An Overview of Recovery Models and Algorithms for Real-time Railway Rescheduling [1]

Valentina Cacchiani[1], Dennis Huisman[2], Martin Kidd[1]
Leo Kroon[2], Paolo Toth[1], Lucas Veelenturf[2], Joris Wagenaar[2]

[1]*DEI, University of Bologna, Viale Risorgimento 2, I-40136 Bologna, Italy*
e-mail:{valentina.cacchiani,martin.kidd,paolo.toth}@unibo.it
[2]*Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands*
e-mail:huisman@ese.eur.nl, {lkroon, lveelenturf, jwagenaar}@rsm.nl

Econometric Institute Report EI2013-29

Abstract

This paper presents an overview of recovery models and algorithms for real-time railway disturbance and disruption management. This area is currently an active research area in Operations Research, including real-time timetable rescheduling and real-time rescheduling of the rolling stock and crew duties. These topics are addressed in this paper. Also research dealing with the integration of more than one rescheduling phase is discussed. Currently, the developed methods have been tested mainly in an experimental setting, thereby showing promising results, both in terms of their solution quality and in terms of their computation times. The application of these models and algorithms in real-life railway systems will be instrumental for increasing the quality of the provided railway services, leading to an increased utilization of the involved railway systems.

1 Introduction

This paper gives an overview of the research that has been carried out recently in the area of recovery models and algorithms for real-time railway disturbance and disruption management. Here *disturbances* are relatively small perturbations of the railway system that can be handled by modifying the timetable, but without modifying the duties for rolling stock and crew. *Disruptions* are relatively large incidents, requiring both the timetable and the duties for rolling stock and crew to be modified. Disturbance and disruption management are currently active research areas in Operations Research. This research has lead to many papers on algorithmic tools for real-time rescheduling of the timetable, the rolling stock and the crews.

Research in this area is scientifically challenging, but also promising from a practical point of view. Indeed, in many countries increasing the market share of public transport, and especially railway transport, is considered as one of the solutions for mobility problems. Moreover, railway transport is seen as a *green* mode of transportation. Thus increasing the market share of railway transport is one of the top priorities of many governments.

However, many potential railway passengers consider the perceived service level and unreliability of railway systems (caused by e.g. train delays, insufficient seating capacity, broken passenger connections, cancelled trains) as a disqualifier to use the train. Thus improved service and reliability will be needed to seduce these potential passengers to travel by train.

[1] The research leading to this paper has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) in the ON-TIME project under Grant Agreement SCP1-GA-2011-285243.

The application of recovery models and algorithms for real-time disturbance and disruption management, implemented in user-friendly decision support systems, is considered as a key element for improving the service and reliability of railway systems. It will lead to improved capacity and delay management, and to less train cancellations.

The basis of a railway system is formed by an extensive planning process, resulting in a timetable and resource duties, often already long time ahead of the real-time operations. Here the resources are the rolling stock and the crews. Note that the infrastructure might be considered as a railway resource as well (in most cases it is actually the most expensive one), and the timetable as the corresponding resource schedule. However, in this paper we will not use this terminology. The timetable and the resource duties usually represent an optimized trade-off between service, efficiency, and robustness of the railway system.

Unfortunately, the real-time operations of a railway system are more or less unavoidably subject to unexpected disturbances and disruptions, which result in infeasibilities in the timetable and possibly in the resource duties. The passengers experience these disturbances and disruptions as delays of trains, broken connections, insufficient seating capacity, and cancelled trains. Also freight trains suffer from disturbances and disruptions, mainly in the form of delays and rerouted trains. Thus there is a need to recover from a disturbed or disrupted situation, and to restore the normal railway services as quickly as possible.

Recovering from a disturbed or disrupted situation to a feasible situation requires to change the timetable, and, if necessary, the rolling stock and crew duties as well. Currently, the involved dispatchers in practice take these decisions mainly manually, based on their experience and craftsmanship, without *intelligent* decision support.

However, recently a large number of mathematical recovery models and algorithms have been developed to support the dispatchers in their decision making processes. This paper presents an overview of this research. Since this is an active research area, we do not pretend to give a complete overview, but we mention at least the most relevant recent papers.

In this overview paper we restrict ourselves to mathematical models and algorithms that have been developed for algorithmically solving railway rescheduling problems. We do not cover organizational nor communication aspects that also play an important role in railway rescheduling. For a brief introduction to these subjects we refer to Jespersen-Groth et al. (2009). We also restrict ourselves to the rescheduling problems related to the timetable, the rolling stock and the crews. We do not focus on the logistic problems related to the cause of the disturbance or the disruption, such as repairing broken railway infrastructure.

This paper is structured as follows. Section 2 presents some background information on railway systems. Thereafter, Sections 3, 4, and 5 deal with models and algorithms for timetable rescheduling, rolling stock rescheduling, and crew rescheduling, respectively. Next, Section 6 covers models and algorithms that integrate different rescheduling phases. Various approaches to solve these problems are discussed in terms of the type and scale of the disruptions dealt with, the network infrastructure and topology, the objective function and constraints considered, and the utilized optimization methods. This paper is finished with conclusions and a perspective on further research in Section 7.

2 Background information

2.1 Timetable, Rolling Stock, and Crew

A railway system is usually based on a timetable, which specifies the trips that are to be carried out by the trains in the system. Here a trip is a movement from a departure station to an arrival station in a specified time slot. The timetable also specifies the detailed routes of the trains through the railway stations. The time slots in the timetable are usually based on certain minimum process times (running times, dwell times, headway times, etc.) plus certain buffer times or time supplements.

Often the timetable is generated in a number of planning cycles: for passenger trains the timetabling process starts a *long time ahead* of the real-time operations, whereas freight trains are scheduled more *just-in-time*. For passenger trains, the timetable is usually based on a line system: here a line is a set of trains with the same origin station and the same destination station and with the same set of underway stations where the trains dwell.

There are two main timetable variants. One variant is the periodic (or cyclic) timetable that is repeated every given time period, for example every hour, with only slight differences between peak hours and off-peak hours. The other variant is the non-periodic timetable, that allows to follow the passenger demand with the frequencies of the trains. In both cases the timetable is usually repeated every day, although there may be differences between weekdays and weekend days. We refer to Cacchiani and Toth (2012), Caprara et al. (2007, 2011), and Lusby et al. (2011) for surveys of the literature on timetable planning.

The timetable is carried out by rolling stock that is staffed by appropriate crew. For a passenger train the rolling stock consists of a number of carriages hauled by a locomotive or by a number of coupled train units. For a freight train the rolling stock consists of one or more locomotives and a number of freight carriages. The crew of a train consists of a train driver and for a passenger train also of one or more conductors, depending on the number of carriages allocated to the train.

The rolling stock and crew form the resources operating in a railway system. They operate according to their preplanned resource duties or schedules. Here a duty is a set of tasks to be carried out by a single resource unit on a single day. A task is connected with a trip in the timetable. Carrying out a task means that the corresponding resource unit moves from the trip's departure station to its arrival station during the trip's time slot. A task can be a *real* task, but also a *repositioning* task for bringing the involved resource unit to the start location of its next task, or to bring it back to its home depot.

In many countries, a railway infrastructure manager is responsible for the capacity of the railway infrastructure and for its allocation to the train operating companies. The train operating companies themselves are responsible for the capacities of their resources and for operating their trains on the railway infrastructure. As a consequence, the timetable is usually created under the responsibility of the infrastructure manager, and in cooperation with the train operating companies. The resource schedules for rolling stock and crews are generated by the train operating companies.

2.2 Disturbances and Disruptions

In principle the timetable and the resource duties are conflict free. If there are no disturbances nor disruptions, then the timetable and the resource duties can be executed exactly as planned. However, in the real-time operations *disturbances* and *disruptions* are inevitable.

In this paper we use the term disturbances for "relatively small" perturbations influencing the railway system, and the term disruptions for "relatively large" external incidents leading to the cancellation of a number of trips in the timetable.

A disturbance corresponds to the fact that certain railway processes (e.g. driving from one station to another, or dwelling in a station) last longer than specified in the timetable. As a consequence, trains may depart and/or arrive later than planned. A delay caused by the fact that such a process lasts longer than planned is called a *primary* delay. Such a delay may easily knock-on from one train to another. In that case the delay of the second train is called a *secondary* delay. By definition, a disturbance is such that it can be handled by rescheduling the timetable only, without rescheduling the resource duties.

A disruption is a relatively large external incident strongly influencing the timetable, and requiring the resource duties to be rescheduled as well. A disruption is usually related to large delays or cancellations of a number of trains. These may be caused by a temporary blockage of the railway infrastructure, for example by malfunctioning infrastructure or rolling stock, or by an accident. Due to the blockage, a number of trains may incur large delays, or a number of trips in the timetable must be cancelled. As a consequence, the corresponding resources do not arrive in time at the station where the next tasks in their duties must be carried out, or they do not arrive there at all. Thus the trips corresponding to these tasks must be cancelled as well, or other resource units must be allocated to them. This means that the resource duties must be rescheduled. Usually, the duration of a disruption is not known at its start. Therefore, the timetable and the resource duties may have to be rescheduled several times, whenever new information about the duration of the disruption becomes available.

2.3 Rescheduling

In case a disturbance or a disruption occurs, the railway system must be rescheduled. As was indicated above, in the case of a disturbance, only the timetable must be rescheduled. This means that trains are retimed (modified times for entering the block sections are determined) or rerouted (a new route is selected from a set of feasible routes inside or between stations), that the order of a pair of consecutive trains on a track is changed, or that passenger connections are maintained or dropped.

In case of a disruption, usually the timetable is rescheduled first. This usually means that several trains are cancelled. Furthermore, in order to make the timetable feasible again, also the measures for dealing with disturbances are applied, such as retiming or rerouting of trains. The rescheduled timetable is then given as input to the rolling stock rescheduling phase. Finally, the rescheduled timetable and rolling stock duties are used as input for the crew rescheduling phase. The modified timetable, rolling stock schedule and crew schedule are then communicated to the involved parties, such as the crews. Also the passengers are informed about the modified timetable.

As was mentioned in Section 2.1, the railway infrastructure manager is usually responsible for the allocation of railway infrastructure capacity for operating trains, and the train operating companies are responsible for the rolling stock and crew schedules. These responsibilities are usually maintained in the real-time rescheduling stage. Thus the timetable is rescheduled under the responsibility of the railway infrastructure manager, and in cooperation with the train operating companies. The train operating companies are responsible for rescheduling the rolling stock and the crews.

An important difference between the planning stage and the rescheduling stage during disruptions is that in the latter case less time is available for rescheduling. In principle, solutions are expected within minutes. For the resources, another important difference is that in general there is less flexibility in the rescheduling stage, since many resource duties have already started at the time of the disruption when the rescheduling is carried out, and cannot be easily diverted. In addition, the solution space is bounded by the remaining time until the end of the rescheduling horizon, which is usually the end of the day. Hence, if the disruption happens in the evening, then the solution space is much smaller than in case the disruption happens in the morning. Moreover, some railway connections in the network may be unavailable due to the disruption, which prohibits transferring resources to some locations where they are needed.

Note that in the real-time operations, the resource units are individual units with their own capabilities and restrictions. For example, each individual train driver has certain licenses for operating trains with certain types of rolling stock and on certain parts of the railway network. Similarly, there may be rolling stock units which have certain defects, which limit their applicability. For example, a train unit may be *en route* to a maintenance facility, or it may not be able to operate on its own without being coupled with another unit. The latter is in contrast with the planning stage, which is often dealing with anonymous and more or less interchangeable resource units.

Various objective functions are considered in rescheduling a railway system. In the real-time operations it is usually most important to keep up as much as possible service for the customers, notwithstanding the occurrence of disturbances or disruptions. Usually, efficiency is not the most important objective: most costs have been determined in the planning stage, and can be influenced only marginally in the real-time operations. Nevertheless, efficiency is not an issue that can be neglected completely in the real-time operations.

In case a disturbance occurs, an objective may be to minimize the delays of the trains or the delays of passengers and freight: delays are an important performance indicator for passengers and freight. In case of a disruption, a general objective may be to recover the system to the normal situation as soon as possible. If certain trips have to be cancelled due to the disruption, then another objective may be to reschedule the timetable and the resources in such a way that as few as possible *additional* trips have to be cancelled due to lacking resources. On the trips that are still operated, the capacities of the trains should preferably be such that also unexpected rerouted passenger flows can be handled. Furthermore, several other operational objectives may play a role. These will be discussed in the following sections.

3 Train Timetable Rescheduling

This section deals with Train Timetable Rescheduling (TTR) in case of disturbances or disruptions, and considering a microscopic or a macroscopic view on the railway system. TTR (also called *train dispatching* or *conflict detection and resolution*) consists of adjusting in real-time an existing timetable that has become infeasible due to unpredicted disturbances or disruptions. The aim of TTR is to quickly re-obtain a feasible timetable of sufficient quality.

In practice, contingency plans may be used in order to deal with rescheduling the timetable during disruptions. Contingency plans are used e.g. in the Netherlands. Each contingency plan is a pre-planned periodic timetable that can still be operated, given the nature of the disruption and the implied restrictions for the infrastructure. In the Netherlands there are over 1000 different contingency plans. The selection of the contingency plan to be used is often done in a non-automated way, based on the experience and craftsmanship of the involved dispatchers. Nevertheless, the set of contingency plans may not cover all potential disruptions, in particular if multiple disruptions occur at the same time. In addition, each time the timetable changes structurally, the contingency plans have to be updated as well. Therefore the application of mathematical recovery models for solving TTR in real-time will be a much more efficient and effective way of dealing with disturbances and disruptions.

In recent years many researchers tackled TTR, and many different mathematical formulations have been proposed. They are characterized by different types of variables and objective functions, and different sets of constraints, depending on their specific application. The common decisions that need to be taken into account consist of:

- the routing of trains,
- the time instants of the departures and arrivals of trains,
- the order of trains on their common track sections.

These decisions can be expressed by using different types of variables and different constraints. Sometimes the routing of the trains is already given on input. Additional decisions may arise depending on the focus of the model. For example, in case of focusing on the effect of delays on passengers, one needs to consider which passenger connections should be maintained and which can be cancelled. Obviously, the latter decisions are implicitly described already by the departure and arrival times of the trains.

The main constraints that have to be satisfied by the timetable, both in the planning stage and in the real-time operations, are the so-called track capacity constraints. These have to guarantee a sufficient distance in time and space between any pair of trains, both inside and between stations, resulting in the well-known so-called *Blocking Time Stairway*, describing the time reservations of the successive block sections by a train.

As a consequence, as shown in Mascis et al. (2001, 2007), TTR in its most general form has a structure that is similar to that of the blocking (or no-store), no-wait job shop scheduling problem. The feasibility version of the problem is an NP-complete combinatorial problem, as proved in Mascis and Pacciarelli (2002).

3.1 Timetable Rescheduling: Mathematical formulation

The different model variants and techniques for solving TTR will be briefly presented in Section 3.2. We refer to the corresponding papers for a more detailed description.

As a representative example, we present in this section a mathematical formulation for TTR as proposed by D'Ariano et al. (2007a). This version of TTR consists of computing a conflict-free train timetable compatible with the actual status of the railway network and such that the global *secondary delay* is minimized. This delay corresponds to the global difference (by taking into account all the considered trains) between the accumulated delay and the *primary delay*, which is the external delay that cannot be recovered by TTR.

TTR takes as input the railway network and its actual status. The network is composed of track segments and signals used to control the traffic in the network. A track segment between two signals is called a block section. Each block section can host at most one train at a time. The actual status of the network describes the delays of the trains and their positions.

TTR can be partitioned into two subproblems. A *retiming problem* is solved in which the time instants at which each train enters each block section of a route is determined, as well as a *rerouting problem*, in which for each train a route is selected from a set of feasible routes inside the stations, and possibly also between the stations.

We start describing a mathematical model for the retiming problem, which is based on job shop scheduling with additional constraints. In particular, jobs correspond to trains and machines to block sections. The travelling of a train through a block section corresponds to an *operation*. The *processing time* of an operation represents the running time of the associated train on the corresponding block section. Since each block section can host at most one train at a time, a conflict occurs when two (or more) trains require the same block section at the same time. In addition, a *setup time* must be taken into account between the entrance of a train in a block section and the entrance of a successive train in the same block section: the block section remains blocked by the first train for a certain amount of time.

In order to determine a conflict-free schedule, the problem can be modeled by using an *Alternative Graph model* introduced by Mascis and Pacciarelli (2002) for no-store job shop scheduling. In the latter, no-store constraints express that a job cannot leave a machine until the subsequent machine becomes available. This situation happens in train scheduling as well. The alternative graph $G = (V, F \cup A)$ is defined as follows. Each node corresponds to an operation. Two consecutive nodes in sequence (corresponding to the passing of a train through a block section and then through the successive block section in the fixed route for the train) are represented by a fixed arc belonging to the set F. Two trains requiring the same block section at the same time can cause a potential conflict. Therefore a processing order and sufficient headway for the corresponding conflicting operations must be determined. This is modeled by pairs of alternative arcs belonging to the set A.

The timing problem, seen as a job shop scheduling problem with additional constraints, calls for determining starting times for the operations, such that processing times are respected and, for each pair of operations corresponding to the same block section, setup time constraints are satisfied. The goal is to minimize an appropriate objective based on the delays of the operations, such as the average or the maximum delay.

Let $t_0, \ldots, t_n$ be variables representing the starting times of operations $0, \ldots, n$, where operations 0 and n correspond to dummy operations. Let $\sigma(i)$ be the successor operation of operation i in the fixed sequence, and f_i the processing time of operation i ($i = 0, \ldots, n$). Let f_{ij} be the setup time between conflicting operations i and j. Now the timing problem can be formulated as the following disjunctive program:

$$\begin{aligned} \min \quad & t_n - t_0 & & (1) \\ \text{subject to} \quad & t_{\sigma(i)} \geq t_i + f_i & (i, \sigma(i)) \in F \quad & (2) \\ & (t_k \geq t_h + f_{hk}) \vee (t_j \geq t_i + f_{ij}) & ((i,j),(h,k)) \in A \quad & (3) \end{aligned}$$

The objective function (1) calls for minimizing the makespan. Constraints (2) impose the processing time constraints for operations corresponding to travelling of a train through consecutive block sections. Constraints (3) require that, for each pair of conflicting operations, i.e. for each alternative arc, a processing order is determined, therefore avoiding conflicts. In particular, operations j and k are conflicting operations, while $h = \sigma(j)$ and $i = \sigma(k)$ are the corresponding successor operations. Constraints (3) can be linearized by introducing binary variables x_{jk} for each pair of conflicting operations j and k. The binary variable x_{jk} assumes the value 1 if operation j is scheduled before operation k, and 0 if operation k is scheduled before operation j. Using a large positive number M, the linearized constraints read as follows:

$$t_k \geq t_h + f_{hk} - M(1 - x_{jk}) \qquad ((i,j),(h,k)) \in A \tag{4}$$

$$t_j \geq t_i + f_{ij} - Mx_{jk} \qquad ((i,j),(h,k)) \in A \tag{5}$$

The rerouting subproblem in TTR corresponds to determining the routes for the trains. Note that changing a route implies changing the sets F and A in the alternative graph G. In particular, if a train can be assigned to different routes, then the set F is still to be determined, and the set A depends on the selection of the set F. The complete TTR consists of determining the set F and an associated feasible timing of the trains. We refer to D'Ariano (2008) and Corman et al. (2010c) for further details on the rerouting model.

3.2 Timetable Rescheduling: Literature

Different approaches have been developed in the literature to tackle TTR. In this section, we describe several of these methods. In this overview, we restrict ourselves to the most recent methods that have been developed, since the older methods are hardly comparable, due to recent developments in hardware and software.

In this paper we make a number of important distinctions for TTR.

- The first one is the distinction between disturbances and disruptions, that was described already in Section 2.2.
- The second distinction concerns the level of detail considered in the railway system, in particular in the timetable. Two approaches can be distinguished, known as *microscopic* and *macroscopic*. The latter considers the railway network at a higher level, in which stations can be represented by nodes of a graph and tracks by arcs, and the details of block sections and signals are not taken into account. In a microscopic approach these aspects are considered in detail. In a microscopic approach, usually blocking time graphs and the underlying data are used to compute detailed running and headway times. The blocking time graph of a train represents the time instants at which a train needs to run safely without hinder at design speed over a sequence of track sections.
- The third distinction involves a focus on the trains or on the customers. Most papers focus on minimizing the delays of trains or the number of cancelled trains. However, there is also a growing number of papers focusing on minimizing the negative effects of disturbances and disruptions for passengers or freight.

In Table 1 we present a schematic overview of the papers that are discussed in the next sections. The table shows that most of the approaches in the literature deal with disturbances affecting the railway system rather than disruptions. Moreover, most papers consider the railway system at a microscopic level rather than at a macroscopic level. This microscopic approach is very much in line with the traditional approach in many railway systems. The bold entries in the table indicate that the corresponding papers explicitly focus on minimizing the delays of passengers or freight.

3.2.1 Disturbances - Microscopic Approach

In this section we discuss TTR when dealing with disturbances where the infrastructure is considered at a microscopic level. Much research related to TTR at a microscopic level

	Disturbances	Disruptions
Microscopic	Albrecht et al. (2011); Boccia et al. (2013); Caimi et al. (2012); Corman et al. (2009, 2010a,b,c, 2011b, 2012); D'Ariano et al. (2007a,b); D'Ariano (2008); D'Ariano et al. (2008a,b); D'Ariano (2008); Gély et al. (2006); Khosravi et al. (2012); Lamorgese and Mannino (2012); Lamorgese and Mannino (2013); Lusby et al. (2013); Lüthi et al. (2007); Mannino (2011); Mannino and Mascis (2009); Meng and Zhou (2011); Pellegrini et al. (2012); Rodriguez (2007); Schaafsma and Bartholomeus (2007)	Hirai et al. (2009); Corman et al. (2011a); Wiklund (2007)
Macroscopic	Acuna-Agost et al. (2011a,b); Chiu et al. (2002); Dollevoet et al. (2011, 2012); Dollevoet (2013); Kanai et al. (2011); Kumazawa et al. (2010), Min et al. (2011); Schachtebeck and Schöbel (2010); Schöbel (2007); Schöbel (2009); Törnquist (2012); Törnquist and Persson (2007)	Albrecht et al. (2013); Louwerse and Huisman (2012); Nakamura et al. (2011); Shimizu et al. (2008)

Table 1: Classification of the recent literature on Train Timetable Rescheduling.

is based on the Alternative Graph model introduced in Mascis and Pacciarelli (2002), and described in Section 3.1.

D'Ariano et al. (2007a) propose a branch-and-bound algorithm for scheduling trains in real-time. The problem is modeled as presented in Section 3.1. The model may include hundreds of machines (block sections) and jobs (trains) for real-life instances, and is therefore very hard to be solved in real-time. To overcome this issue, the authors propose to use *dynamic and static implication rules*, that are used to speed up the branch-and-bound approach, by taking into account relations between pairs of alternative arcs. The static rules are computed off-line on the basis of the network topology. An initial partial solution (possibly empty) is computed and a list of partial solutions is maintained (at the beginning it contains only the initial solution) during the algorithm. At each step, the algorithm chooses a partial solution from the list and an unselected pair of alternative arcs $((i, j), (h, k))$, and builds two partial solutions, one containing arc (i, j) and the other one containing arc (h, k), taking into account the static implication rules. Dynamic implications are then computed. Jackson's preemptive schedule is used, whose value gives a lower bound on any extension of the two partial solutions, see Jackson (1955). If the lower bound is smaller than the value of the best solution found, the two partial solutions are added to the list. Three heuristic algorithms which emulate the commonly adopted methods used in railway practice are used to compute the initial upper bound. Computational experiments are executed on a large sample of real-world size instances in the area around Schiphol Amsterdam Airport in the Netherlands. The timetable contains 54 trains circulating each hour, and different delay scenarios are evaluated. The branch-and-bound algorithm is compared with the three heuristics. The computation time limit is set to 120 seconds on a laptop with a 1.6 GHz Pentium M processor. The branch-and-bound algorithm is able to obtain the optimal solution for 297 out of 300 instances. Moreover, it is able to reduce the maximum secondary delay by approximately 100 seconds with respect to the three heuristics.

D'Ariano et al. (2008a) describe the implementation of the Alternative Graph model in the real-time railway traffic management system ROMA (Railway traffic Optimization by Means of Alternative graphs). This tool aims to support dispatchers in order to resolve conflicts when dealing with delays and disturbances.

Other papers focusing on the Alternative Graph model are the following: Corman et al. (2009, 2010a,b,c, 2011b, 2012), D'Ariano et al. (2007a,b); D'Ariano (2008); D'Ariano et al. (2008b), and D'Ariano and Pranzo (2009). All experiments described in these papers were carried out with ROMA. They concerned the line between Utrecht and 's Hertogenbosch, and the congested areas around Utrecht Central Station and around Schiphol Amsterdam Airport in the Netherlands. Different delay scenarios were considered in these papers.

D'Ariano et al. (2008b) relax some timetabling constraints in the ROMA model in order to obtain more real-time flexibility. Flexibility in arrival and departure times enforces to have two timetables: the operational timetable which takes into account the time windows (i.e. minimum and maximum arrival and departure times), and the public timetable presented to the passengers, which only includes minimum departure times and maximum arrival times. The objective is to minimize the delays of trains. Experimental results are performed on the congested area around Schiphol Amsterdam Airport with randomly generated disturbances. In particular, the authors start from a "rigid" timetable provided by Netherlands Railways. This timetable presents only a limited amount of buffer time to recover from delays. The authors construct a flexible timetable by replacing the scheduled arrival and departure times with maximum arrival and minimum departure times. Different types of disturbances are considered. Rescheduling is applied to the rigid timetable and to the flexible timetable for comparison. The results show that flexible timetables are promising. The branch-and-bound algorithm of ROMA outperforms its heuristics, attaining a deviation of not more than 26% from the lower bound, while for the three heuristics the deviation from the lower bound is over 85%. With an increase in flexibility from 0 to 120 seconds, the branch-and-bound algorithm exhibits a decrease in delays of more than 30%. The results were obtained by coding the model in C++ on a laptop with 1.6 GHz. The best solution is always found within 30 seconds of computation time.

The concept of flexible timetables is related to dynamic traffic management. Here it is important to allow a high degree of freedom in the real-time operations by relaxing some timetabling constraints, as described by Schaafsma and Bartholomeus (2007). In the planning stage only a partial order of the trains is determined, and arrival and departure times are provided only as time windows. The assignment of trains to platforms is postponed to the real-time rescheduling stage. Dynamic traffic management is applied in practice in the congested railway network around Schiphol Amsterdam Airport in the Netherlands. It is considered as an adequate measure for handling a large number of trains in a bottleneck area.

Mannino and Mascis (2009) study an application provided by Azienda Trasporti Milanesi (ATM), the major municipal public transport company of Milan in Italy. They develop a branch-and-bound algorithm to obtain optimal solutions. As described in Section 3.1, TTR can be decomposed by first selecting a routing for the trains and then calculating a timing. The proposed solution method consists of enumerating all the feasible routings for the trains and then solving, for each routing, the corresponding job-shop scheduling model. The model is formulated as in Section 3.1, but has a different objective function: it minimizes the deviations of the actual schedule from the original plan and the costs due to violating regularity (i.e. varying headways between consecutive departures). The authors report computational results on eight realistic instances provided by ATM, which contain between four and eight trains that are to be rescheduled. All instances were solved to optimality within a few seconds on a Pentium 4 processor with 3.2 GHz. The results are compared to those obtained by human dispatchers, and show that the proposed method is able to improve both the punctuality and the regularity indicators with up to 8%.

In Mannino (2011), Lamorgese and Mannino (2012) and Lamorgese and Mannino (2013), TTR is decomposed into smaller subproblems, associated with lines and stations. In particular, the approach is based on decomposing the problem into the two following subproblems: the *line traffic control problem* and the *station traffic control problem*. A Mixed Integer Programming (MIP) model is proposed for solving TTR on single-track railways. The model uses binary variables indicating whether or not a pair of trains meet in a station and continuous variables representing the arrival and departure times of trains at stations. The constraints impose that every crossing pair of trains (i.e. trains that can possibly meet on the line) meet in a station, and that the station capacity is respected. Two different formulations are proposed for the latter constraints: a non-compact formulation and a compact flow based one. Computational results, conducted by Lamorgese and Mannino (2012), show that the non-compact formulation provides better results on real-life instances of Italian single-track railways. The average computation time is less than 8 seconds for the compact formulation and about 2 seconds for the non-compact one, on a server with an Intel Core processor with 1.87 GHz.

In Boccia et al. (2013), a new MIP model is presented. A set of routes for each train is considered and tracks are subdivided into sections. The model uses binary variables indicating whether a route in the set is assigned to a train, and continuous variables representing the time at which a train reaches a block section. Two heuristic algorithms are proposed, both based on the MIP model. These are tested on a set of instances provided for the Railway Application Section (RAS) of INFORMS 2012, as well as on instances derived from the RAS ones. The results show that the proposed heuristic algorithms were able to find feasible solutions to all instances within the time limits (set to 3 or 15 minutes) imposed by the RAS competition, using an Intel Core i7 3.4GHz computer.

Lamorgese and Mannino (2013) introduce a new modeling approach which allows to reduce the computation times. The new variables are complementary to the original ones, i.e. they model that a pair of crossing trains *will not* meet at a given track. In particular, the binary variables model whether a train i meets a train j before a given track or after it. The approach used to solve the model is based on delayed row generation, similarly to Benders' decomposition. Computational results on real-life single-track instances are compared to those obtained by the model presented in Lamorgese and Mannino (2012), showing the effectiveness of the new formulation in reducing the computation times. The considered instances have up to 29 trains in a time horizon of nine hours. With the new formulation, the computation times are always below one second (except for one instance which requires about 13 seconds to be solved) on an Intel Core i7-2640M processor with 2.80 GHz. The previous method required sometimes more than one minute on the same computer.

A lot of research concerns congested bottleneck areas, since these are the most critical ones in case of disturbances or disruptions, where delays easily propagate from one train to another. Albrecht et al. (2011) use speed and dwell time control in order to reach certain defined targets and to improve the throughput of trains on parts of the Dutch and German railway networks. Two types of targets are considered, called *target points* and *target windows*. Target points correspond to arrival and departure times from stations, which are important for maintaining passenger connections. Other target points consist of passing times, i.e. when overtaking or crossing of trains occurs. When no connections need to be respected (as often happens in minor stations), target windows can be considered. The planned time can be exceeded by a delay and an upper limit is imposed. The authors propose a two-level optimization approach: optimization of speed between consecutive target points, and optimization of running times between target points, if target windows exist between the target points. They conclude that their method is a useful tool to improve timetable adherence, to save energy, and to increase the throughput through bottleneck areas.

Caimi et al. (2012) propose a model predictive control framework for railway traffic management in bottleneck areas, which manages traffic by retiming and rerouting of trains as well as partial speed profile coordination. In particular, they propose a closed-loop discrete-time control system. A forecast module computes a forecast of the evolution of the system. The rescheduling model computes a new disposition timetable, which can be combined with the forecasted time effects in the system for the dispatcher. Once the dispatcher takes a decision, this is forwarded to the trains and the intended commercial offer is changed accordingly, thus closing the control loop. A case study of the railway area of Berne in Switzerland is considered, which presents 13 platforms and 1500 trains. In total 10 delay scenarios are considered. Realistic solutions and short computation times make the approach viable for practice. In particular, a computation time above 1 minute was never experienced in the rescheduling model using a 24 processor Intel Xeon X5650 with 2.67 GHz and 95 GB RAM.

Gély et al. (2006) present a mathematical model to describe the timetable rescheduling problem in case of delays of trains. The model presents continuous variables representing arrival or departure times of trains from each visited node in the network, and binary variables which are used to express re-ordering (i.e. whether a certain train passes before another one at a given node of the network), track choices (i.e. whether a train uses a certain track or not), and extra-stops (i.e. whether a certain train stops at a node at which it was not supposed to stop in the original timetable). The constraints impose to respect headway times and running times, and require that every train cannot leave a station before the planned time in the original schedule. The objective is to minimize the total accumulated delay of trains. The algorithms proposed to solve the model include evolutionary algorithms and hybrid techniques. The methods are applied to real-life instances of the French railway company SNCF involving the railway line between Tours and Bordeaux. These instances present 200 trains in the time window between 15:00 and 23:00. The results show the feasibility and the effectiveness of decision support systems for real-time railway traffic management.

Khosravi et al. (2012) consider a railway network where signals are used to divide the network into blocks. The problem of assigning trains to blocks as they pass through a network after a disruption is formulated as a job shop scheduling problem with additional constraints. The authors propose three different modified versions of the shifting bottleneck (SB) procedure, a well-known heuristic from the literature for solving job shop scheduling problems, see Adams et al. (1988). Tests are performed on real-life instances from the very dense and congested London Bridge area in the United Kingdom. Comparisons are made between the three algorithms and the First-Come-First-Serve (FCFS) solution, the latter of which is close to the dispatcher's behavior when real-time decisions have to be taken. Three different types of disturbances are considered: there are small (less than 15 minutes), medium (between 15 and 30 minutes) and large (more than 30 minutes) disturbances. The results show that in 26 out of 31 test cases the modified SB algorithms are able to find schedules with less propagated delay than that obtained by the FCFS rule, which often results in deadlocks. For some cases the computation times of the SB algorithms are as large as 26 minutes using C++ on a dual core PC with 3 GHz and 4 GB RAM, while for the FCFS rule the computation time is less than one minute. The computation times of 26 minutes are not realistic in terms of real-time decision making, but it is reported that there is still room for improving the efficiency of the algorithms.

Lusby et al. (2013) present a Set Packing inspired formulation for rescheduling a number of trains in a complex junction. A resource based constraint system is used, in which resources correspond to track sections. In this system, any train path can be represented as a column with a value 1 element in correspondence of a track section that the train claims (and a value 0 element if the train does not claim the corresponding track section). In addition, banned track sections are considered, i.e. track sections that are banned by two trains that are not conflicting. In this case, a value 1/2 is used in the column. The value 1/2 is chosen according to the specific case study, in which no more than two trains can simultaneously ban the same track section. Binary variables are used to specify the selected train paths in the solution. The constraints ensure that all trains receive a path, and that at most one train can claim any track section during each time interval. The objective is to minimize the total deviation from the scheduled arrival times, weighted according to the train priorities. The model is solved by a branch-and-price approach. A real-life instance provided by the major German railway company Deutsche Bahn is used to test the model and the algorithm. It presents 143 trains in a time horizon of 2 hours. Several delay scenarios are generated by delaying sets of trains between 5 and 15 minutes. The results indicate the efficiency of the proposed approach by confirming that, with a given time limit of 270 seconds, practical problems can be solved within 3.5 percent of optimality. The integer solution is usually not improved after 60 seconds have elapsed. The results were obtained by coding the approach in C++ on an Intel Core 2 Duo, 2.4 GHz computer with 2 GB RAM .

Lüthi et al. (2007) combine rescheduling of trains in real-time, i.e. updating an existing timetable in response to disturbances, and controlling the trains so that they follow a given trajectory with a predefined accuracy, in an integrated system. The proposed method is tested on a case study of the railway network around Lucerne in Switzerland. It reduced the total system delay significantly. It also showed the importance of controlling the trains precisely according to the timetable: even small delays may easily increase to larger ones.

Pellegrini et al. (2012) consider the timetable rescheduling problem in case the entrance of a given subset of trains into a control area has been delayed due to a disturbance in the network. A MIP formulation is presented for the problem, modeling the infrastructure in terms of track-circuits (i.e. electrical circuits) to reliably detect the presence of trains on particular track sections. The track is split into a sequence of track circuits. The objective is to minimize any additional secondary delays in the control area resulting from given primary delays. For each train a number of different routes through the control area are defined. Continuous variables are used to represent the time at which a train enters a track circuit, and the delay assigned to a train in a track circuit. Binary variables are used to express whether a given train uses a given route, and whether a given train uses a track circuit before another given train. The model calls for assigning a route to each train, as well as a possible delay for the train on each track circuit. Various time constraints (such as running and clearing times) and capacity constraints are considered, as well as constraints related to shared rolling stock and connections between trains. The model is tested on real-life instances including the Lille-Flandres station in France. For the large majority of the instances considered, the optimal solution is found within 3 minutes of computation time using CPLEX version 12 in C++ and an Intel Xeon quad core 2.93 GHz processor with 6 GB RAM.

Rodriguez (2007) studies rerouting and reordering possibilities of trains for the railway network around the Pierrefitte-Gonesse junction north of Paris. A job shop scheduling model with additional resource constraints is proposed to detect conflicts between trains. The conflicts are solved with constraint programming techniques. Compared to the effects of decisions applied by the French operator SNCF, the constraint programming techniques are able to reduce the delays by 62-95% within a time limit of 180 seconds. The experiments are conducted with ILOG Solver 5.3 using a 2.66 GHz Pentium IV processor.

Meng and Zhou (2011) studied the robustness of a plan for a disrupted single-track rail line, against random variations both in the running times and in the duration of the disruption. A macroscopic stochastic programming model is used in a rolling horizon framework to identify robust schedules for every period. The quality of the solution is then evaluated to understand how it is affected by the accuracy of the information on the disruption duration. Numerical experiments on a real-world train schedule are conducted to show the importance of robust train dispatching in a stochastic environment. The experiments are solved on average within 800 seconds on a processor with 1.8 GHz and 1.5 GB RAM, depending on the traffic density.

3.2.2 Disturbances - Macroscopic Approach

In this section we discuss TTR when dealing with disturbances where the infrastructure is considered at a macroscopic level. We make a distinction between the papers that focus on the delays of trains, and the papers that focus on the delays of passengers or freight.

Train orientation Törnquist and Persson (2007) present a MIP model that takes into account reordering and rerouting of trains. The model contains continuous variables for representing start and end times of an event, and continuous variables that represent the delay of an event. In addition, it contains binary variables to express whether an event uses a track, and binary variables are used to decide the order of trains. The model considers fixed headway times between trains and fixed running times along segments between stations. The goal is to minimize a cost function based on the train delays. To solve the complete model may require a lot of computational effort. In addition, since the sequence of trains on the tracks as specified by the initial timetable will mostly remain the same, only a few modifications may be necessary to achieve a significant improvement. Therefore, four different strategies are evaluated, which are mainly based on maintaining the order of trains but allowing some swaps. The paper considers the problem of rescheduling railway traffic on the southern part of the Swedish railway network connecting Stockholm, Gothenburg, and Copenhagen (in Denmark). The instances contain more than 500 trains. In the experiments disturbances are generated during rush hours with a time horizon between 30 and 90 minutes.

Since the approach described in Törnquist and Persson (2007) sometimes has difficulties to find a good solution within a few seconds, Törnquist (2012) describes a heuristic greedy approach for the same problem. In order to quickly find a good solution the heuristic performs a depth-first search that branches according to a set of criteria. The heuristic provides solutions for the instances that could not be solved by the approach of Törnquist and Persson (2007). In general the heuristic provides solutions that are good enough very fast, but according to the author, it could be further improved.

Acuna-Agost et al. (2011a) also formulate the railway rescheduling problem as a MIP model, thereby extending the model presented in Törnquist and Persson (2007). There are two main extensions with respect to the latter model. The first is that the authors explicitly consider the fact that unplanned stops will change the minimal and maximal allowed travel time because of acceleration and braking; the second one is the modification of some constraints to admit more than one train in the same section running in the same direction. The problem is solved by limiting the search space around the original non-disrupted timetable. In particular, hard and soft fixing of integer variables with local-branching-type cuts are used, see Fischetti and Lodi (2003). The method is tested on instances based on railway networks in France and Chile. The network in France includes 67 trains in a time horizon of 7 hours. The scenarios consider one or two trains that are hindered by a disturbance, causing delays of 10, 20 or 30 minutes. The network in Chile includes 40 trains and a time horizon of 24 hours. The scenarios present up to 4 trains that are hindered by a disturbance, causing delays of 10, 20 or 30 minutes as well. Using CPLEX 11.1 and an Intel Core 2 processor with 1.66 GHz and 2 GB RAM, it is shown that solutions with an average optimality gap of less than 1% may be obtained in less than 5 minutes of computation time. The latter is viable in practice in a real-time rescheduling context.

In a follow-up paper, Acuna-Agost et al. (2011b) study the same problem as in Acuna-Agost et al. (2011a). Since the latter model may lead to long computation times for certain instances, the authors develop another solution approach called SAPI (Statistical Analysis of Propagation of Incidents). This method is based on the same MIP model, but it proceeds by estimating the probability that an event in the railway network is affected by a set of disturbances, and by reducing the search space accordingly. The method is tested on the same railway networks in France and Chile. The results show that SAPI is useful in practice and improves the performance of the earlier developed method, in particular with respect to the computation times. The method finds solutions with an average gap of 0.5% in 128 seconds for the network in France, and an average gap of 0.05% in 15 seconds for the network in Chile, using CPLEX 11.1 and an Intel Core 2 processor with 1.66 GHz and 2 GB RAM.

Min et al. (2011) propose a MIP formulation to solve the train-conflict resolution problem in the metropolitan railway network of Seoul in South-Korea. They showed that an instance with two single-tracked segments and three stations is already NP hard. Therefore they proposed a heuristic Fix and Regenerate Algorithm (FRA) that exploits the structure of the problem. Based on several observations of the Seoul metropolitan railway network, they formulate a number of assumptions and imply that their problem is separable under those assumptions. A column generation approach is then used. The model provides a timetable within 1 minute using an Intel Quad processor with 2.50 GHz and 4.0 GB RAM.

Passenger orientation Chiu et al. (2002) describe an interactive tool which is based on the application of constraint propagation. Two optimality criteria are considered: minimizing the number of modified station visits and minimizing the passenger delays. The tool takes into account speed limits, headway times between trains occupying the same track, headway times between arrivals and departures of trains at a station, overtaking, and minimum dwell times of trains in a station. The rescheduling consists of adjusting the departure and arrival times of the trains. Two heuristic algorithms are tested on instances involving the railway line between Nanjinxi and Shanghai in China. The experiments confirm the feasibility of the described approach, usually within seconds.

Schöbel (2007) studies the problem of delay management, which consists of deciding if connecting trains should wait in a station for delayed feeder trains or if they should depart on time. In other words, the problem is to decide which connections should be maintained and which can be dropped, with the goal of minimizing the total passenger delay. The author presents a path-oriented MIP model and an activity based MIP model. Both models are based on representing the delay management problem by means of an *activity-on-arc project network*. The latter contains nodes for all arrival and departure events, and a set of arcs, called activities, corresponding to waiting or driving of the trains, and to changing of the passengers from one train to another at a station. A timetable is obtained by assigning each event a time, such that the minimal duration for performing each activity is satisfied. The path-oriented model has a binary variable for each path, used for deciding whether all connections on the path are maintained or not. Two sets of constraints guarantee that the delay at the start of an activity is transferred to its end, where it can be reduced by the slack time of the activity (i.e. the time that can be saved by performing the activity as fast as possible). Additional constraints are used to satisfy timing requirements for the passenger changing activities that are maintained. The objective is to minimize the total passenger delay. This leads to a quadratic objective function that can be linearized. The activity based model is equivalent to the path-oriented model as they both lead to the same set of optimal solutions. The activity based model has a cubic objective function, and presents a larger number of variables and constraints, but it has the advantage that a branch-and-bound algorithm can be easily implemented with branching on arc variables. A simplification of the cubic model is also presented, which results in an integer linear program. It is shown that this linearization is correct under specific conditions.

Schöbel (2009) and Schachtebeck and Schöbel (2010) also include constraints on the limited capacity of the tracks. A branch-and-bound algorithm and several heuristic approaches are developed in order to solve these problems. These algorithms are tested and compared with each other on real-world data of the German railway company Deutsche Bahn involving parts of the German railway network, using an AMD Opteron 275 system with a 2.2 GHz processor and 12 GB RAM.

Dollevoet et al. (2012) extend the delay management problem with rerouting of passengers. That is, it is assumed that, given the status quo of the railway system, the passengers know which connections between trains will be maintained in the near future. Based on that, they may decide to take an alternative route to their destination. The underlying model is a large MIP model that is solved with CPLEX 11.1 on an Intel Xeon Quad Pc with 3 GHz. The model is based on the activity based MIP model presented in Schöbel (2007). Binary variables are used to decide whether a connection is maintained or not. The new aspect that is taken into account corresponds to the routes that the passengers take between a given origin-destination pair. To model the routing decisions, binary variables are used that express whether a connection is used by passengers in the path between a given origin-destination pair. Besides the constraints of the activity based model presented in Schöbel (2007), additional constraints are used to model the routing decisions. The objective is to minimize the total passenger delay. The model is able to solve 3 out of 4 presented cases within one minute, but is not able to solve the largest case.

The delay management model is further extended in Dollevoet et al. (2011) and in Dollevoet (2013), where also station capacities are taken into account. In order to deal with its large size, the resulting model is solved in an iterative way, using CPLEX 12.2 on an Intel Core i5-2410M with 4 GB RAM, within 3 minutes. The models are tested on parts of the Dutch railway network. The results of the experiments show that especially the possibility for passengers to choose an alternative route through the network leads to a significant reduction of their delays. The algorithms outperform the static waiting rules for maintaining connections that are used in Dutch practice.

Also Kumazawa et al. (2010) aim at minimizing the dissatisfaction experienced by the passengers due to disturbances. They propose a rescheduling algorithm that calculates a value for the amount of dissatisfaction experienced by passengers due to disturbances on the Japanese railway network. In addition to a conventional passenger flow analysis, the passenger overflow, defined as the waiting time experienced by a passenger while waiting on a platform, is considered. The approach consists of two parts: the first one creates the train plan by changing the arrival and departure times of the trains, while the other one evaluates the plan based on an estimation and simulation of passenger behavior. The authors claim that their extension leads to a more realistic evaluation of the passenger flows.

Kanai et al. (2011) propose various measures of passenger dissatisfaction as possible objective functions in their models. They develop a simulation model which simulates railway traffic and passenger flows. The simulation model is then used to calculate passenger dissatisfaction given a strategy for maintaining and dropping connections. Furthermore, the authors propose a tabu search algorithm that aims at finding good strategies for the management of train connections in case of delays. This algorithm incorporates the simulation model for railway traffic and passenger flows. The algorithm is tested on data of the Japanese railway network with promising results. The authors report an average computation time of about six minutes using a Core 2 Duo PC with Windows XP.

Eberlein (1998) investigated the real-time deadheading problem in a transit railway network. The real-time deadheading problem is to determine in real-time time which trains must be deadheaded and how many stations must be skipped by certain trains in order to minimize the total passenger waiting time. He formulated a MIP model and solved a case study from the Green Line of the Boston metro system, thereby showing that the model may improve the passenger waiting time significantly. He concludes that for practical purposes, the algorithms derived are highly applicable to the general model, fast for real-time use, and easy to implement. The computation time is no more than a few seconds using a 486 PC.

3.2.3 Disruptions - Microscopic and Macroscopic Approaches

In this section we discuss TTR for dealing with disruptions. We make a distinction between the papers that consider the infrastructure at a microscopic level and those that consider it at a macroscopic level.

Microscopic Approach Wiklund (2007) describes a simulation procedure called RailSys for simulating railway traffic on a microscopic level in order to determine the effectiveness of various recovery strategies in case of a disruption. A specific case study is considered, namely a case in June 2000 where a fire caused excessive damage to the interlocking system at the Järna station southwest of Stockholm. The effectiveness of five strategies were compared in terms of train traffic mileage and propagated delays. The results show that micro-simulation can be a useful tool for testing recovery strategies for such disruption scenarios.

Hirai et al. (2009) consider the problem of train stop deployment, where, in the event of a disruption in the network, decisions have to be made about where the obstructed trains should stop so that the unobstructed trains can still reach their destinations in other parts of the network. A Petri-net model is used to represent the problem and to model the transitions between potential stop locations for the trains. These do not necessarily include only platforms in stations, but also track segments outside stations. The problem is formulated as a MIP model aiming at minimizing the number of stops outside stations as well as the deviations from the original timetable. The method is tested on a number of real-world instances on the Japanese railway network. Solutions to two test cases were obtained in 12 seconds and 27 seconds, respectively, using a PC with 1.6 GHz and 1.49 GB RAM, showing that solutions may be obtained in a practical calculation time.

Corman et al. (2011a) study the case of handling disruptions in large and busy railway networks. In particular, they consider the case of double track railway lines where some block sections of one track are temporarily unavailable. Centralized and distributed approaches are presented: in the centralized approach the entire rescheduling problem is solved, while in the distributed approach a coordinator sets constraints between different areas and delegates the scheduling decisions to the local schedulers. Computational experiments on a large railway network in the Netherlands show that both approaches face increasing difficulty in finding feasible schedules in a short computation time for increasing time horizons of traffic prediction. The considered network includes the stations Utrecht Central Station, Arnhem and Den Bosch, and time horizons of 30, 60 or 90 minutes. It is shown for the time horizons of 30 and 60 minutes that the distributed approaches achieve average consecutive delays of less than 20 seconds, while for the same instances the centralized approaches achieve average consecutive delays of more than 20 seconds. For a time horizon of 90 minutes all algorithms achieve approximately the same average consecutive delays where feasible schedules were found, yet the distributed approaches produce a larger percentage of feasible solutions in this case.

Macroscopic Approach Real-time reordering of trains is considered by Shimizu et al. (2008) in a case study involving the Shinkansen railway network north of Tokyo in Japan. In this paper it is assumed that an earthquake has disrupted a large part of the railway network between two major stations, and that this disruption potentially affects the entire railway network. A constraint programming approach is developed which considers changing the departure order of trains from stations in order to minimize delays. Hereby constraints handle the deviations of the departure times from those in the original timetable. Practical solutions are found within 15 seconds of computation time.

Nakamura et al. (2011) present an algorithm for train rescheduling during disruptions which takes as input train groups, train cancellation sections, and return patterns. These factors are predetermined by the dispatchers. Here a train group consists of a set of trains that share the same assignment of rolling stock. Train cancellation sections are sections of the railway infrastructure bounded by two stations in which all trains are cancelled if a disruption occurs inside the section. Finally, a return pattern defines connections between trains in the same group at stations that bound a disrupted section. In case of a disruption obstructing a section of the network, the developed algorithm determines a new timetable by cancelling trains, combining return patterns, and changing the train departure order at stations in a series of steps. The efficiency of the rescheduling plan is evaluated in terms of passenger dissatisfaction caused by propagated delays, which is determined via a micro-simulation. The algorithm is tested on railway line in a metropolitan area in Japan. It is shown that good schedules can be obtained in less than 5 minutes using an Intel Core 2 Duo processor with 3.33 GHz and 3.25 GB RAM, making the algorithm suitable for real-time use.

Louwerse and Huisman (2012) develop a MIP model for designing an alternative periodic timetable in case the normal timetable cannot be operated due to a disruption. Slight changes in the departure and arrival times of the trains that are not cancelled are allowed. The authors take into account certain regularity aspects. For example, in case of a partial blockade of the tracks, they aim at operating approximately the same numbers of trains in each direction. Moreover, in such a situation they aim at operating both intercity trains and regional trains, so that in principle all passengers can be transported. Such constraints are also added in order to increase the probability that there will exist feasible rolling stock and crew schedules. Computational experiments based on instances of Netherlands Railways indicate that this method leads to less cancelations of trains than the contingency plans that are currently used in the Netherlands. The model is solved in a few seconds using CPLEX 12.4.

Albrecht et al. (2013) consider the problem of disruptions due to track maintenance, arising when longer maintenance operations are required than planned, leading to additionally cancelled trains. A problem space search meta-heuristic is developed to construct an integrated timetable which includes track maintenance. In addition, the heuristic is used as an operational tool where a revised timetable can be generated quickly to take into account the status quo of a disrupted system. The method is tested on a case study for a single track rail network in Queensland, Australia. It presents 50 trains that are operated in a time horizon of 24 hours with a scenario with 6 hours of unscheduled maintenance. The value of the best solutions found indicate improvements between 17% and 34% in total and maximum delay, respectively, when compared to the simulated manual rescheduling solutions.

4 Rolling Stock Rescheduling

If a disruption has occurred in a railway network, the original rolling stock allocation is usually no longer feasible. Due to the cancellation of a number of trips, several rolling stock units may not be able to carry out certain tasks in their duties. Thus the rolling stock needs to be rescheduled, using the updated timetable and the original rolling stock allocation as input.

In the planning stage of a railway system, rolling stock has been assigned to the trips in the timetable. Several papers have been published tackling the problem of allocating rolling stock to trips in the timetable according to the constraints and the objectives of the involved railway operator. Aspects to be taken into account are the allocation of rolling stock to lines, the matching of capacity and passenger demand, and the end-of-day balances in the stations. Other important aspects are related to the shunting possibilities inside the stations and to the depot planning. For surveys on these problems as well as on the use of combinatorial optimization techniques, see, e.g., Cacchiani et al. (2012); Caprara (2010); Caprara et al. (2007); Huisman et al. (2005).

4.1 Rolling Stock Rescheduling: Mathematical Formulation

The problem of rolling stock rescheduling may be formulated as a multi-commodity flow model on a graph where the nodes correspond to stations at specific times, while the arcs correspond to trips that have to be performed according to the timetable. Fioole et al. (2006) formulated the rolling stock allocation model as a multi-commodity flow model. Several rescheduling models are based on this model formulation. If the order of the train units in a train is important, then various train compositions are typically enumerated. One train composition must be assigned to each trip in the timetable. The objective is to minimize certain costs and to maximize the service for the passengers. Several constraints have to be taken into account, for example related to the shunting possibilities in the stations and the end-of-day rolling stock balances in the stations. In some papers, e.g. Cadarso and Marín

(2012), shunting movements are penalized, in order to increase the robustness of the railway system, since shunting movements may easily lead to technical failures. Other papers, e.g. Nielsen et al. (2012), penalize *changes* in the shunting plans, in order to keep the real-time shunting processes as much as possible the same as originally planned.

A generic model formulation for rescheduling train units of different types to trips on a single day due to a disruption, where the order of the train units in a train composition is important, can be described as follows. Let R be the set of trips to be performed, M the set of different train unit types, C the set of possible train compositions, $\mathcal{R} \subset R \times R$ the set of pairs of consecutive trips, and S the set of stations.

The following parameters are used in the model: first, the original allocation of composition c to trip r is denoted by $x^0_{r,c}$. Next, $I^0_{s,m}$ is the available inventory of train unit type m in station s by the start of the day, and $I^\infty_{s,m}$ is the desired inventory of train unit type m in station s by the end of the day. Furthermore, the parameter $a_{s,r,c,m}$ represents the increase or decrease in the inventory of train unit type m in station s caused by trip r using composition c. $R_{s,r} \subset R$ is the set of trips that arrive at or depart from station s before trip r departs from station s. The parameter τ denotes the time of rescheduling, and R_τ denotes the set of trips with start time earlier than τ, i.e. the set of trips that started already at the time of rescheduling (and that possibly finished already as well).

The following sets of decision variables are used in the model. First, $x_{r,c}$ indicates that composition $c \in C$ is assigned to trip $r \in R$. Next, $z_{r,c,c'}$ for $r \in R$ indicates that composition $c \in C$ is changed to composition $c' \in C$ in the station where trip r ends. Furthermore, $i^\infty_{s,m}$ is the realized inventory of train unit type m in station s by the end of the day. Finally, $d^+_{s,m}$ and $d^-_{s,m}$ denote the off-balances of train unit type m in station s by the end of the day.

Then the model can be described as follows:

$$\begin{array}{lllr}
\min & f(x,z,d) & & (6) \\
\text{subject to} & \sum_{c\in C} x_{r,c} = 1 & r \in R & (7) \\
& x_{r,c} = \sum_{c'\in C} z_{r,c,c'} & r \in \mathcal{R}, c \in C & (8) \\
& x_{r',c'} = \sum_{c\in C} z_{r,c,c'} & (r,r') \in \mathcal{R}, c' \in C & (9) \\
& I^0_{s,m} + \sum_{r'\in R_{s,r}} \sum_{c\in C} a_{s,r',c,m} x_{r',c} \geq 0 & s \in S, r \in R, m \in M & (10) \\
& I^0_{s,m} + \sum_{r\in R} \sum_{c\in C} a_{s,r,c,m} x_{r,c} = i^\infty_{s,m} & s \in S, m \in M & (11) \\
& i^\infty_{s,m} = I^\infty_{s,m} + d^+_{s,m} - d^-_{s,m} & s \in S, m \in M & (12) \\
& x_{r,c} = x^0_{r,c} & r \in R_\tau, c \in C & (13) \\
& x_{r,c} \in \{0,1\} & r \in R, c \in C & (14)
\end{array}$$

Note that (6)–(14) is a simplified version of the rolling stock scheduling model described by Nielsen et al. (2012). In this model, the objective (6) is a function of the decision variables $x_{r,c}$, $z_{r,c,c'}$ and $d_{s,m}$, which typically measures costs related to carriage kilometers, seat shortage kilometers, shunting movements, as well as differences between the original and the new rolling stock allocation, the deviation from the end of day balance, and the cancellation of trips due to lack of rolling stock.

Constraints (7) ensure that exactly one composition is assigned to each trip. Constraints (8) and (9) handle the allocation of possibly different compositions to consecutive trips in the timetable. Furthermore, constraints (10) guarantee that the inventory of each train unit type in each station at any time instant is not negative. Constraints (11) determine the realized inventory of each train unit type m in station s. Constraints (12) are used to determine the values of the off-balance decision variables $d^+_{s,m}$ and $d^-_{s,m}$. Constraints (13) guarantee that the rolling stock compositions on the trips that have started already at the time of rescheduling τ are not changed. Finally, constraints (14) specify the binary character of the composition assignment variables. All other decision variables are non-negative.

Constraints (12) and (13) are the main difference between this real-time rolling stock rescheduling model and the model of Fioole et al. (2006). However, another important aspect in a real-time rescheduling situation is that the passenger flows are different than they were planned to be, since in a disrupted situation the passengers will dynamically look for different routes to get to their destinations. This dynamic passenger behavior is described by Kroon et al. (2010) and Cadarso et al. (2013).

4.2 Rolling Stock Rescheduling: Literature

In this section, we present the main papers dealing with rolling stock rescheduling in case of a disruption. In most of the papers, the updated timetable is assumed to be given, together with the original rolling stock schedule.

Budai et al. (2010) introduce the Rolling Stock Rebalancing Problem (RSRP), which is a problem faced either in the short-term planning stage or in the real-time rescheduling stage in the event of a disruption of the railway network. The problem takes as input a timetable for a given planning period as well as an allocation of rolling stock units that is feasible except for possible off-balances in the rolling stock inventories by the end of the day. The problem is then to apply changes to the rolling stock duties with the objective of minimizing these off-balances, as well as the costs related to carriage kilometers, passengers in excess of carriage capacities, and the number of shunting operations. Two heuristic solution approaches to the RSRP are proposed in this paper. One is a greedy approach where small feasible transformations to the rolling stock allocation are carried out iteratively until no further improvements are possible. The other approach is a two-phase heuristic, where in the first phase a number of feasible transformations are identified, and in the second phase a subset of these transformations with minimum cost is selected using a MIP model. Instances based on the 3000 line of Netherlands Railways between Den Helder and Nijmegen are considered in this paper. The authors show that both heuristics are fast and can be used effectively both in the short-term planning stage and in real-time rescheduling in case of disruptions. The first heuristic has a computation time between 4 and 5 seconds for each run, and the second heuristic between 1 and 2 minutes using a PC equipped with a Pentium IV processor with 3.0 GHz and 1 GB RAM.

Nielsen et al. (2012) propose a rolling horizon solution approach for real-time rescheduling of rolling stock. Here, in the event of a disruption, the rolling stock is rescheduled periodically over a rolling horizon with a limited length, thereby taking into account new information concerning the disruption as time progresses. After a disruption has occurred, the input to the rescheduling algorithm is a sequence of updated timetables, each of which becomes available at a certain point in time. At such a point in time, no information is available yet about future updates of the timetable. The rolling stock is rescheduled over a rolling horizon with a MIP model. The model is an adapted version of the model by Fioole et al. (2006) and is similar to

(6)–(14). One objective is to minimize the deviations from the original rolling stock schedule. In particular, the criteria for solving such an instance include the number of cancelled trips, the changes to the shunting plans, and the predicted end-of-day off-balances of rolling stock at the stations. The authors consider case studies involving disruptions on the complex so-called North-East intercity lines of Netherlands Railways. Rolling stock rescheduling is complex here due to combining and splitting of trains on these lines. With the right parameter settings the algorithm finds solutions with small deviations from the original schedule within a short computation time. With a horizon of less than three hours, the computation time is always within a few seconds. Using a horizon larger than three and a half hour may lead to longer computation times with CPLEX 10.1 using a Pentium 4 with 2.8 GHz and 1 GB RAM.

In a subsequent paper, Kroon et al. (2010) consider real-time rescheduling of rolling stock in the case of a disruption, taking also the effect of dynamic passenger flows into account. Due to the disruption, unexpected passenger flows may occur along the possible detour routes. In this paper it is attempted to adapt the rolling stock capacity on the detour routes so as to facilitate these unexpected passenger flows. The solution approach uses a two-stage feedback loop. In the first stage, the rolling stock is rescheduled by solving a MIP model. In the second stage, the effect of the passenger flows on the current allocation of rolling stock is determined by means of a passenger simulation. The passenger simulation provides feedback in terms of the expected passenger delays due to insufficient capacity of the rolling stock. This feedback is used then in the objective function of the next iteration of the MIP model to reallocate the rolling stock again. Here the objective function also includes the same system related costs as in Nielsen et al. (2012). Given the reallocation of the rolling stock, the passenger simulation stage is carried out again. This process continues until no further improvement in the objective function is observed or a maximum number of iterations is reached. The algorithm was tested on a number of instances of Netherlands Railways. The authors report that the computations times are appealing for real-time use in most cases, namely up to 70 seconds with CPLEX 11.0 on an Intel Core 2 duo with 3.33 GHz and 3 GB RAM. However, larger cases need longer computation times. In addition, the application of this approach may lead to a significant reduction in the passenger delays in case of a disruption.

At DSB S-Tog, the suburban rail operator in the greater Copenhagen area in Denmark, an important recovery strategy in case of a major disruption in the network is to cancel all trains on the involved line of the network. When the disruption has ended, the cancelled trains must be reinserted as soon as possible. Jespersen-Groth et al. (2006) present a MIP model for calculating the best way to reinsert the cancelled trains in the network covered by a periodic timetable. If the reinsertion has begun from a certain rolling stock depot, the remaining trains to be inserted from that depot must be inserted according to frequency, so that at no time there occurs a frequency interval with an uncovered departure. Furthermore, also the time needed to supply the reinserted trains with a train driver must be taken into account. Test results on real-life instances show that the solutions are often better than the manually created solutions. Moreover, the running time of the rolling stock reinsertion model is approximately 0.5 seconds on average using CPLEX.

Jespersen-Groth (2008) considers several other problems and mathematical models related to rolling stock rescheduling at DSB S-Tog. The first model is the train unit position model. This model aims at assigning anonymous train units of two types to the trains in the timetable after a disruption has occurred. The model is to a large extent comparable to (6)–(14). The difference is that in this model the positions of the train units in the trains are considered explicitly. This is a feasible approach, since each train consists of at most two train units. The train unit position model does not consider physical train units yet. In the complementary train unit sequencing and routing models, physical train units are assigned to the positions in the trains. A large number of computational experiments have been carried out. It turns out that the computation times strongly depend on the parameter settings. The models cannot always be solved to optimality in a short computation time, but within a couple of minutes the optimality gap is within a few percent using CPLEX 10.1.

Sato et al. (2009) give a general formulation of the problem of reallocating resources to trips in a railway network in case of a disruption. Here the resources may refer either to rolling stock or to crew. It is assumed that the timetable has been rescheduled in case of a disruption. This updated timetable is provided as input to the reallocation problem together with the original resource duties. The objective is then to determine a reallocation of the resources which differs from the original plan as little as possible. As a solution method a two phase algorithm is implemented. The first phase attempts to resolve conflicts created by the disruption through small changes in the resource duties, starting from the original plan, and the second phase is a local search heuristic which attempts to iteratively improve the rescheduled resource duties. The algorithm is tested on one line of the Japanese railway network. Feasible solutions are obtained within 30 seconds when the rescheduling period is 72 hours or shorter, using CPLEX 12.1 on a Core i7 processor with 3.2 GHz and 3 GB RAM.

In a subsequent paper, Sato and Fukumura (2012) consider the problem of reassigning locomotives to tasks in the case of a disruption in the railway network. Here a task consists of hauling a number of freight carriages from one station to another. It is assumed that an updated timetable is given as input to the problem, where a disruption resulted in delays or the cancellation of trains. The approach adopted by the authors is to first enumerate possible sequences of tasks, and to determine the corresponding costs for each sequence. Then a MIP model based on set-partitioning is presented in order to assign locomotives to sequences of tasks with minimum cost, and a column generation technique is proposed as a solution approach. The column generation technique is to some extent comparable to what is usual for crew scheduling and rescheduling, as described in Section 5. First, the set-partitioning constraints are relaxed into set-covering constraints in order to obtain the restricted master problem. During the column generation process, a subset of set-partitioning constraints is restored. The column generation subproblem is in turn reduced to an unconstrained shortest path problem and solved using Dijkstra's algorithm. Based on the solutions found for instances from the Japan Freight Railway Company between Kuroiso and Shimonoseki in Japan, the authors conclude that locomotive reassignments can be found within a practical amount of time. A real-world case consisting of 786 trains and 185 locomotives required 422.5 seconds to be solved, using a Pentium 4 processor with 3.6 GHz and 2 GB RAM. In this case a locomotive assigned to a train broke down for 2 hours between 2 stations.

Almodóvar and García-Ródenas (2013) propose a real-time optimization method for rescheduling rolling stock in case of emergencies. These emergencies arise in the presence of a huge and unexpected passenger demand, which exceeds the service line capacity. The proposed method is based on a discrete-event simulation model, which determines how to reassign rolling stock from other lines to a line with high demand. The on-line optimization model is solved using two greedy heuristics which automatically generate near-optimal decisions about rolling stock reassignments. The objective is to minimize the total in-system time of the passengers. The proposed method is tested on a synthetic network as well as on a real network provided by RENFE, the Spanish national railway company. The results show that the first greedy heuristic provides near-optimal decisions, since it uses predictive simulations.

However, the computation times are high using an Intel Core 2 Duo E6850 processor with 4 GB RAM. With the second greedy algorithm a trade-off between accuracy and computation time is found: in particular, a single predictive simulation is performed within 3 minutes. Therefore the second greedy algorithm can be used in real-time.

5 Crew Rescheduling

The railway crew rescheduling problem deals with assigning tasks to train drivers and conductors after the timetable and the rolling stock have been rescheduled. Due to the rescheduled timetable and rolling stock, some crew duties may have become infeasible. For example, if trains are delayed, then some crew members may arrive too late to perform the next tasks in their duties. And if trains have been cancelled, then some crew members may not be able to reach their next tasks at all. In such cases, these tasks must be assigned to other crew members, which means that the tasks must be rescheduled.

When rescheduling the duties, the original crew schedule is used as input, since the rescheduled duties should not differ too much from the original ones. Furthermore, as many tasks as possible must be reassigned to the crew members, thereby respecting certain workload regulation rules. The most common rules for the duties used in practice are the following:

- A rescheduled duty may not be more than a certain amount of time longer than the original duty.
- A rescheduled duty has to end at the home depot of the corresponding driver.
- A rescheduled duty must have a certain number of breaks of a certain minimum length.
- A rescheduled duty must have a certain working time before and after a break.
- A rescheduled duty must have a certain transfer time when a crew member transfers from one train to another.

5.1 Crew rescheduling: Mathematical Formulation

The crew rescheduling problem can be formulated as an Extended Set Covering problem. In that sense, it is comparable to the crew scheduling problem in the planning stage. The main difference between the model for the rescheduling stage and the planning stage is that in the rescheduling stage the original duties of the crew members have to be taken into account, since the rescheduled duties may not differ too much from the original ones. See, among others, Abbink et al. (2011), Caprara et al. (1999a), Caprara et al. (1999b), and Lee and Chen (2003) for details on the crew scheduling problem in the planning stage.

In the model (15)–(19) for the crew rescheduling problem, D denotes the set of crew members who work on the day the disruption occurs, and who have not yet finished their duty yet at the moment that the disruption occurs. Note that it is necessary to reschedule the duties that have become infeasible, and that in principle the duties that are still feasible need not be rescheduled. Nevertheless, it may still be worthwhile to reschedule some of these duties to help the affected crew members, for instance by swapping tasks.

Furthermore, the set T represents all tasks that still have to be completed and that have not started at the beginning of the disruption. All these tasks must be reassigned to the available crew members. For every crew member j, the set K_j represents all possible completions to finish his/her duty on time. Each completion $k \in K_j$ coincides with the duty of crew member j until the time of rescheduling, since the past cannot be modified. The binary parameter a_{ij}^k indicates whether task i is covered by the completion k of crew member j ($a_{ij}^k = 1$) or not ($a_{ij}^k = 0$). The parameter c_j^k represents the cost of assigning completion k to crew member j, and the parameter f_i indicates the cost of cancelling task i.

The binary decision variable x_j^k represents whether completion k is assigned to crew member j ($x_j^k = 1$) or not ($x_j^k = 0$). The binary decision variable y_i indicates if task i can be covered by a crew member ($y_i = 0$) or not ($y_i = 1$). If $y_i = 0$, then it means that task i as well as the corresponding trip of a train must be cancelled. Now the extended Set Covering model can be described as follows:

$$\begin{aligned}
\min \quad & \sum_{j \in D} \sum_{k \in K_j} c_j^k x_j^k + \sum_{i \in T} f_i y_i && && (15)\\
\text{subject to} \quad & \sum_{j \in D} \sum_{k \in K_j} a_{ij}^k x_j^k + y_i \geq 1, && \forall i \in T && (16)\\
& \sum_{k \in K_j} x_j^k = 1 && \forall j \in D && (17)\\
& x_j^k \in \{0,1\} && \forall j \in D \; \forall k \in K_j && (18)\\
& y_i \in \{0,1\} && \forall i \in T && (19)
\end{aligned}$$

The objective function (15) aims at minimizing the rescheduling costs of the crew and the costs of cancelling additional tasks. Furthermore, constraints (16) ensure that each task is either covered by at least one completion or is cancelled. Constraints (17) ensure that every crew member gets exactly one completion of his/her duty. Finally, constraints (18) and (19) specify the binary character of the decision variables.

In the same way as in the crew scheduling problem for the planning stage, all complexities related to the rules that have to be satisfied by the individual duties are covered in the generation of the sets K_j of potential completions of duties.

The formulation of the problem as an extended Set Covering model is quite compact. However, if the number of tasks increases, then the number of potential completions of duties increases exponentially, resulting in high computation times. Therefore, advanced algorithms are designed, usually based on column generation and/or Lagrangian Relaxation.

In column generation, the idea is that not all potential completions of duties are taken into account from the beginning. The algorithm starts with only a small set of potential completions. At each iteration new promising potential completions are added to the current set of completions. The method continues until the solution cannot be improved anymore.

The idea of Lagrangian Relaxation is that some of the constraints are relaxed and considered only in the objective function. The advantage is that the resulting problem may be solved more easily. However the solutions are not feasible in general, due to the relaxation of the constraints. Therefore, additional methods must be constructed which make the solutions also feasible. Usually, heuristic methods are used for that purpose.

5.2 Crew Rescheduling: Literature

In this section we present the main papers dealing with crew rescheduling in case of disruptions. In most of the papers, the updated timetable, the updated rolling stock circulation and the current crew schedule are given. The goal is to find a new feasible crew schedule that covers as many tasks as possible, while taking the relevant rules into account.

The problem of crew rescheduling was studied much earlier in the airline industry than in the railway industry. There, the models are also based upon Set Covering models. For an overview of the existing literature in airline disruption management see Clausen et al. (2010).

In the airline industry the crew rescheduling problem is closely related to the crew rostering problem, since duties may cover several consecutive days. This is less the case in the railway industry where a duty often describes the tasks to be carried out on a single day. Only in cases with long distance trains or many overnight trains, the start location or start time of a duty on the next day can be influenced by a disruption on the previous day. This means that a new duty must be generated for the next day as well. However, the currently available literature does not take this crew rostering aspect into account.

Thus railway crew rescheduling is the problem of assigning completions of duties to crew members whose duties are affected by the disruption. It only recently got attention in the academic literature. To the best of our knowledge, Walker et al. (2005) were the first who investigated the crew rescheduling problem in the passenger railway industry. However, they formulated an approach to tackle crew rescheduling and timetable rescheduling in an integrated way. This approach will be discussed in the next section.

A group of papers which based their models on the Set Covering model are Huisman (2005), Rezanova and Ryan (2009), Potthoff et al. (2010) and Sato and Fukumura (2011). The models described in these papers are all tested on large real world instances. Although the general approach in these papers is the same, there is some variation in constraints (16). Huisman (2005) and Sato and Fukumura (2011) do not have the option of cancelling a task, and therefore they do not include the y_i variables. However, the models of Rezanova and Ryan (2009) and Potthoff et al. (2010), including the option of cancelling tasks, still aim at cancelling tasks as few as possible by putting a large penalty f_i on cancelling a task i.

For determining these penalties, Potthoff et al. (2010) distinguish between cancelling tasks that start and end at the same station and cancelling tasks that do not start and end at the same station. In the first case, it is much easier to adapt the corresponding rolling stock duties in case such a task is cancelled. The advantage of the models with the option of cancelling tasks is that they always provide a feasible solution. Another difference between the papers of Huisman (2005), Rezanova and Ryan (2009), Potthoff et al. (2010) and Sato and Fukumura (2011) is their approach to solve the Set Covering model.

Huisman (2005) does not focus on real-time crew rescheduling but on rescheduling the duties of train drivers of Netherlands Railways briefly before the operations. This may be necessary due to maintenance of the Dutch railway infrastructure. The model has as additional possibility the option to appoint extra crew members if necessary. The model is solved with a column generation algorithm where possible completions are generated via a shortest path formulation with resource constraints. The latter are required for dealing with the meal break constraints. The solution method is only applicable for the short term rescheduling stage, because the computation times are too long to be applied for real-time crew rescheduling (more than an hour using a Pentium IV processor with 3 GHz and 512 MB RAM).

The same model is used by Sato and Fukumura (2011) to reschedule the crew members of the Japanese Freight Railway Company in real-time in case of a disruption. This model is also solved with a column generation technique. Only the duties that are affected by the disruption are taken into account. The authors apply the model on a simulated disruption in an area in Japan with a high density of railway traffic. This disruption influences 102 duties and 413 tasks. The computation times are about 15 seconds, using a processor with 2.66 GHz and 3.5 GB RAM. This method can thus be used for real-time crew rescheduling.

Rezanova and Ryan (2009) and Potthoff et al. (2010) build on the experience of Huisman (2005). The model of Rezanova and Ryan (2009) has a small difference with the model (15)–(19). The model includes an extra binary variable e^k in constraints (17). This variable is equal to 1 if there is no completion of a duty available for a crew member and 0 otherwise. In Potthoff et al. (2010) this variable is redundant, since they include an artificial completion at a high cost which means that the crew member will travel home by taxi.

Furthermore, Rezanova and Ryan (2009) use equality constraints for constraints (16), but this does not really make a difference. In principle there is no problem if more drivers cover a certain task: a second driver can be a passenger on the train.

Experience from short term planning with the model (15)-(19) has indicated that it is not possible to consider all crew members and tasks in the real-time rescheduling model. This is due to long computation times. Obviously, at least all affected crew members and tasks must be taken into account. However more crew members may have to be rescheduled. Both Rezanova and Ryan (2009) and Potthoff et al. (2010) use this fact, but their approaches differ slightly in how they select the subsets of crew members and tasks.

Rezanova and Ryan (2009) suggest to use a dynamic column generation approach with a strategy in which the subsets of drivers and train tasks which are taken into account increase during the solution approach. They start with a small subset of the original duties (only the duties that are directly affected by the disruption) and a limited recovery period. As long as constraints are violated while solving the LP-relaxation, the subset of duties or the recovery period is expanded. This method has been tested on instances of DSB S-Tog, the suburban rail operator in the greater Copenhagen area. All instances are solved within 36 seconds of computation time, using a Pentium 4 processor with 340 GHz and 1 GB RAM.

Potthoff et al. (2010) first solve an initial problem containing the crew members with infeasible duties and some crew members which are in some sense close to the disrupted area. If a task cannot be covered by any crew member, then new problems, representing the neighborhood of an uncovered task, are solved iteratively. In the neighborhood of the uncovered task, they include in a first step a number of duties that currently cover tasks that depart around the same moment in time from the same station as the uncovered task. In a second step they add duties which are similar to the ones selected in the first step. The similarity of two duties is defined as the number of stations that are visited by both duties at roughly the same time. The problems are explored with a column generation based heuristic. Within the column generation procedure, Lagrangian relaxation is applied to the restricted master problems. The model has been tested on different instances of Netherlands Railways showing fast computation times. All instances are solved within 4 minutes on an Intel Pentium D processor with 3.4 GHz and 2 GB RAM. Currently, Netherlands Railways is testing the method of Potthoff et al. (2010) for real-time rescheduling of its train drivers. The method is currently also used for ultra-short term rescheduling of the crews of Netherlands Railways. That is, when the timetable, the rolling stock schedule and the crew schedules have to be updated for the operations of tomorrow because of bad weather forecasts.

Potthoff (2010) also applied dynamic constraint aggregation (DCA) to the crew rescheduling problem. The idea behind DCA is that not every task is treated individually, but that a number of tasks are aggregated in a cluster. Each cluster is represented in the model by one aggregated constraint. The resulting model is the aggregated master problem. A subset of the columns is considered in the aggregated restricted master problem. In principle, the problem can be solved faster in this way. However, the computation time needed to find solutions for large and difficult instances turned out to be too high for practical purposes, using an Intel Pentium D processor with 3.4 GHz and 2 GB RAM.

Abbink et al. (2009) use an agent based system for constructing solutions for real-time crew rescheduling problems. In this approach, crew members are represented by driver-agents who negotiate about rescheduling options in case of a disruption: they can swap parts of their duties with other driver-agents. If, due to a disruption, a driver-agent cannot cover one of his tasks anymore, he asks the other driver-agents if they can fulfill this task. All driver-agents will look if they can take over the task. It can happen that the task fits in the duty of a driver-agent without any problem, for example if the driver-agent had a long break. On the other hand, in most cases the driver-agent can only take over the task if he skips some other tasks of his duty. For these tasks, he asks all other driver-agents if they can take over these tasks. All together, this may lead to quite complex exchanges of tasks.

A simple example is the following: driver-agent 1 cannot carry out task 1, and asks which other driver-agent can take over this task 1; driver-agent 2 can do this task if someone else takes over his tasks 2 and 3; driver-agent 3 can take over task 2 without any problem, and driver-agent 4 can take over task 3 without any problem. This is an example of exchanges where in the end each task is covered by a driver-agent. If such an exchange is found and no better exchanges can be found, the algorithm stops. The authors tested the application on several real-life instances of Netherlands Railways. The approach seemed to perform quite well on relatively small instances. However, for large instances the computation times were too high, on an Intel Pentium processor with 3.4 GHz and 2.0 GB RAM. It turned out that the method could not compete with more mathematically oriented crew rescheduling methods.

6 Integrated Approaches

In this section, we present papers that deal with the integration of different phases of real-time railway rescheduling, with the aim of determining a good new schedule for the timetable, and the rolling stock and crew duties when a disruption occurs. By treating several rescheduling phases at the same time, it can be expected that better solutions are obtained. On the other hand, the integrated models are usually much more complex and difficult to solve than the models dealing with a single phase. This is especially relevant in the real-time rescheduling stage, where the rescheduling problems must be solved soon.

In the planning stage, more computation time is available for solving complex models. For example, Cadarso and Marín (2012) propose an integrated MIP model to adapt the frequencies in the timetable together with the rolling stock circulation in order to deal with increased passenger demand and traffic congestion in a rapid transit network. They also take into account the shunting of rolling stock in the depots. Shunting operations are heavily penalized in order to increase the robustness of the solution. On a PC with an Intel Core 2 Quad Q9950 processor with 2.83 GHz and 8 GB RAM, this complex integrated model leads to computation times of up to one hour on instances from RENFE's "Cercanías Madrid". This is not yet appropriate for real-time rescheduling purposes.

6.1 Timetable and Rolling Stock Rescheduling

Adenso-Díaz et al. (1999) develop an optimization model for real-time timetable and rolling stock rescheduling in case of large delays of one or more trains. In this situation, each train must consist of exactly one rolling stock unit. Therefore the problems of timetable rescheduling and rolling stock rescheduling are strongly connected with each other. The combined problem is described as a single MIP model. However, since the model appeared to be too large to be solved by MIP solvers, the authors develop a heuristic backtracking method that develops a search tree to find a good feasible solution in a heuristic way. The search space is limited by applying several rules, for example by assuming that each event cannot be delayed more than a certain number of minutes. The model was implemented in a Decision Support System that is used by the Spanish railway company RENFE for certain regional train lines in Asturias. For these regional train lines the system produces appealing solutions usually within five minutes, which is sufficient for providing real-time support.

Cadarso et al. (2013) consider the integration of the rescheduling of the timetable and the rolling stock in a single model. They consider the inclusion of additional trips, the cancellation of trips, and the possible allocation of additional rolling stock in order to alleviate some of the negative effects of the disruption. They formulate a MIP model, where the objective includes costs related to the operation of train services and empty movements, the number of unattended passengers, and the allocation of additional rolling stock. The dynamic behavior of the passengers due to the disruption is modeled by a multi-nomial Logit model. The model was tested on instances from RENFE's "Cercanías Madrid". The instances were solved in a few minutes using an Intel Core 2 Quad processor with 2.83 GHz and 8 GB RAM. The obtained results were received positively by the operator's dispatchers.

Fekete et al. (2011) attempt to integrate models for recovering the timetable and for rescheduling the rolling stock in case of a disruption. They specifically focus on subway networks. Their model takes as input the original timetable and rolling stock allocation together with information regarding the disruption of the network. A new timetable and rolling stock allocation are then determined using a MIP model, where the objective is to minimize the number of cancelled trips, while minimizing the total delay of the trains is treated as a secondary objective. The model is used to solve four instances of Vienna's subway network. It is shown that the computation times to obtain feasible solutions that are close to optimal are small enough to be used in real-time practice. Indeed, on an Intel Core 2 Duo processor with 3.0 GHz and 2 GB of RAM, an appropriate solution is always found within 60 seconds.

Veelenturf et al. (2011) aim at integrating timetable and rolling stock rescheduling. Their paper focuses on minimizing the dissatisfaction for the passengers during a disruption. Due to the disruption, the passenger flows may change. This requires the rolling stock duties and/or the timetable to be changed accordingly, in order to provide sufficient seating capacity for the modified passenger flows. The authors integrate these two rescheduling phases, but they limit the timetable adaptations to decisions on the stopping patterns of the trains. The rolling stock rescheduling is based on the multi-commodity flow model of Fioole et al. (2006) and Nielsen et al. (2012). In the passenger simulation model it is assumed that the passengers know the modified timetable, and that they try to get to their destination as soon as possible, as in Kroon et al. (2010). The model is tested on a number of instances of Netherlands Railways, showing that adapting the timetable can have a positive effect indeed. On average an instance is solved within 15 minutes using CPLEX 12.3.

6.2 Timetable and Crew Rescheduling

Walker et al. (2005) present a simultaneous approach for solving the timetable and crew rescheduling problem. They formulate the problem as a MIP model with two distinctive blocks for the timetable and for the crew schedule. These blocks are coupled with constraints on the sequencing of the tasks and on the shift lengths. Their objective is to minimize the deviation from the original schedule. The problem is solved by an LP relaxation with column generation and branch-and-bound. The model is tested on the Wellington Metro line, covered by 36 trains, with a time window of 18 hours. The model is tested by applying different disruption scenarios to the schedule. It takes between 26 and 120 seconds to solve the integrated problem instances on a Pentium 200 MHz computer with 64 MB of RAM. This approach works well for this kind of relatively small problems, but for countrywide complex railway systems with thousands of trains a day, computation times will increase enormously.

Finally, the model of Potthoff et al. (2010) is extended by Veelenturf et al. (2012) by allowing the possibility of delaying the departure of some trains. Therefore a number of copies of the considered tasks are made in which the departure and arrival times are slightly modified. Only one copy of every original task must be covered by a duty. This is a first step in integrating the timetable and crew rescheduling phase. In this approach usually less trains are cancelled in case of a disruption. Indeed, it may happen that some tasks cannot be covered by any crew member in the regular approach, but can be covered if the departure and arrival times of the corresponding trains are slightly changed. Computational experiments are carried out on two cases using a quad core processor with 2.99 GHz and 3.25 GB RAM. The extra flexibility of the model comes at the price of increased computation times.

7 Conclusions and Future Research

In this paper we present a review of the state-of-the-art in recovery algorithms for disturbance and disruption management in railway systems. In particular, we consider algorithms for timetable, rolling stock, and crew rescheduling, and algorithms that consider the integration of different rescheduling phases. We describe various approaches to solve these rescheduling problems in terms of the type and scale of the disturbances or disruptions, the network infrastructure and topology, the considered objective functions and constraints, and the utilized optimization methods.

Most papers about timetable rescheduling deal with relatively small delays of many trains, instead of large disruptions, i.e. huge delays of some trains or the unavailability of some tracks during a certain time interval, requiring the cancellation of a number of trains. Disturbances are assumed to not affect the feasibility of the rolling stock and crew duties. Thus for this kind of disturbances the timetable rescheduling problem, including the consequences for passengers or freight, can be solved on its own, without considering the consequences for rolling stock and crew. The current research mainly focused on instances representing relatively small railway networks where the infrastructure is considered at a microscopic level. Further research may lead to applications on larger railway networks. This may require the railway network to be considered at a macroscopic level.

In the context of disruptions, most of the papers deal with a single rescheduling phase. There are just a few papers that integrate two phases, namely either timetable and rolling stock rescheduling, or timetable and crew rescheduling. Future research should be devoted to the further integration of the three main rescheduling phases (timetable, rolling stock and crew rescheduling). One possible strategy for obtaining a full integration can be to design a framework that consists of a closed loop in which each rescheduling phase is solved by an efficient algorithm to find a good feasible solution and gets feedback from the other phases in order to obtain a good feasible solution for the whole system.

Although most of the presented models show promising results in the described experiments, it will be a large challenge to bring these methods into the real-time railway operations. As was mentioned in the introduction already, 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. Unfortunately, the railway industry has never been a quick adopter of newly available and innovative methods and concepts. Nevertheless, there are signals that practice has discovered the added value that can be provided by real-time rescheduling methods, based on the successes that have been achieved by the application of optimization methods in the railway planning stage.

Currently, within several railway companies, development projects are being carried out focusing on the implementation of the developed optimization models in practice. For example, within Netherlands Railways, the real-time crew rescheduling model of Potthoff et al. (2010) is currently being implemented. However, such implementation projects take a lot of time, since the models have to be integrated with the real-time railway information systems, which is a complex and risky operation. Moreover, the use of real-time algorithmic tools influences the working methods of the railway dispatchers. Adapting the dispatching organization and getting it acquainted with these new working methods is also a time-consuming process: the railway industry is not known to be very quick in adopting new innovations.

These are relevant differences between the planning stage and the real-time rescheduling stage. The use of decision support tools for supporting planning processes can be gradually introduced into practice, which is less the case for real-time rescheduling.

As a consequence, despite the large amount of research that has been accomplished in the last decades, the *regular* application of the research results in practice will still need quite some time. In addition, once these results are being applied in practice, new research challenges in this area will come up, for example how to deal in an integrated way with the inherent uncertainty and dynamics in a real-time rescheduling environment.

References

E.J.W. Abbink, D.G.A. Mobach, P.J. Fioole, L.G. Kroon, E.H.T van der Heijden, and N.J.E. Wijngaards. Actor-agent application for train driver rescheduling. *Proceedings of the 8th international conference on autonomous agents and multiagent systems*, 1:513–520, 2009.

E.J.W. Abbink, L. Albino, T. Dollevoet, D. Huisman, J. Roussado, and R.L. Saldanha. Solving large scale crew scheduling problems in practice. *Public Transport*, 3:149–164, 2011.

R. Acuna-Agost, P. Michelon, D. Feillet, and S. Gueye. A MIP-based local search method for the railway rescheduling problem. *Networks*, 57(1):69–86, 2011a.

R. Acuna-Agost, P. Michelon, D. Feillet, and S. Gueye. SAPI: Statistical Analysis of Propagation of Incidents. A new approach for rescheduling trains after disruptions. *European Journal of Operational Research*, 215(1):227–243, 2011b.

J. Adams, E. Balas, and D. Zawack. The shifting bottleneck procedure for job shop scheduling. *Management Science*, 34:391–401, 1988.

B. Adenso-Díaz, M. Oliva González, and P. González-Torre. On-line timetable rescheduling in regional train services. *Transportation Research Part B*, 33:387–398, 1999.

A.R. Albrecht, D.M. Panton, and D.H. Lee. Rescheduling rail networks with maintenance disruptions using problem space search. *Computers & Operations Research*, 40:703–712, 2013.

T. Albrecht, A. Binder, and C. Gassel. An overview on real-time speed control in rail-bound public transportation systems. In *Proceedings of the 2nd International Conference on Models and Technologies for Intelligent Transportation Systems, Leuven, Belgium*, 2011.

M. Almodóvar and R. García-Ródenas. On-line reschedule optimization for passenger railways in case of emergencies. *Computers & Operations Research*, 40(3):725–736, 2013.

M. Boccia, C. Mannino, and I. Vasilyev. The dispatching problem on multitrack territories: Heuristic approaches based on mixed integer linear programming. *Networks*, page doi: 10.1002/net.21528, 2013.

G. Budai, G. Maróti, R. Dekker, D. Huisman, and L.G. Kroon. Rescheduling in passenger railways: the rolling stock rebalancing problem. *Journal of Scheduling*, 13:281–297, 2010.

V. Cacchiani and P. Toth. Nominal and robust train timetabling problems. *European Journal of Operational Research*, 219(3):727–737, 2012.

V. Cacchiani, A. Caprara, and P. Toth. Models and algorithms for the train unit assignment problem. In A.R. Mahjoub et al., editor, *Proceedings of ISCO 2012*, volume 7422 of *LNCS*, pages 24–35. Springer-Verlag Berlin Heidelberg, 2012.

L. Cadarso and Á Marín. Integration of timetable planning and rolling stock in rapid transit networks. *Annals of Operations Research*, 199(1):113–135, 2012.

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

G. Caimi, M. Fuchsberger, M. Laumanns, and M. Lüthi. A model predictive control approach for discrete-time rescheduling in complex central railway station areas. *Computers & Operations Research*, 39:2578–2593, 2012.

A. Caprara. Almost 20 years of combinatorial optimization for railway planning: from Lagrangian Relaxation to Column Generation. In T. Erlebach and M. Lübbecke, editors, *Proceedings of the 10th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS10)*, OpenAccess Series in Informatics (OASIcs), 2010.

A. Caprara, M. Fischetti, P.L. Guida, P. Toth, and D. Vigo. Solution of large scale railway crew planning problems: the Italian experience. *Computer Aided Transit Scheduling, Springer-Verlag*, 1999a.

A. Caprara, M. Fischetti, and P. Toth. A heuristic method for the set covering problem. *Operations Research*, 47:730–743, 1999b.

A. Caprara, L.G. Kroon, M. Monaci, M. Peeters, and P. Toth. Passenger railway optimization. In C. Barnhart and G. Laporte, editors, *Handbooks in Operations Research and Management Science*, volume 14, pages 129–187. Elsevier, Amsterdam, 2007.

A. Caprara, L.G. Kroon, and P. Toth. Optimization problems in passenger railway systems. In J.J. Cochran, L.A. Cox, P. Keskinocak, J.P. Kharoufeh, and J. Cole Smith, editors, *Wiley Encyclopedia of Operations Research and Management Science*, volume 6, pages 3896–3905. Wiley, 2011.

C.K. Chiu, C.M. Chou, J.H.M. Lee, H. Leung, and Y.W. Leung. A constraint-based interactive train rescheduling tool. *Constraints*, 7(2):167–198, 2002.

J. Clausen, A. Larsen, J. Larsen, and N.J. Rezanova. Disruption management in the airline industry-concepts, models and methods. *Computers & Operations Research*, 37:809–821, 2010.

F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo. Evaluation of a green wave policy in real-time railway traffic management. *Transportation Research Part C*, 17(6):607–616, 2009.

F. Corman, A. D'Ariano, and I.A. Hansen. Disruption handling in large railway networks. *Proceeding of Computers in Railways, Beijing, China*, pages 629–640, 2010a.

F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo. Centralized versus distributed systems to reschedule trains in two dispatching areas. *Public Transport*, 2(3):219–247, 2010b.

F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo. A tabu search algorithm for rerouting trains during rail operations. *Transportation Research Part B*, 44(1):175–192, 2010c.

F. Corman, A. D'Ariano, I.A. Hansen, D. Pacciarelli, and M. Pranzo. Dispatching trains during seriously disrupted traffic situations. In *Proceeding of the IEEE International Conference on Networking, Sensing and Control*, pages 323–328, 2011a.

F. Corman, A. D'Ariano, M. Pranzo, and I.A. Hansen. Effectiveness of dynamic reordering and rerouting of trains in a complicated and densely occupied station area. *Transportation Planning and Technology*, 34(4):341–362, 2011b.

F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo. Bi-objective conflict detection and resolution in railway traffic management. *Transportation Research Part C*, 20(1):79–94, 2012.

A. D'Ariano. *Improving Real-Time Train Dispatching: Models, Algorithms and Applications.* PhD thesis, TRAIL Thesis Series T2008/6, The Netherlands, 2008.

A. D'Ariano and M. Pranzo. An advanced real-time train dispatching system for minimizing the propagation of delays in a dispatching area under severe disturbances. *Networks and Spatial Economics*, 9(1):63–84, 2009.

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

A. D'Ariano, M. Pranzo, and I.A. Hansen. Conflict resolution and train speed co-ordination for solving real-time timetable perturbations. *IEEE Transactions on Intelligent Transportation Systems*, 8(2): 208–222, 2007b.

A. D'Ariano, F. Corman, D. Pacciarelli, and M. Pranzo. Reordering and local rerouting strategies to manage train traffic in real time. *Transportation Science*, 42(4):405–419, 2008a.

A. D'Ariano, D. Pacciarelli, and M. Pranzo. Assessment of flexible timetables in real-time traffic management of a railway bottleneck. *Transportation Research Part C*, 16(2):232–245, 2008b.

T. Dollevoet. *Delay Management and Dispatching in Railways.* PhD thesis, Erasmus University Rotterdam, 2013.

T. Dollevoet, D. Huisman, M. Schmidt, and A. Schöbel. Delay management including capacities of stations. In A. Caprara and S. Kontogiannis, editors, *Proceedings of the 11th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems*, volume 20 of *OpenAccess Series in Informatics (OASIcs)*, pages 88–99, Dagstuhl, Germany, 2011. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

T. Dollevoet, D. Huisman, M. Schmidt, and A. Schöbel. Delay management with rerouting of pasengers. *Transportation Science*, 46:74–89, 2012.

X.J. Eberlein. Real-time deadheading problem in transit operations control. *Transportation Research B*, 32:77–100, 1998.

S.P. Fekete, A. Kröller, M. Lorek, and M. Pfetsch. Disruption management with rescheduling of trips and vehicle circulations. *5^{th} ASME/ASCE/IEEE Joint Rail Conference*, 2011.

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

M. Fischetti and A. Lodi. Local branching. *Mathematical Programming*, 98(1):23–47, 2003.

L. Gély, G. Dessagne, and C. Lerin. Modelling train re-scheduling with optimization and operational research techniques: Results and applications at SNCF. *Technical Report*, 2006.

C. Hirai, T. Kunimatsu, N. Tomii, S. Kondou, and M. Takaba. A train stop deployment planning algorithm using a petri-net-based modelling approach. *Quarterly Report of RTRI*, 50(1):8–13, 2009.

D. Huisman. A column generation approach for the rail crew re-scheduling problem. *European journal of Operational Research*, 53:1007–1023, 2005.

D. Huisman, L.G. Kroon, R.M. Lentink, and M.J.C.M. Vromans. Operations Research in passenger railway transportation. *Statistica Neerlandica*, 59:467–497, 2005.

J.R. Jackson. Scheduling a production line to minimize maximum tardiness. *Research Report 43, Management Science Research Project*, 1955.

J.J. Jespersen-Groth. *Decision Support for the Rolling Stock Dispatcher.* PhD thesis, DTU, Copenhagen, Denmark, 2008.

J.J. Jespersen-Groth, J. Clausen, and J. Larsen. Optimal reinsertion of cancelled train lines. Technical report, IMM-DTU, aug 2006. URL `http://www2.imm.dtu.dk/pubdb/p.php?5095`.

J.J. Jespersen-Groth, D. Potthoff, J. Clausen, D. Huisman, L.G. Kroon, G. Maróti, and M.N. Nielsen. Disruption management in passenger railway transportation. In R.K. Ahuja, R.H. Möhring, and C. Zaroliagis, editors, *Robust and Online Large-Scale Optimization*, volume 5868 of *LNCS*, pages 399–421. Springer, 2009.

S. Kanai, K. Shiina, S. Harada, and N. Tomii. An optimal delay management algorithm from passengers' viewpoints considering the whole railway network. *Journal of Rail Transport Planning & Management*, 1:25–37, 2011.

B. Khosravi, J.A. Bennell, and C.N. Potts. Train scheduling and rescheduling in the UK with a modified shifting bottleneck procedure. In D. Delling and L. Liberti, editors, *Proceedings of the 12th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems*, volume 25 of *OpenAccess Series in Informatics (OASIcs)*, pages 120–131, Dagstuhl, Germany, 2012. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

L.G. Kroon, G. Maróti, and L.K. Nielsen. Rescheduling of railway rolling stock with dynamic passenger flows. *Forthcoming in: Transportation Science*, ERS-2010-045-LIS, 2010.

K. Kumazawa, K. Hara, and T. Koseki. A novel train rescheduling algorithm for correcting disrupted train operations in a dense urban environment. In I.A. Hansen, editor, *Timetable Planning and Information Quality.* Wit Press/Computational Mechanics, 2010.

L. Lamorgese and C. Mannino. An exact decomposition approach for the real-time train dispatching problem. Technical Report N. A23274, SINTEF ICT, Norway, 2012.

L. Lamorgese and C. Mannino. The track formulation for the train dispatching problem. *Electronic Notes in Discrete Mathematics*, 41:559–566, 2013.

C. Lee and C. Chen. Scheduling of train driver for Taiwan Railway Administration. *Journal of the Eastern Asia Society for Transportation Studies*, 5:292–306, 2003.

I. Louwerse and D. Huisman. Adjusting a railway timetable in case of partial or complete blockades. Technical Report EI-2012-23, Econometric Institute of Erasmus University Rotterdam, 2012.

R.M. Lusby, J. Larsen, M. Ehrgott, and D.M. Ryan. Railway track allocation: Models and methods. *OR Spectrum*, 33(4):843–883, 2011.

R.M. Lusby, J. Larsen, M. Ehrgott, and D.M. Ryan. A set-packing inspired method for real-time junction train routing. *Computers & Operations Research*, 40:713–724, 2013.

M. Lüthi, G. Medeossi, and A. Nash. Evaluation of an integrated real-time rescheduling and train control system for heavily used areas. In *Proceedings of the International Seminar on Railway Operations Modelling and Analysis (IAROR) 2007 Conference, Hannover, Germany*, 2007.

C. Mannino. Real-time traffic control in railway systems. In *Proceedings of ATMOS'11*. A. Caprara and S. Kontogiannis (Eds.), OASICS Vol.20, 2011.

C. Mannino and A. Mascis. Optimal real-time traffic control in metro stations. *Operations Research*, 57(4):1026–1039, 2009.

A. Mascis and D. Pacciarelli. Job-shop scheduling with blocking and no-wait constraints. *European Journal of Operational Research*, 143(3):498–517, 2002.

A. Mascis, D. Pacciarelli, and M. Pranzo. Train scheduling in a regional railway network. In *Proceedings of the Fourth Triennial Symposium on Transportation Analysis, Sao Miguel, Portugal*, pages 487–492, 2001.

A. Mascis, D. Pacciarelli, and M. Pranzo. Scheduling models for short-term railway traffic optimization. *Computer-Aided Systems in Public Transport, Lecture Notes in Economics and Mathematical Systems*, 600, 2007.

L. Meng and X. Zhou. Robust single-track train dispatching model under a dynamic and stochastic environment: A scenario based rolling horizon solution approach. *Transportation Research Part B*, 45:1080–1102, 2011.

Y.H. Min, M.J. Park, S.P. Hong, and S.H. Hong. An appraisal of a column-generation-based algorithm for centralized train-conflict resolution on a metropolitan railway network. *Transportation Research Part B*, 45:409–429, 2011.

T. Nakamura, C. Hirai, and Y. Nishioka. A practical train rescheduling algorithm using three predetermined factors. In *Proceedings of the 4th International Seminar on Railway Operations Modelling and Analysis - RailRome 2011*, Rome, Italy, February 2011.

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

P. Pellegrini, G. Marlière, and J. Rodriguez. Real time railway traffic management modeling track-circuits. In D. Delling and L. Liberti, editors, *Proceedings of the 12th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems*, volume 25 of *OpenAccess Series in Informatics (OASIcs)*, pages 23–34, Dagstuhl, Germany, 2012. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

D. Potthoff. *Railway Crew Rescheduling: Novel Approaches and Extensions*. PhD thesis, Erasmus University, Rotterdam, The Netherlands, 2010.

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

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

J. Rodriguez. A constraint programming model for real-time train scheduling at junctions. *Transportation Research Part B*, 41:231–245, 2007.

K. Sato and N. Fukumura. Real-time freight train driver rescheduling during disruptions. *IEICE Transactions*, 94(6):1222–1229, 2011.

K. Sato and N. Fukumura. Real-time freight locomotive rescheduling and uncovered train detection during disruption. *European Journal of Operational Research*, 221:636–648, 2012.

K. Sato, S. Sakikawa, T. Morita, N. Ueki, and T. Murata. Crew and vehicle rescheduling based on a network flow model and its application to a railway train operation. *IAENG International Journal of Applied Mathematics*, 39(3), 2009.

A. Schaafsma and M. Bartholomeus. Dynamic traffic management in the Schiphol bottleneck. In I.A. Hansen, A. Radtke, J. Pachl, and E. Wendler, editors, *Proceedings of the Second International Seminar on Railway Operations Modelling and Analysis, Hannover, Germany*, 2007.

M. Schachtebeck and A. Schöbel. To wait or not to wait and who goes first? Delay management with priority decisions. *Transportation Science*, 44:307–321, 2010.

A. Schöbel. Integer programming approaches for solving the delay management problem. In F. Geraets, L.G. Kroon, A. Schöbel, D. Wagner, and C. Zaroliagis, editors, *Robust and Online Large-Scale Optimization*, volume 4359 of *LNCS*, pages 145–170. Springer, 2007.

A. Schöbel. Capacity constraints in delay management. *Public Transport*, 1(2):135–154, 2009.

H. Shimizu, H.Tanabe, and M. Yamamoto. The proposal system for Shinkansen using constraint programming. In *Proceedings of the 8th World Congress of Railway Research*, 2008.

J. Törnquist. Design of an effective algorithm for fast response to the rescheduling of railway traffic during disturbances. *Transportation Research Part C*, 20:62–78, 2012.

J. Törnquist and J.A. Persson. N-tracked railway traffic re-scheduling during disturbances. *Transportation Research Part B*, 41(3):342–362, 2007.

L.P. Veelenturf, L.K. Nielsen, G. Maróti, and L.G. Kroon. Passenger oriented disruption management by adapting stopping patterns and rolling stock schedules. In *Proceedings of the 4th International Seminar on Railway Operations Modelling and Analysis - RailRome 2011*, Rome, Italy, February 2011.

L.P. Veelenturf, D. Potthoff, D. Huisman, and L.G. Kroon. Railway crew rescheduling with retiming. *Transportation Research Part C*, 20:95–110, 2012.

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

M. Wiklund. Serious breakdowns in the track infrastructure - calculation of the effects on rail traffic. In *Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis - RailHannover 2007*, Hannover, Germany, March 2007.