithms from literature mainly deal with rescheduling either the timetable, or the rolling stock, or the crew. It is currently unknown whether it is possible to combine the algorithms for individual resources and come up with an overall feasible solution that is satisfactory for the passengers. A solution is overall feasible if the three resource schedules are feasible in themselves and are mutually compatible. The latter means that both rolling stock and crew are available for each trip in the timetable. This might be one of the reasons why the models from literature have not been implemented in practice yet.

In this paper, we make a first step in bridging this gap between theory and practice, by introducing an iterative framework for timetable, rolling stock, and crew rescheduling. We show that a satisfactory, overall feasible solution can usually be found in only a few iterations. This suggests that the approaches for rescheduling individual resources can be combined and applied in practice during a disruption.

In the iterative framework, we use earlier published models and algorithms on (macroscopically) adjusting the timetable, rescheduling the rolling stock, and rescheduling crew schedules. The framework first computes a new timetable. Then, it reschedules the rolling stock, covering as many trips in the timetable as possible. Trips that cannot be covered by rolling stock are then cancelled in the timetable. Finally, the crew duties are rescheduled. Again, the objective is to cover as many trips from the timetable as possible. If some trips cannot be covered by crew, these trips are cancelled, and another iteration of the framework is necessary. Otherwise, if all trips are covered by crew, the algorithm terminates. We emphasize that our framework is very generic: Instead of the particular models and algorithms we use, other methods can be used in the framework as well, as long as they solve a similar problem.

We demonstrate the effectiveness of the iterative approach on real-world instances from Netherlands Railways (Nederlandse Spoorwegen, or NS). We consider 976 instances in total. In half of them, one of the tracks between two stations is blocked for a certain period of time. Then, only limited train traffic is possible between these stations. In the other half, all tracks between two stations are blocked and no train traffic is possible at all. The most important objective is to minimize the total duration of the cancelled train services.

The contribution of this paper is threefold. Firstly, we introduce an iterative framework to reschedule the timetable, rolling stock, and crew. This all-in-one framework leads to an overall feasible solution for all resources. Secondly, we show that the algorithm converges to a satisfactory solution for all considered real-world instances in a few iterations. This shows that the proposed iterative approach is sufficient and suggests that an integrated approach is not required to obtain satisfactory solutions that are overall feasible. Thirdly, we show that the framework and the underlying algorithms that we use are able to solve practical problems and can be of great benefit to railway operators. In this way, we hope to reduce the earlier mentioned gap between theory and practice.

The remainder of the paper is structured as follows. Section 2 reviews the relevant literature. Section 3 contains a description of the iterative framework. This section includes a short description of the algorithms we use to reschedule the individual resources. In Section 4, we present results on 976 disruptions on the railway network in the Netherlands. Finally, we finish the paper with some concluding remarks in Section 5.

2 Literature review

A disruption usually causes the timetable, the rolling stock schedule, and the crew schedule to be infeasible. The timetable and rolling stock schedule may contain trips that make use of infrastructure that is temporarily unavailable. These trips cannot be operated, which might prevent some crew members to perform all tasks in their duties. As a result, the resource schedules need to be adjusted. In current practice, this is mostly done manually. First, often with the help of contingency plans, the timetable is rescheduled. Then, with the new timetable as input, the rolling stock and crew tasks are rescheduled manually, one by one. This is a time consuming process, so decision support tools are most welcome.

Most of the scientific literature on railway disruption management focuses on rescheduling only one of the three resources. In this section, we will briefly review the literature on rescheduling the timetable, the rolling stock, and the crew. Fore a more in depth review we refer to Cacchiani et al. (2014).

The literature on timetable rescheduling can be classified in two parts: macroscopic and microscopic timetable rescheduling. Macroscopic approaches to timetabling model the infrastructure on a high level of abstraction and usually deal with larger disruptions. For example, certain tracks might be unavailable for a couple of hours. Amongst others, Louwerse and Huisman (2014), and Veelenturf et al. (2016a) have recently developed a macroscopic model for timetable rescheduling and have performed tests on the Dutch railway network. Zhan et al. (2015) developed a different macroscopic model and tested it on the Chinese railway network.

In contrast, microscopic models consider the railway infrastructure with a high level of detail. By doing so, the propagation of delays can be modelled with high accuracy. These models are usually applied to resolve smaller disturbances, e.g., few delays of up to half an hour. We refer to D'Ariano et al. (2007) and Corman et al. (2011) for examples of microscopic approaches to timetable rescheduling tested on the Dutch railway system and to Lamorgese and Mannino (2015) for microscopic rescheduling cases tested and implemented on the Italian and Norwegian railway network, respectively.

There are multiple papers with a focus on rescheduling the rolling stock. For instance, Nielsen et al. (2012) adjusted the Composition Model from Fioole et al. (2006) and applied it in a disruption management setting. In this model, the rolling stock rescheduling problem is formulated as a multi-commodity flow model. Here, the nodes correspond to stations and the arcs represent the trips between stations, or waiting inside stations. Furthermore, there is also a transition graph describing the feasible transitions of compositions in the stations. In the transition graph we have nodes representing trips for which rolling stock is required and arcs representing possible changes in the rolling stock composition between these trips. Haahr et al. (2014) developed a unit based model for a similar problem, where a specific path is created for each rolling stock unit separately. The model is then solved by means of column generation. The performance of these models is compared in Haahr et al. (2016) on both the Dutch and the Danish railway network.

The third resource is the crew. Multiple researchers have investigated crew rescheduling. Rezanova and Ryan (2009) model crew rescheduling as a Set Partitioning Problem and solve it by column generation. In a similar fashion, Potthoff et al. (2010) solve a Set Covering Problem by column generation and Lagrangian relaxation. This latter approach is extended by Veelenturf et al. (2012) with the possibility of retiming some of the tasks. Using a completely different method, Abbink et al. (2009) solve the crew rescheduling problem by means

of an agent based system. Here, agents correspond to crew members and can swap parts of their duties.

All these papers show that models and algorithms can be used as decision support tools for rescheduling one resource individually. However, it has never been tested whether these individual rescheduling algorithms can be combined and lead to a solution that is overall feasible. If, for instance, no train driver can be found for a particular trip, it means that this specific trip cannot be executed. As a result, the timetable and rolling stock schedule become infeasible, and need to be rescheduled again. In the next section we propose an iterative framework that copes with these interactions.

There are few papers that investigate the integration of all or at least two of the rescheduling steps. However, these papers focus mainly on small or less complex railway networks. Examples are Walker et al. (2005), who integrate timetable and crew rescheduling, and Cadarso et al. (2013) and Cadarso et al. (2015), who integrate timetable and rolling stock rescheduling. Cadarso et al. (2013) also explicitly consider the effect of the rescheduling measures on the passenger demand and on the required seat capacity.

3   Framework

In this section, the iterative framework for real-time railway rescheduling is introduced. Furthermore, we describe the interactions between the different modules in the framework and we discuss the modules individually. Note that the modules that we use have been developed with a sequential approach in mind. Our framework performs this sequential approach iteratively. As a consequence, the models and algorithms do not need to be adjusted or reconfigured, but can be used without any modifications.

3.1   Framework

The real-time railway rescheduling framework can be used for disruptions which lead to a temporary blockage of one or more tracks. In this paper, we assume that the duration of the disruption is known and fixed. The process can, however, be repeated as soon as there is new information available about the duration. In other words, the process can be embedded in a rolling horizon algorithm to handle the uncertainty regarding disruptions. A similar approach has been suggested by Nielsen et al. (2012) for rolling stock rescheduling. Alternatively, the uncertainty about the duration can be incorporated in the individual modules comprising the framework. As an example, Veelenturf et al. (2016b) develop a quasi-robust crew rescheduling algorithm that takes into account the uncertainty about the duration of the disruption.

Due to the blockage of (some of) the tracks, the timetable becomes infeasible and needs to be rescheduled. Furthermore, the crew and rolling stock schedules need to be adjusted as well. A schematic overview of the framework can be found in Figure 1.
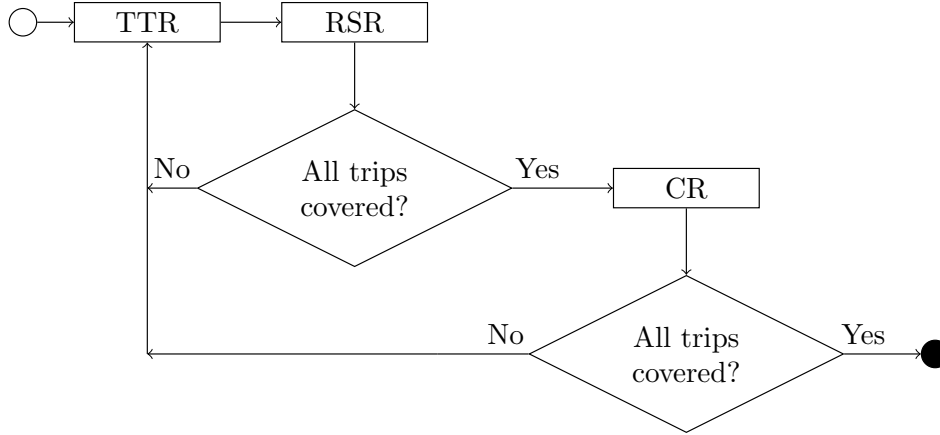
Figure 1: Overview of the iterative framework.

The framework starts by rescheduling the timetable macroscopically (TTR). The timetable should be adjusted by means of delaying or cancelling train services. The objective when rescheduling the timetable is to find a balance between cancelling as little train services as possible and minimizing the delay introduced in the timetable. Output of this module is a disposition timetable that is feasible with respect to the reduced infrastructure capacity.

The new disposition timetable is given as input to the module responsible for rescheduling the rolling stock (RSR). Cancelled train services can result in the original rolling stock circulation being infeasible. The goal when rescheduling the rolling stock is to assign a rolling stock composition (a set of combined rolling stock units) to as many trips in the disposition timetable as possible. Here, a trip is a part of a train service between two stations where the rolling stock composition can be changed.

It might be impossible to cover all trips from the disposition timetable with rolling stock. In that case, the timetable should be rescheduled a second time, preferably in such a way that rolling stock is available for all trips in the new timetable. One obvious solution is to cancel all trips in the timetable without a rolling stock composition assigned to it, but one could also consider more elaborate approaches. This process is repeated until a timetable is obtained for which rolling stock can be assigned to all trips.

Due to cancelled or delayed train services, the original crew schedule might be infeasible as well. The third module of the framework is responsible for rescheduling the crew (CR). This is done by appointing a new duty to every crew member. A duty is a list of tasks to be performed by a single crew member. A task corresponds to performing work (e.g., as a driver or as a conductor) on a certain trip. The most important constraints in the crew rescheduling part are the crew regulation rules (e.g., the presence of a meal break and a maximal working duration).

There might be tasks that cannot be assigned to any crew member. As a consequence, the corresponding trip cannot be performed. In that case, the timetable should be adapted in such a way that all trips can be covered with crew. After such an adjustment of the timetable, also the rolling stock might have to be rescheduled again. To prevent several iterations in the loop being necessary, the objective of the crew rescheduling approach is to assign crew to as many tasks as possible.

Algorithm 1 Iterative framework

1: **Input**: Characteristics of the disruption
2: Reschedule the macroscopic timetable (TTR).
3: Use output of TTR to reschedule the rolling stock (RSR).
4: **if** Not all trips are covered by rolling stock **then**
5:     Timetable needs adjustments; go back to Line 2.
6: **end if**
7: Use output of TTR and RSR to reschedule the crew (CR).
8: **if** Not all trips are covered by crew **then**
9:     Timetable needs adjustments; go back to Line 2.
10: **else**
11:     Overall feasible schedule found.
12: **end if**

A summary of the iterative framework is shown in Algorithm 1. It needs a timetable, a rolling stock and a crew rescheduling approach. In Sections 3.3-3.5 we briefly describe the timetable, rolling stock and crew rescheduling algorithms we apply in this research. However, note that one can replace the particular approach we have chosen for a resource by any other rescheduling algorithm for that resource.

As is common in practice, we reschedule first the rolling stock and then the crew. One reason is that drivers cannot run all rolling stock types and that the number of conductors required for a trip can depend on the length of the rolling stock composition. However, if these dependencies are discarded, it might be possible to reschedule the crew first, and then the rolling stock. In that case, Lines 3-5 and Lines 7-9 are interchanged in the algorithm above. We also test this variant of the framework. The results are presented in Section 4.3.2.

3.2   Interactions

There are several interactions between the different modules comprising the iterative framework. These interactions, indicated by arrows in Figure 1, are discussed in this section in more detail.

First, at the time a disruption occurs, all modules require information on the current state of the railway system. This includes all events that have taken place up to this time. These events can no longer be changed. For the rolling stock and crew rescheduling modules, this information furthermore includes the original rolling stock and crew schedule. The rolling stock rescheduling module requires the passenger demand for each trip, such that enough capacity can be provided.

The timetable rescheduling module requires more information on the current state. First, it requires the timetable as it has ran up to the current time, in order to determine the location of all trains. Second, it needs the time the disruption takes place (the current time), because trains that have already departed cannot be cancelled any more. The timetable rescheduling approach we use also requires the duration of the disruption as input. Finally, the location of the blockage and the specific tracks that are blocked are required as input.

Another interaction is the exchange of information between the timetable rescheduling module and the rolling stock and crew rescheduling modules. The rolling stock and crew rescheduling modules need to know the newly constructed disposition timetable. This timetable describes for all non-cancelled train services all departure and arrival times at the

stations. The interaction back from the rolling stock and crew rescheduling module to the timetable rescheduling module consists of the trips for which no rolling stock or crew can be found.

The last interaction is an information stream between the rolling stock and the crew rescheduling modules. The rolling stock schedule influences the transfer time for crew members. For example, if the next trip of a crew member uses the same rolling stock, then the crew does not have to walk to another train and no buffer time is needed. Furthermore, not every crew member is allowed to run all types of rolling stock. We decided to make this interaction one-way: When rescheduling the rolling stock, we do not keep track of which crew member is assigned to the trip. Therefore, the crew rescheduling module should take into account which type of rolling stock is assigned to a certain trip.

3.3 Timetable Rescheduling

We use two different approaches for rescheduling the timetable in the iterative framework. The first time the timetable is rescheduled, we use a sophisticated approach, which is based on Veelenturf et al. (2016a). This approach considers the reduced infrastructure capacity caused by the disruption. Thereafter, if the timetable must be adjusted because of a lack of rolling stock or crew, we use a greedy approach. This combination of the sophisticated and greedy approach ensures that the number of iterations is finite. Furthermore, it reduces computation time. We now first explain the sophisticated approach and then the greedy approach.

Sophisticated approach

If the timetable is rescheduled for the first time, we use the approach suggested by Veelenturf et al. (2016a). Here, the macroscopic timetabling problem is modelled using an event activity network. The events in this network represent the departures and arrivals of train services and the activities represent the minimal times (e.g. running, dwell and headway times) which should be scheduled between two events. The infrastructure capacity is modelled as the number of available tracks, which means that the precise locations of switches and signals are neglected. An important feature of this model is that it also takes the rolling stock capacity into account by keeping track of the number of rolling stock compositions entering and leaving a station. For each train service, it checks whether a rolling stock composition is available. If no rolling stock composition is available, that particular train service is cancelled. Detailed information about rolling stock types and coupling and uncoupling options are not considered. This means that the rolling stock capacity is measured on the level of compositions instead of on the level of units. This leads to a high probability that a feasible rolling stock schedule can be found for the disposition timetable, in which all trips are covered by rolling stock.

The model is formulated as a MIP and then solved using a general purpose solver (e.g., Cplex). The aim of the approach is to cancel and delay as few train services as possible by considering the reduced capacity and by deciding which train services should be delayed and/or cancelled. The order of train services on tracks can be switched and rolling stock turnings can be adapted.

Greedy approach

The greedy approach, which is used in case the rolling stock or crew rescheduling module was unable to cover all trips with rolling stock or crew, respectively, is quite basic. It cancels all

trips that are not covered by rolling stock or crew. If the rolling stock module was unable to cover certain trips, all other trips are included in a feasible rolling stock circulation. If we cancel the tasks that could not be covered, then, by construction, we obtain a timetable that is feasibly covered by rolling stock. Similarly, if the timetable algorithm was executed because some trips could not be covered by the crew, and these trips are cancelled, we obtain a timetable in which all tasks are covered by crew. If, thereafter, the rolling stock module is able to cover all remaining trips, we can skip the subsequent crew rescheduling approach and we are done. In this way, we might possibly save one step in the next iteration.

Our choice for the greedy approach also implies that the iterative algorithm always terminates. In every iteration, except for the first one, it holds that another iteration is only performed if at least one trip is not covered by rolling stock in the rolling stock rescheduling step and, after cancelling this trip with the greedy approach, another trip is not covered by crew in the crew rescheduling step. Otherwise, the framework terminates with a globally feasible solution. So, in every iteration, at least two trips are cancelled. As a consequence, after $\frac{\#\text{trips}}{2}$ iterations all trips are cancelled, and this is a feasible solution as well. In this situation, every rolling stock unit ends up at its location when the disruption starts. We assume that all units can be moved to their desired location during the night. Similarly, all crew members have to be sent to their depot by a taxi. Our models assume that these solutions are feasible, even though they carry high penalty costs. However, in most cases a feasible solution is obtained already after a few iterations, as will be shown in our computational results.

3.4 Rolling Stock Rescheduling

The rolling stock rescheduling module implements the approach of Nielsen et al. (2012). The rolling stock rescheduling problem is formulated as a multi-commodity flow model. The aim is to appoint rolling stock compositions to trips and make sure that there is enough capacity for all passenger demand. The objective is to minimize the number of non-covered trips and the deviation from the original plan. Decision variables in the model indicate which composition is assigned to each trip. Furthermore, the model contains decision variables to indicate the rolling stock composition changes taking place between two consecutive trips. Not all composition changes are allowed. For example, for a composition consisting of three different units, it is impossible to uncouple only the middle unit from the composition. The model is solved using a general purpose solver (e.g., Cplex).

The number of available rolling stock units is given as input to the module. In order to get some flexibility, the number of rolling stock units which need to be parked at each station during the night is not fixed. However, it is heavily penalized if at a station less units are available at the end of the day in the new schedule. Such end-of-day unbalances require additional deadheading trips during the night. Similarly, additional shunting movements are penalized. Every new shunting movement must be communicated to local shunting crew and must then be scheduled in between other shunting work. We want to minimize the additional work for local shunting crew, and therefore penalize any new shunting movements.

3.5 Crew Rescheduling

The approach of Veelenturf et al. (2012) is used for rescheduling the crew. This approach is based on Potthoff et al. (2010) and combines column generation with Lagrangian relaxation. It assigns new duties to crew members such that as many tasks as possible are covered by

the duties of the crew members. Furthermore, this approach allows to delay certain tasks by a few minutes in order to reduce the number of non-covered tasks. However, in this paper we do not use the possibility to delay certain tasks.

The crew rescheduling problem is formulated as a Set Covering Problem. Here, the duty of each crew member must be replaced by a new duty. The model contains decision variables that indicate which replacement duty is assigned to each crew member. Because the original duties can be replaced by many new duties, column generation is applied to generate promising replacement duties. Besides the main objective of covering as many tasks as possible, another objective is to have that the new duties deviate as little as possible from the original duties. Every deviation in the duties should be communicated to the crew members. This takes time and could lead to errors in practice.

In order to speed up the solution process, only a subset of the crew members is considered. In particular, the crew members whose duty became infeasible due to the disruption and the crew members in the neighbourhood of the disruption are included in the problem. The duties of the other crew members are fixed. In an iterative way, other neighbourhoods are explored as long as non-covered tasks are left.

4 Computational Experiments

In this section, we describe the computational experiments that we have executed to assess the performance of the iterative framework. We first describe the cases that we have considered and then discuss the results we have obtained. The iterative framework has been implemented in Java using Eclipse Kepler. All computational tests are performed on a desktop with an Intel Quad Core i7 processor and 4GB of RAM. We used Cplex 12.6 as solver for the MIP models.

4.1 Case description

We have used the Dutch railway system to test our iterative framework. We have obtained the timetable, rolling stock data, and the crew schedule for a specific day in June 2012 from Netherlands Railways.

In Figure 2, a picture of the Dutch railway network is shown. The solid lines represent the railway network that is operated by Netherlands Railways. The dotted lines are operated by other railway operators. In general, the timetable is half-hourly periodic, with some exceptions in the rural areas. This means that the majority of long distance and regional train services run once every thirty minutes. On some parts of the network, two different train services share a part of their route between two stations. It follows that a train service is operated every 15 minutes between these stations.

For the timetabling stage, we use only part of the railway network and require that all train services run as planned outside this region. This approach is in line with current practice and follows the literature on this topic, see, e.g., Veelenturf et al. (2016a) and Louwerse and Huisman (2014). The region that we consider in the timetabling phase is depicted by the circle in the figure and in more detail in Figure 3. In the latter figure also the number of tracks within a station and the number of tracks between stations are displayed. In this region, 116 train departures from stations depicted in Figure 3 are scheduled per hour in the timetable.
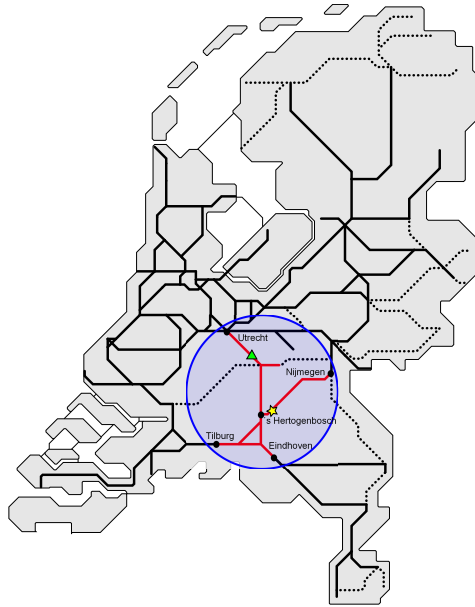
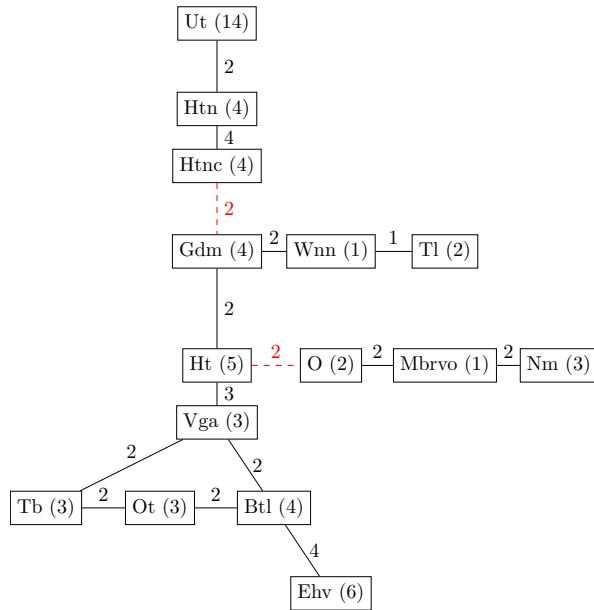Figure 2: Map of the railway network in the Netherlands (2012).

Figure 3: Graph of the railway network taken into account when rescheduling the timetable. Rectangles are the nodes and correspond to stations or important junctions. For each station, the number of tracks within the station is specified. Numbers at the edges indicate how many tracks connect two stations. The dashed edges represent the locations of the disruptions.

We consider disruptions at two different locations: either between 's-Hertogenbosch (Ht) and Nijmegen (Nm) (indicated by the star in Figure 2) or between 's-Hertogenbosch and Utrecht (Ut) (indicated by the triangle). The dashed edges in Figure 3 also indicate the locations of the two disruptions. On both locations, there is a double track between the stations. We consider both a complete and a partial blockage. In a complete blockage, both tracks are blocked and it is impossible to operate any train services between these stations. In a partial blockage, only one of the tracks is blocked, which allows train services to cross the disrupted region.

In the railway network considered when rescheduling the timetable, there are in total 15 nodes (stations or important junctions), with the number of tracks in the nodes varying from 1 until 14. Only stations or junctions are considered where ordering decisions between trains are necessary. With ordering decisions between trains we mean the decision which train utilizes the specific track before the other. As a consequence, the considered stations and junctions are the ones to influence the macroscopic routing options of the train services. There are in total 5 stations (Ut, Gdm, Ht, Ehv and Nm) that have a shunting yard. We assume that all shunting yards have a sufficient number of tracks to store rolling stock units.

To analyse how the performance of our approach depends on the characteristics of the disruption, we consider different start times and different durations of the disruption. The start time of the duration varies between 7:00 AM and 5:00 PM in steps of ten minutes, giving us 61 different start times. The duration of the disruption is either 60, 80, 100, or 120 minutes. Recall that we assume that the duration of the disruption is known directly at the start of the disruption.

For the rolling stock and crew rescheduling phase, we consider the complete railway network. We no longer focus on a specific part of the network alone. Furthermore, for the rolling stock rescheduling phase, note that in the Netherlands we distinguish between long distance train services and regional train services. A regional train service is scheduled to stop at every station, while a long distance train service only dwells at the larger stations. The rolling stock units available for long distance train services are different from those available for regional train services. For example, rolling stock units for regional train services accelerate faster. In our experiments, it is not allowed to use rolling stock meant for long distance train services for a regional train service or vice versa. This allows us to decompose the rolling stock rescheduling problem into two independent problems: one rolling stock rescheduling problem for the long distance train services and one for the regional train services.

In the crew rescheduling step we assume that drivers and conductors work in pairs and that one of each is required per train service. As a consequence, by rescheduling the drivers we have immediately rescheduled the conductors.

4.2   Implementation

The models discussed in Sections 3.3-3.5 require certain settings and input data. In this section, we discuss the details of the actual implementation of the iterative framework. The main objective in our experiments is to minimize the total duration of cancelled train services. Therefore, in each of the three approaches, this will be the individual main objective. For the rolling stock and crew rescheduling this is done indirectly by focusing on covering as many trips as possible with rolling stock and crew. If no rolling stock or crew is assigned to a certain trip, this trip will be cancelled by the timetable rescheduling. The penalties for not assigning rolling stock or crew to a certain trip depend on the duration of the trip. The

values of the other penalties and parameters are set at the values proposed in the original publications on these algorithms. Note that the algorithms have been developed and tested in close collaboration with NS.

4.2.1 Settings timetable rescheduling

The first time the timetable is rescheduled, we apply the approach developed by Veelenturf et al. (2016a). Each next iteration, the timetable rescheduling module is called because either the rolling stock or the crew rescheduling was not able to cover all trips. For these cases we implemented the greedy timetable rescheduling approach which cancels all trips for which no rolling stock or crew was found.

The timetable rescheduling approach of Veelenturf et al. (2016a) requires the original scheduled timetable as input. Every scheduled trip is represented by a corresponding departure and arrival event. Such an event contains the time when it takes place, the corresponding station where it takes place, and the scheduled track on which it takes place.

Next to the timetable, a description of the railway network is necessary. Furthermore, the model requires parameters specifying the regulations between train services. Within the stations a headway time of 2 minutes is considered in between two consecutive train services assigned to the same track. This headway time of 2 minutes also applies to two consecutive train services running in the same direction assigned to the same track in between stations. The headway time of train services using a single track in opposite directions is 0 minutes.

Detailed settings for the rolling stock part of the timetable rescheduling module are necessary as input as well. When a train service ends, the rolling stock of that train service may be used by a starting train service at the same location. This is called a turning pattern. The minimum time between these two train services is set to 1 minute. If a rolling stock turning is chosen with a turn around time longer than 10 minutes and there is a shunting yard available, the rolling stock is shunted away. In such a situation, 5 minutes after the train service has ended, the rolling stock is shunted away, and 5 minutes before the new train service starts, the rolling stock is considered to be back at a station track.

In the objective function, a single cancelled minute is penalized by 50,000 since minimizing cancellations is the main objective. For each minute of arrival delay, a penalty of 1 applies. Furthermore, an event may be delayed by at most 8 minutes. Recall that a train service is operated every 15 minutes on some parts of the network. The maximum delay is set at half of this value. In case of a larger delay, the passengers can as well wait for the next train. Note that we only take direct train delays into account. Longer passenger delays, due to missed connections for example, are not considered.

Another penalty is set upon deviating from a preferred turning pattern. For example, turning patterns of the same series are preferred. A list of preferred turning patterns is given as input and using a different turning pattern is penalized by 10.

Finally, the model requires as input a duration that specifies from what time onwards the timetable should be equal to the original timetable. We set this duration to 60 minutes. This means that from 60 minutes after the disruption has ended onwards, all trains must be operated as planned again.

All penalty values for the sophisticated timetable rescheduling approach are summarized in Table 1.

Description	Penalty
Cancelled minute	50,000
One minute arrival delay	1
Turning pattern deviation	10

Table 1: Penalties timetable rescheduling approach.

4.2.2 Settings Rolling Stock Rescheduling

The rolling stock rescheduling module based on Nielsen et al. (2012) uses many settings. In our experiments, there are four different rolling stock types available. There are two types for the long distance train services, namely one with 3 carriages and one with 4 carriages, and two types for the regional train services, also consisting of 3 and 4 carriages. Rolling stock units can be coupled to each other to form a rolling stock composition. In this way more capacity can be appointed to a trip. Only units of the same type are allowed to be coupled into a composition. The maximum length of a composition is 15 carriages.

At the start of the day, each station with a shunting yard contains a starting inventory of rolling stock units. This starting inventory denotes the number of available rolling stock units per rolling stock type at that station. Next to the starting inventory, the desired end-of-day inventory is needed per station. This is the amount of rolling stock units of a certain type that preferably is present at the end of the planning period at the corresponding station. We set a penalty of 100 per unit deviation from the desired end-of-day inventory.

Furthermore, an original rolling stock circulation is required for both the long distance and the regional train services. This original circulation contains a list of trips, where a trip is defined as a part of the train service between two stations where the composition may be changed. Every trip has the following characteristics: departure station, arrival station, departure time, arrival time, successor of the trip (turning pattern), and the originally appointed composition. The last information which the model requires is the start time of the disruption.

The largest penalty is set upon not covering a trip. A trip is not covered in the rolling stock rescheduling phase if no composition can be appointed to the trip. The penalty for not covering a trip equals 100,000 plus 1,000 times the duration of the trip in minutes. In this way we minimize the number of cancelled train services, and if we have to cancel a train service, then we prefer to cancel the train service with the shortest duration.

Next, we want to minimize the differences between the rescheduled rolling stock circulation and the original rolling stock circulation. First of all, the deviation in the number of carriages between the two circulations is penalized. A penalty of 1,000 is given per additional carriage on a trip and a penalty of 10,000 is given for every missing carriage on a trip in the rescheduled rolling stock circulation in comparison with the original rolling stock circulation. The penalty for missing a carriage is larger than the penalty for an additional carriage, because one missing carriage means a capacity reduction of approximately 100 seats for passengers on that trip. By doing so, we implicitly assume that the passenger demand used for the original circulation is still accurate. Another option would be to incorporate how passengers react when a disruption occurs. This issue is discussed, for example, by van der Hurk (2015). In that case, one can penalize seat capacity shortages directly when rescheduling the rolling stock circulations.

The final penalty is set upon deviating from the original shunting plan. If an originally

planned shunting movement (either coupling or uncoupling) is cancelled, then there is a penalty of 100 for not performing that shunting movement. A penalty of 1000 is set upon newly added shunting movements. For these shunting movements, new shunting crew must be arranged, which could be a lot of work, so the penalty is larger than the penalty for cancelling a shunting movement.

All penalties used in the rolling stock rescheduling approach are summarized in Table 2.

Description	Penalty
Cancelled trip	$100{,}000 + 1{,}000 \cdot$ duration of the trip
End-of-day balance deviation	100
Positive carriage deviation	1,000
Negative carriage deviation	10,000
Cancelled shunting movement	100
New shunting movement	1,000

Table 2: Penalties rolling stock rescheduling approach.

In order to be able to solve the rolling stock rescheduling problem in short time, we use a rolling horizon. Note that all trips before the start of the disruption are fixed, so our complete planning horizon is from the start of the disruption (t^s) up to the end of the day (t^∞). Solving the rolling stock rescheduling problem with the complete planning horizon can take long, so we split it by means of a dynamic planning horizon. The first part is from the start of the disruption up to time $t^s + \frac{t^\infty - t^s}{2} = \frac{t^s + t^\infty}{2}$, this is exactly halfway the complete planning horizon. The second part is from $\frac{t^s + t^\infty}{2}$ up to the end of the complete planning horizon. The solution of the first part is given as input to the second part, such that a feasible rolling stock circulation is found for the complete planning horizon.

Furthermore, we use a fixed computation time limit of 5 minutes per horizon. The optimal rolling stock circulation, with respect to our penalties, is found relatively fast. However, proving that this solution is optimal might take time. That is why we use a computation time limit of 5 minutes per horizon. In this time limit the optimal solution is almost always found, but not yet proven to be optimal.

4.2.3 Settings Crew Rescheduling

The crew rescheduling approach of Veelenturf et al. (2012) uses penalties for not covering a task and for deviations in comparison to the original schedule. In the schedule, we distinguish between three types of tasks: tasks corresponding to operating a train service with a different start and end location, tasks corresponding to operating a train service with the same start and end location (operating the same rolling stock back and forth), and tasks which do not have to do anything with operating train services (e.g., training tasks). Since the aim is to operate as many train services as possible, the first two types of tasks get heavily penalized if they are not covered, while the third one is penalized less.

The penalty for not covering a task which is not related to operating train services, called local task, equals 250. For not covering a task related to operating train services, we test two settings. In the first setting, all tasks are treated equally and the penalty for not covering a task is equal to 20,000 plus 100 times the duration of the task. In the second setting

14

we make a distinction between tasks in which the start and end location are the same (so called *AA-tasks*) and tasks in which the end location is different from the start location (so called *AB-tasks*). It can be argued that the rolling stock schedule remains feasible if an AA-task is cancelled. However, if an AB-task is cancelled, rolling stock rescheduling is definitely necessary. Therefore, in the second setting, we prefer not covering an AA-task over not covering an AB-task. For not covering an AA-task the penalty will be 3,000 plus 100 times the duration of the task. For not covering an AB-task the penalty remains 20,000 plus 100 times the duration of the task. It is expected that this second option leads to more cancellations in the crew rescheduling step but to less cancellations in the subsequent rolling stock rescheduling step.

The fixed costs for changing a duty equals 400 and for each new task in a duty a penalty of 50 applies. Each new transfer between tasks which was not present in any duty in the original crew schedule is penalized by 1. If the driver is directly sent home by a taxi (since no replacement duty is available which complies with the rules), a penalty of 3,000 is used.

The new duties may not take longer than the original duties and at maximum 5.5 hours may pass without a break of at least 30 minutes. The transfer time in between tasks on different rolling stock compositions equals 10 minutes.

The penalty values used for crew rescheduling are different from the ones used for rescheduling the rolling stock. The penalty values are all commonly used in literature. All three modules have the same overall objective: cancelling as little train services as possible.

The penalties used in the crew rescheduling approach are summarized in Table 3.

Description	Penalty
Cancelled task setting 1	$20{,}000 + 100 \cdot$ duration of the trip
Cancelled AA task setting 2	$3{,}000 + 100 \cdot$ duration of the trip
Cancelled AB task setting 2	$20{,}000 + 100 \cdot$ duration of the trip
Cancelled local task	250
Change duty	400
New task in duty	50
New transfer in duty	1
Send driver home by taxi	3,000

Table 3: Penalties crew rescheduling approach.

The approach of Veelenturf et al. (2012) also has an option to slightly delay tasks to have less tasks which cannot be covered. In our experiments we did not allow these delays, since then the crew rescheduling is interfering with the timetable rescheduling.

4.3 Experiments

In this section, we present the results for complete and partial blockages on various settings of the framework. First, we test the general framework and discuss the associated results. Thereafter, the differences are presented between the general framework and the variant where the order of the rolling stock and crew rescheduling step are switched. Then, we test whether having a lower penalty for not covering AA-tasks leads to less cancelled trips overall. We end this section with a discussion of some practical issues.

4.3.1 Results of the general framework

First, the results of the general framework are presented. Table 4 gives an overview of the number of cancelled train services and the average total duration of the cancelled train services. These numbers include the train services that inevitably need to be cancelled because they are scheduled on the blocked tracks. Recall that for each location, duration, and type of disruption, we have 61 possible start times of the disruption. Each number in the table represents an average over these 61 instances. That leads to a total of 976 instances to test the framework upon, because there are two locations where a disruption occurs, four different durations, two different types (complete and partial), and 61 different start times.

Type	Duration (min)	Cancelled trips			Cancelled minutes		
		TTR	RSR	CR	TTR	RSR	CR
Ht-Ut Complete	60	14.33	0.20	0.75	284.46	5.08	19.30
	80	19.64	0.15	1.23	413.39	2.72	34.05
	100	25.36	0.13	1.72	526.71	2.28	47.38
	120	31.00	0.16	1.97	638.54	3.05	53.64
Ht-Ut Partial	60	5.33	0.03	0.89	88.51	0.56	21.39
	80	9.67	0.16	1.41	136.80	2.80	36.56
	100	9.34	0.05	0.77	167.95	0.69	22.26
	120	12.00	0.41	1.54	209.36	11.92	38.89
Ht-O Complete	60	8.00	0.03	0.84	124.80	0.43	27.44
	80	11.69	0.39	0.98	214.98	12.90	36.85
	100	13.00	0.13	1.39	226.15	2.20	48.66
	120	16.00	0.08	1.62	249.61	1.21	50.57
Ht-O Partial	60	4.00	0.02	0.61	59.51	0.21	18.54
	80	5.69	0.28	0.74	106.31	9.87	23.43
	100	6.66	0.00	0.93	120.71	0.00	32.15
	120	8.00	0.00	0.85	141.61	0.00	29.00
All cases		12.48	0.14	1.14	231.84	3.50	33.76

Table 4: Results of the General Framework. *The first column denotes the location and the type of disruption and the second column the duration of the disruption. The third, fourth and fifth column denote the average number of cancelled train services in timetable, rolling stock, and crew rescheduling. The sixth, seventh, and eighth column show the average total duration of all the cancelled trips in timetable, rolling stock, and crew rescheduling in minutes.*

First consider the number of cancelled train services. As expected, most of the train services are cancelled in the timetabling phase. Remember that in case of a complete blockage, all tracks are blocked. As a result, the module cancels most of the train services operated on those tracks. If these train services would not be cancelled, they would queue up in the railway system, causing knock-on effects over the whole country. As a result, we observe more cancellations for complete blockages than in case of partial blockages. Furthermore, as expected, the number of cancelled train services when rescheduling the timetable increases if the duration of the disruption increases. This holds for both complete and partial blockages.

Secondly, we observe that train services are cancelled in the rolling stock phase only rarely.

This can be attributed to the timetabling algorithm, which increases the probability that a feasible rolling stock schedule exists that does not need to cancel any additional train services. Consequently, almost no train services are cancelled in the first iteration of the rolling stock rescheduling module.

This does not hold for the crew rescheduling stage. On average 1 to 2 train services are cancelled in that stage. In case of a complete blockage, the average number of tasks for which no crew can be found is increasing in the duration of the disruption. We do not see this pattern for the partial blockages. One explanation could be that the longer tracks are completely blocked, the more difficult it becomes to get the crew members home on time. In case of partial blockages, it is easier to get crew members home since still some train services are operated. However, this does not mean that we can conclude that partial blockages lead to less or more cancelled train services due to lack of crew in general. On the one hand, partial blockages cause less gaps in duties by cancelled train services due to timetable rescheduling, but on the other hand due to the lack of gaps there is less buffer to adapt duties.

The numbers of cancelled minutes of train services follow a similar pattern as the number of cancelled train services for both complete and partial blockages.

Next, we consider the iterative behaviour of the framework. In Table 5, we indicate the amount of instances that turn out to be feasible after executing each module for all iterations. As can be seen, for both the complete and partial blockages, at least 40% of the instances are solved in one iteration and 99% in two iterations. Furthermore, among the instances which stop in the second iteration, most have already been stopped after the rolling stock rescheduling step. In other words, for more than 90% of the instances the crew rescheduling step is performed only once. All instances are solved in at most three rolling stock and three crew rescheduling steps, thus after three iterations.

Complete blockages		
Iteration	Cumulatively stopped after RSR	Cumulatively stopped after CR
1	-	193 (40%)
2	450 (92%)	482 (99%)
3	488 (100%)	
Partial blockages		
Iteration	Cumulatively stopped after RSR	Cumulatively stopped after CR
1	-	228 (47%)
2	475 (97%)	485 (99%)
3	486 (99%)	488 (100%)

Table 5: Iterative behaviour of the general framework. *The first column denotes the iteration. The second column gives the number of instances for which a feasible overall solution is found after rescheduling the rolling stock. The third column gives the number of instances for which a feasible overall solution is found after rescheduling the crew.*

The computation times are presented in Figures 4 and 5. In both figures, the left side gives an overview of the average computation time for each of the modules in the iterative framework. The computation time of a single instance of, for example, the crew rescheduling module is the total time it takes to reschedule the crew (so all iterations combined). On the right side a histogram of the total computation time is shown. It gives an overview of

the percentage of instances that are solved within 0-3, 3-5, 5-7, 7-9, and 9-11 minutes. Note that the regional and long distance rolling stock rescheduling step can be solved in parallel. However, we have solved them sequentially. As a consequence, the total time it takes to solve an instance is the sum of the computation times for timetable rescheduling (TTR), rolling stock rescheduling for regional (RSR R) and long distance train services (RSR L), and crew rescheduling (CR). For both complete and partial disruptions more than 80% of the cases are solved within 5 minutes and less than 6% of the runs take more than 7 minutes. Note that the total computation time averaged over all instances is almost similar for complete and partial blockages: 3.8 and 3.9 minutes, respectively.

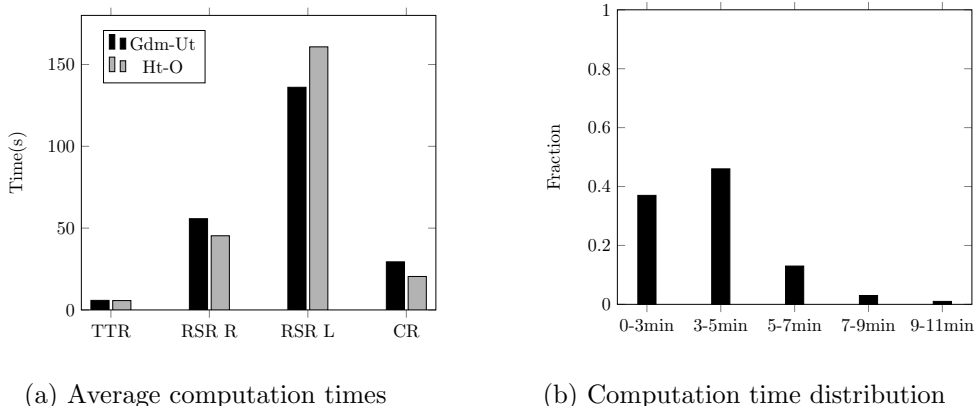(a) Average computation times

(b) Computation time distribution

Figure 4: Computation times for complete blockages in the general framework. Here R abbreviates the regional train services and L the long distance train services.

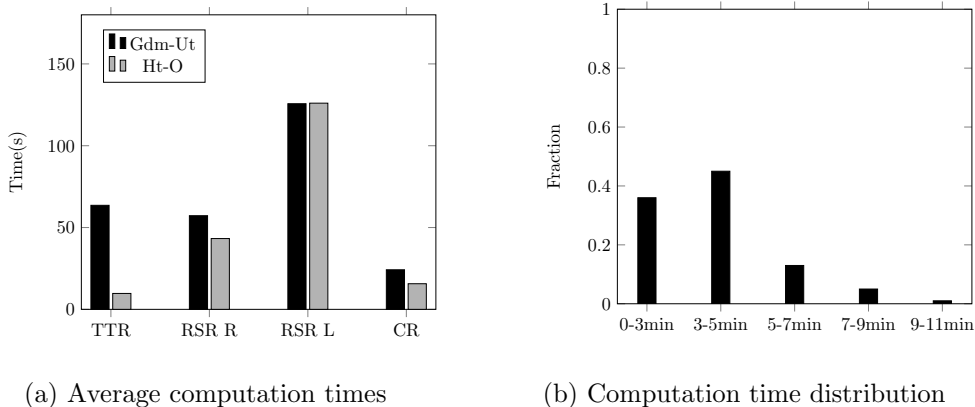(a) Average computation times

(b) Computation time distribution

Figure 5: Computation times for partial blockages in the general framework. Here R abbreviates the regional train services and L the long distance train services.

Figure 6 shows the average computation time per step in the framework for complete (left) and partial (right) blockages. As explained above, at most three iterations are needed within the framework. This explains why the rolling stock and crew rescheduling phases appear three times in the figure. Note that the sophisticated approach for timetabling is only applied

18

in the first iteration. The computation time for this approach is reported in the figure. In the second and third iteration, the greedy approach for timetabling is applied. The greedy approach cancels the trips without rolling stock or crew. This can be done instantaneously. Thus, the computation times for the greedy approach are all 0 and are not reported in the figure. There is no third iteration necessary for crew rescheduling for any of the instances during a complete blockage, so we have left that one out of the figure. Also, there was no third iteration required for any of the Ht-O instances with partial blockages and not for any of the regional train instances with a complete blockage between Ht-O.

Note that for the second and third iterations, the average computation time is computed over the instances for which a second or third iteration was required, respectively. It can be seen that in case of a complete blockage, the timetable rescheduling step takes a couple of seconds, while in case of a partial blockage, it takes about half a minute. The average computation time spent in crew rescheduling is less than half a minute and is in case of a partial blockage lower than in case of a complete blockage. Rescheduling the rolling stock takes most of the time: The average time required is a couple of minutes and again less for instances with partial blockages than for instances with complete blockages.

Summarizing, the total average computation time is the same for partial and complete blockages. However, the division of the computation time over the rescheduling steps differs. Partial blockages require more time for the timetable rescheduling, but that is compensated by requiring less time for rescheduling the rolling stock and crew.

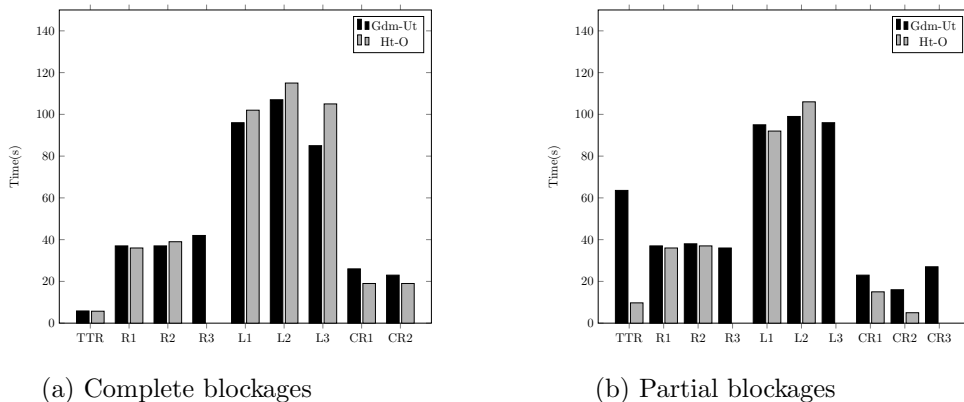(a) Complete blockages   (b) Partial blockages

Figure 6: Computation times for each of the steps in the general framework. Here R abbreviates the regional train services and L the long distance train services. $R1$ abbreviates then the first iteration for the regional train services, etc.

4.3.2 Crew First Variant

As discussed in Section 3, the order of the rescheduling modules can be changed in the iterative framework. In this section, we consider the variant where we reschedule the crew before rescheduling the rolling stock in each iteration. In practice, this is not always possible. First, buffer time between two train services is not required if the train services are operated with the same rolling stock. Thus, the minimal buffer times depend on the rolling stock circulation. Secondly, drivers have a license to operate a subset of all rolling stock types. Again, the rolling stock type for all trips is only known after rescheduling the rolling stock.

However, in this section, we assume that the crew schedule can be generated independently of the rolling stock schedule. In this case we are able to reschedule the crew before the rolling stock. This special variant of our framework will be referred to as the Crew First Variant. The Crew First Variant is interesting, because some rolling stock constraints are taken into account in the timetable rescheduling step of Veelenturf et al. (2016a). Consequently, less iterations might be required to find an overall feasible solution. Therefore, we have tested this setting. The results are presented in Tables 6 and 7 and Figures 7 and 8.

Type	Duration (min)	Cancelled trips			Cancelled minutes		
		TTR	RSR	CR	TTR	RSR	CR
Ht-Ut Complete	60	14.33	0.20	0.79	284.46	5.08	19.77
	80	19.64	0.10	1.23	413.39	1.69	34.05
	100	25.36	0.13	1.72	526.71	2.28	47.38
	120	31.00	0.12	1.97	638.54	2.02	53.64
Ht-Ut Partial	60	5.33	0.03	0.89	88.51	0.56	21.39
	80	9.67	0.16	1.41	136.80	2.80	36.56
	100	9.34	0.05	0.77	167.95	0.69	22.26
	120	12.0	0.33	1.71	209.36	9.00	43.02
Ht-O Complete	60	8.00	0.03	0.84	124.80	0.43	27.44
	80	11.69	0.39	1.00	214.98	12.89	37.25
	100	13.00	0.05	1.39	226.15	0.74	48.66
	120	16.00	0.08	1.62	249.61	1.21	50.57
Ht-O Partial	60	4.00	0.02	0.61	59.51	0.21	18.54
	80	5.69	0.26	0.79	106.31	9.31	24.85
	100	6.66	0.00	0.93	120.71	0.00	32.15
	120	8.00	0.00	0.85	141.61	0.00	29.00
All cases		12.48	0.12	1.16	231.84	3.06	34.16

Table 6: Results of the Crew First Variant. *See Table 4 for the description of the columns.*

In Table 6 the number of cancelled train services and the average duration of the cancelled train services are presented. There is not much difference in terms of the number of cancellations and the duration of the cancelled train services in comparison with the general framework. However, if we look at the number of iterations in Table 7, we see that at most two rolling stock iterations are necessary now. Furthermore, the percentage of instances solved in one iteration is very large (about 90%). In the general framework, about 50 to 60 percent of the instances needed at least two rolling stock rescheduling steps (see Table 5). With the Crew First Variant, only one percent of the instances need a second rolling stock rescheduling step. As the rolling stock rescheduling step is the most time consuming step, this leads to a decrease in the average computation time, as can be seen in Figures 7 and 8. Here, both the computation times of the General Framework and of the Crew First Variant are displayed. We note that the average computation time equals 2.8 and 3.2 minutes, respectively, for the instances with complete and partial blockages in the Crew First Variant. This is about one minute (about 25%) faster than in the general framework. The distribution of the computation times per step in the Crew First Variant is the same as in the general framework (see Figure 6), therefore we do not show such a figure again.

	Complete blockages	
Iteration	Cumulatively stopped after CR	Cumulatively stopped after RSR
1	-	428 (88%)
2	481 (99%)	488 (100%)
	Partial blockages	
Iteration	Cumulatively stopped after CR	Cumulatively stopped after RSR
1	-	451 (92%)
2	483 (99%)	486 (99%)
3	488 (100%)	

Table 7: Iterative behaviour of the Crew First Variant. *See Table 5 for a description of the columns.*
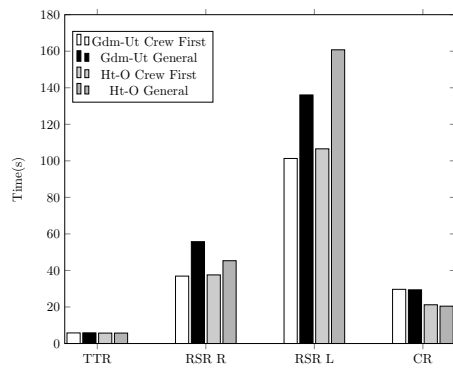
(a) Average computation times
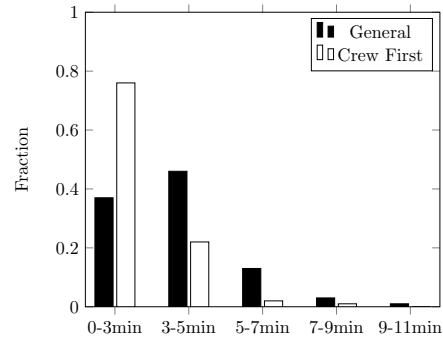
(b) Computation time distribution

Figure 7: Computation times for complete blockages in the Crew First Variant. Here R abbreviates the regional train services and L the long distance train services.

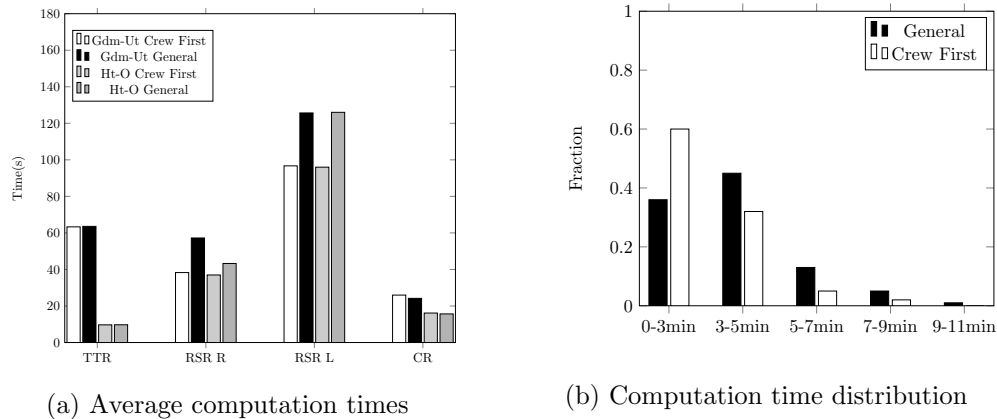(a) Average computation times
(b) Computation time distribution

Figure 8: Computation times for partial blockages in the Crew First Variant. Here R abbreviates the regional train services and L the long distance train services.

4.3.3 Different Costs Variant

In the previous section, we demonstrated that computation times could be reduced by changing the order of the rolling stock and crew rescheduling steps. This might not be applicable in reality because it depends on assumptions which do not always hold in practice.

Another idea to save time for the rolling stock rescheduling phase is to consider rolling stock properties already in the crew rescheduling module. If a crew task with a different start and end location (AB-task) is cancelled, it causes a gap and leads to an infeasibility in a rolling stock duty. However, if a crew task with the same start and end location (AA-task) is cancelled, it is assumed that the gap in the rolling stock duty does not make the duty infeasible. During the crew rescheduling phase we could aim to prefer cancellations of AA-tasks over cancellations of AB-tasks, by having different penalties for not covering these tasks (as discussed in Section 4.2.3). This could result in less cancellations in the next rolling stock rescheduling step. We refer to this as the Different Costs Variant. In the Different Costs Variant, we first reschedule the rolling stock, and then the crew in each iteration. Hence, the costs are changed compared to the base case discussed in Section 4.3.1.

22

		Cancelled trips			Cancelled minutes		
Type	Duration (min)	TTR	RSR	CR	TTR	RSR	CR
	60	14.32	0.21	0.74	284.46	5.36	20.97
Ht-Ut	80	19.64	0.16	1.21	413.39	3.00	35.26
Complete	100	25.36	0.13	1.70	526.71	2.30	52.71
	120	31.00	0.20	1.97	638.54	3.62	56.84
	60	5.33	0.03	0.87	88.51	0.56	29.18
Ht-Ut	80	9.67	0.16	1.36	136.80	2.80	42.82
Partial	100	9.34	0.07	0.77	167.95	0.97	25.51
	120	12.00	0.41	1.61	209.36	11.92	49.66
	60	8.00	0.02	0.89	124.80	0.21	32.66
Ht-O	80	11.69	0.39	1.02	214.98	12.90	40.12
Complete	100	13.00	0.13	1.48	226.15	2.20	56.67
	120	16.00	0.07	1.69	249.61	0.95	57.97
	60	4.00	0.02	0.62	59.51	0.21	19.84
Ht-O	80	5.69	0.28	0.80	106.31	9.87	30.79
Partial	100	6.66	0.00	1.05	120.71	0.00	41.46
	120	8.00	0.00	0.97	141.61	0.00	39.90
All cases		12.48	0.14	1.17	231.48	3.55	39.52

Table 8: Results of the Different Costs Variant. *See Table 4 for a description of the columns.*

In Table 8, the results of the Different Costs Variant are presented. As expected, the number of cancelled train services and corresponding minutes in the crew rescheduling step have increased in comparison to the general framework. This is due to the fact that it is now cheaper to cancel AA-tasks. The percentage of cancelled AA-tasks with respect to all tasks cancelled due to lack of crew has increased from 28% to 38%. However, this has not led to a decrease in the number of cancelled tasks due to lack of rolling stock. Instead, it has led to a very slight increase in the duration of the cancelled trips due to lack of rolling stock. Most likely this is caused by the fact that in total more crew tasks are cancelled.

In terms of computation times, there are no significant differences as can be seen in Figure 9 and Figure 10.

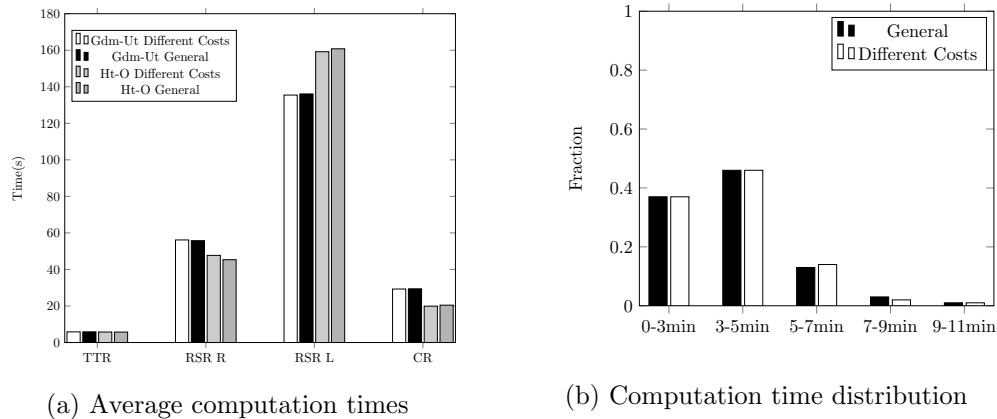(a) Average computation times

(b) Computation time distribution

Figure 9: Computation times for complete blockages in the Different Costs Variant. Here R abbreviates the regional train services and L the long distance train services.
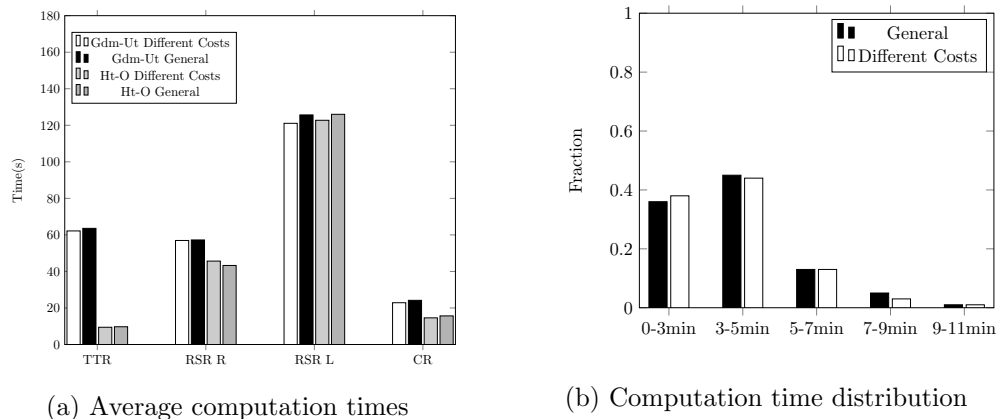

(a) Average computation times

(b) Computation time distribution

Figure 10: Computation times for partial blockages in the Different Costs Variant. Here R abbreviates the regional train services and L the long distance train services.

4.4 Practical considerations

Within Netherlands Railways, there is a lot of know-how about rescheduling during disruptions. These experiences in rescheduling result in several interesting findings, which we like to investigate now we have the relevant data.

4.4.1 Influence of start time disruption on crew rescheduling

One of these interesting findings arising from practice is that if a disruption occurs between 12:00 and 14:00, many more train services get cancelled due to lack of crew than at any other time during the day. We decided to check whether this is true. Figure 11 shows the distribution of the number of cancelled train services due to lack of crew. In this figure, the start time of the disruption is plotted on the horizontal axis. On the vertical axis, we show the number of tasks that cannot be covered by crew. As can be seen, both for complete and partial

24

blockages, the peak is between 11:00 and 14:00. So the start time of the disruption indeed has a large influence on the number of train services being cancelled by crew rescheduling. This is probably due to the fact that there are many duties that end around 14:00 at Netherlands Railways, which makes it difficult to get the involved crew members back on time at their end location. This finding could help with allocating and scheduling the reserve crew.
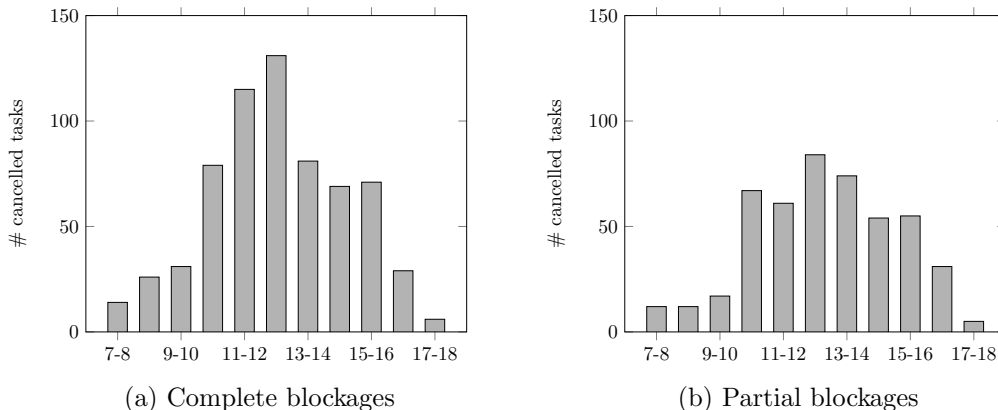

(a) Complete blockages   (b) Partial blockages

Figure 11: Spread of cancelled train services due to lack of crew members

4.4.2 Redundant iterative steps removed

A second finding is that if in the crew rescheduling step only AA-tasks get cancelled, we no longer have to reschedule the rolling stock again. The start and end location of an AA-task is the same, so cancelling an AA-task can probably be solved in the rolling stock circulation fairly easily. As a consequence, the rolling stock circulation should remain feasible. Recall that this was the reason to consider different cost penalties for AA-tasks and AB-tasks in Section 4.3.3. Our results demonstrate that this is indeed the case. If only AA-tasks get cancelled in the crew rescheduling module, then the next rolling stock rescheduling step is redundant.

By not performing the redundant rolling stock step, we can save one iteration. As a result, in total we are able to save 42 iterations in case of a complete blockage, and 30 iterations in case of a partial blockage in the General Framework. No iterations can be saved by not performing a redundant rolling stock rescheduling step in the Crew First Variant, because there were no redundant rolling stock rescheduling steps. For the Different Costs Variant, we are able to save in total 61 iterations in case of a complete blockage and 74 in case of partial partial blockages. Observe that we can indeed save more iterations in the Different Costs Variant.

Table 9 shows the average computation time required to solve the instances with and without the redundant rolling stock rescheduling step. As can be seen, on average around 10 seconds of computation time is saved for the instances with a full blockage and around 5 seconds for the partial blockages for both the General Framework and the Different Costs Variant.

Type	Module	General	Crew First	Different Costs
Complete	RSR L	290.59	207.87	285.48
No redundant	RSR R	96.42	74.47	95.90
	Total	448.41	344.71	442.16
Complete	RSR L	296.84	207.87	294.66
Redundant	RSR R	101.08	74.47	103.86
	Total	459.32	344.71	459.30
Partial	RSR L	250.52	192.69	241.93
No redundant	RSR R	96.40	75.27	92.21
	Total	460.06	383.06	443.21
Partial	RSR L	251.68	192.69	243.82
Redundant	RSR R	100.50	75.27	102.61
	Total	465.32	383.06	455.49

Table 9: Average computation times: *the first column denotes whether instances with a complete or partial blockage are solved and whether the redundant rolling stock step is performed or not. The second column denotes for which part of the framework we present the results (RSR L, RSR R, or the total framework). The third, fourth, and fifth column denote the average computation time required to solve the instances for different variants of the framework.*

Figures 12, 13, and 14 present the computation time distribution for the General Framework, the Crew First Variant and the Different Costs Variant, respectively. Every figure presents both the distribution when redundant steps are performed and when not. For both the General Framework and the Different Costs Variant, around 5%-10% more instances can now be solved within 0-3 minutes.
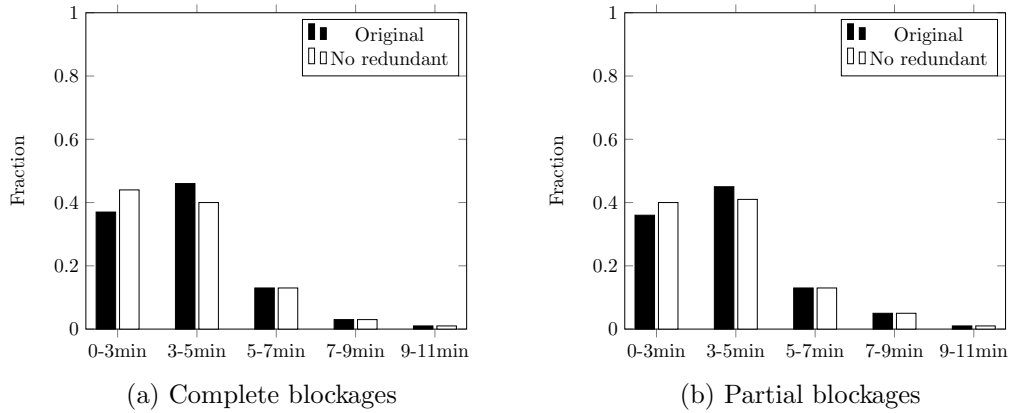

(a) Complete blockages

(b) Partial blockages

Figure 12: Total computation time distribution for the General Framework

(a) Complete blockages   (b) Partial blockages
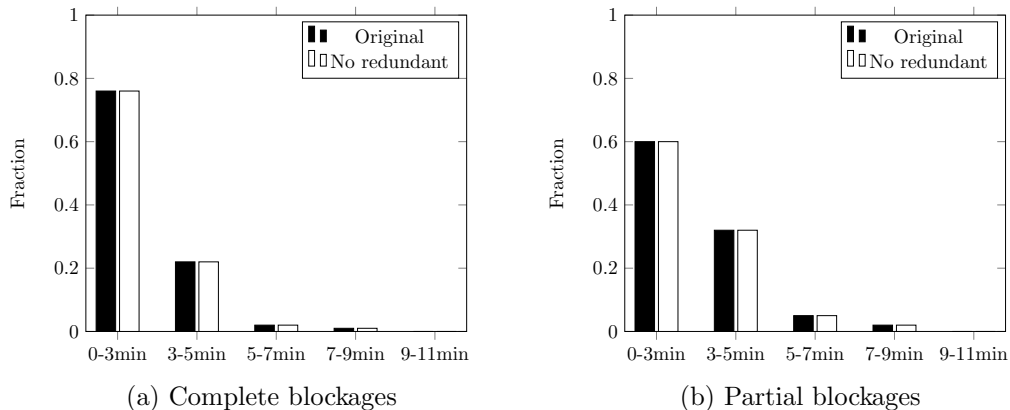
Figure 13: Total computation time distribution for the Crew First Variant.
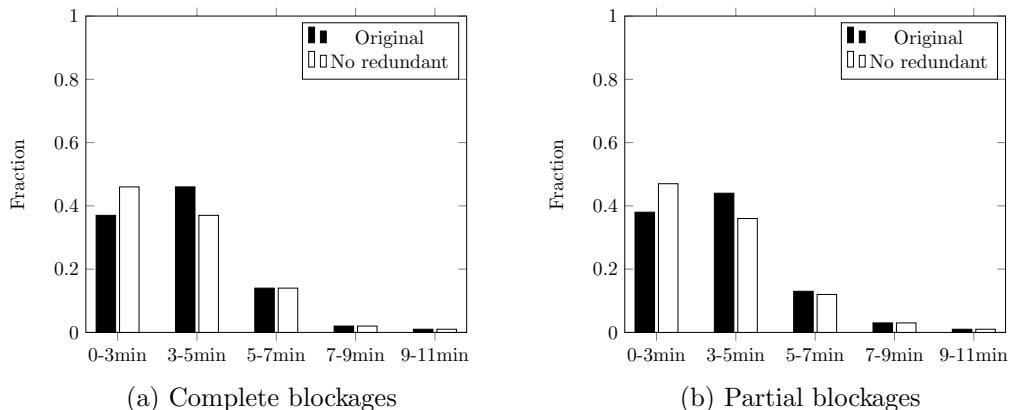

(a) Complete blockages   (b) Partial blockages

Figure 14: Total computation time distribution for the Different Costs Variant.

4.4.3 Evaluation of contingency plans

We have described our iterative framework as a tool for real-time rescheduling. However, it can also be applied in a different setting. NS and Prorail are currently implementing the framework to test the quality of their contingency plans. These contingency plans describe what to do in case of a disruption, such as a blockage of one or more tracks. If a track is blocked, the contingency plan prescribes which train services to cancel in the timetable, and how to adjust the turnings in the rolling stock circulation. After manually inserting the timetable from the contingency plan as a disposition timetable, the rolling stock and crew are rescheduled by the framework. This allows to investigate whether a feasible rolling stock and crew schedule exist for the given contingency plan, given a certain start time and duration of the disruption.

    If a satisfactory solution exists (i.e. all tasks from the disposition timetable in the contingency plan can be covered by rolling stock and crew) for a certain given percentage of possible start times and durations, the contingency plan is accepted. However, if this threshold percentage is not reached, the contingency planner has a detailed look at the cause of

these issues. For instance, it is possible that several tasks cannot be covered by rolling stock, because the total travel and turn around time in the contingency plan is larger than in the original timetable. In such a case, the contingency plan needs to be adjusted. In another situation, where, e.g., for a certain instance one task cannot be covered by a driver, it could be possible to cover this task by allowing a small violation of a labour rule. In such a case, the contingency plan will be accepted.

5   Conclusion

Most studies on disruption management in passenger railways focus on the rescheduling of one resource (timetable, crew or rolling stock) schedule at the time, see Cacchiani et al. (2014). However, it had not been investigated yet whether these approaches can be combined to find an overall feasible solution. Therefore, we presented an iterative framework considering all the resource schedules in this paper.

This framework has been tested with existing models on a large number of disruption scenarios of Netherlands Railways. The experiments demonstrate that with this framework a railway operator is able to find a new timetable, rolling stock, and crew schedule in short time in case of track blockages. Furthermore, few trips tend to get cancelled in the rolling stock rescheduling and the crew rescheduling modules.

Our computational experiments show that the framework does not need many iterations between the different modules. The General Framework and the Different Costs Variant solves 40 (47)% of the full (partial) blockage instances after one iteration and already 99 (99)% after two iterations. This indicates that the chosen models perform well on an individual basis and do not come up with solutions which make it hard to reschedule the other resources.

The Crew First Variant performs even better by solving 88 (92)% of the full (partial) blockage instances after one iteration. However, this variant might not always be applicable in practice, because there can be crew specific rules depending on the rolling stock circulation.

Our framework is able to find a new feasible timetable, rolling stock circulation, and crew schedule after the occurrence of a disruption. The potential of this framework is significant. Railway operators can use it for real-time railway disruption management in practice. It provides a feasible schedule based on infrastructure, rolling stock, and crew constraints. The fact that this framework uses individual rescheduling modules for each resource allows railway operators to use their own preferred approaches for the rescheduling modules.

The interchangeability of the rescheduling modules makes the framework also useful for researchers. They can test whether their suggested approaches for rescheduling one resource also performs well on a global scale where the other resources are considered as well.

We see several interesting directions for future research. Firstly, it would be interesting to compare our results to those obtained with a microscopic approach as a timetabling module. By using a microscopic representation of the infrastructure, the running and headway times can be computed more accurately. Our framework allows to replace the macroscopic timetabling module by a microscopic one, or by a model that incorporates both microscopic and macroscopic aspects. Secondly, it would be interesting to improve the feedback loop from the rolling stock and crew rescheduling modules to the timetabling module. By adjusting the solution of the timetabling module instead of cancelling the trips that cannot be covered by rolling stock or crew, solutions might be obtained that are overall of better quality. Finally, our further research will also focus on modelling passenger behaviour in a more realistic way

than in the current paper.

References

Abbink, E.J.W., D.G.A. Mobach, P.J. Fioole, L.G. Kroon, E.H.T van der Heijden, N.J.E. Wijngaards. 2009. Actor-agent application for train driver rescheduling. *Proceedings of the Eighth International Conference on Autonomous Agents and Multiagent Systems*. 513–520.

Cacchiani, V., D. Huisman, M. Kidd, L. Kroon, P. Toth, L. Veelenturf, J. Wagenaar. 2014. An overview of recovery models and algorithms for real-time railway rescheduling. *Transportation Research Part B: Methodological* **63** 15 – 37.

Cadarso, L., Á. Marín, G. Maróti. 2013. Recovery of disruptions in rapid transit networks. *Transportation Research Part E: Logistics and Transportation Review* **53** 15–33.

Cadarso, L., G. Maróti, Á. Marín. 2015. Smooth and controlled recovery planning of disruptions in rapid transit networks. *Intelligent Transportation Systems, IEEE Transactions on* **16**(4) 2192–2202.

Corman, F., A. D'Ariano, I.A. Hansen, D. Pacciarelli, M. Pranzo. 2011. Dispatching trains during seriously disrupted traffic situations. *Networking, Sensing and Control (ICNSC), 2011 IEEE International Conference on*. 323–328.

D'Ariano, A., D. Pacciarelli, M. Pranzo. 2007. A branch and bound algorithm for scheduling trains on a railway network. *European Journal of Operational Research* **183**(2) 643–657.

Fioole, P-J., L.G. Kroon, G. Maróti, A. Schrijver. 2006. A rolling stock circulation model for combining and splitting of passenger trains. *European Journal of Operational Research* **174**(2) 1281–1297.

Haahr, J.T., R.M. Lusby, J. Larsen, D. Pisinger. 2014. A branch-and-price framework for railway rolling stock rescheduling during disruptions. Tech. rep., DTU Management engineering. URL `http://findit.dtu.dk/en/catalog/2203007514`.

Haahr, J.T., J.C. Wagenaar, L.P. Veelenturf, L.G. Kroon. 2016. A comparison of two exact methods for passenger railway rolling stock (re)scheduling. *Transportation Research Part E: Logistics and Transportation Review* **91** 15 – 32.

Lamorgese, L., C. Mannino. 2015. An exact decomposition approach for the real-time train dispatching problem. *Operations Research* **63**(1) 48–64.

Louwerse, I., D. Huisman. 2014. Adjusting a railway timetable in case of partial or complete blockades. *European Journal of Operational Research* **235**(3) 583–593.

Nielsen, L.K., L.G. Kroon, G. Maróti. 2012. A rolling horizon approach for disruption management of railway rolling stock. *European Journal of Operational Research* **220**(2) 496–509.

Potthoff, D., D. Huisman, G. Desaulniers. 2010. Column generation with dynamic duty selection for railway crew rescheduling. *Transportation Science* **44**(4) 493–505.

Rezanova, N.J., D.M. Ryan. 2009. The train driver recovery problem- a set partitioning based model and solution method. *Computers & Operations Research* **37**(5) 845–856.

van der Hurk, E. 2015. Passengers, Information and Disruptions. Ph.D. thesis, Erasmus University Rotterdam.

Veelenturf, L.P., M.P. Kidd, V. Cacchiani, L.G. Kroon, P. Toth. 2016a. A railway timetable rescheduling approach for handling large-scale disruptions. *Transportation Science* **50**(3) 841–862.

Veelenturf, L.P., D. Potthoff, D. Huisman, L.G. Kroon. 2012. Railway Crew Rescheduling with Retiming. *Transportation Research Part C: Emerging Technologies* **20**(1) 95 – 110.

Veelenturf, L.P., D. Potthoff, D. Huisman, L.G. Kroon, G. Maróti, A.P.M. Wagelmans. 2016b. A quasi-robust optimization approach for crew rescheduling. *Transportation Science* **50**(1) 204–215.

Walker, C.G., J.N. Snowdon, D.M. Ryan. 2005. Simultaneous disruption recovery of a train timetable and crew roster in real time. *Computers & Operations Research* **32** 2077–2094.

Zhan, S., L.G. Kroon, L.P. Veelenturf, J.C. Wagenaar. 2015. Real-time high-speed train rescheduling in case of a complete blockage. *Transportation Research Part B: Methodological* **78** 182 – 201.