



Railway crew scheduling with semi-flexible timetables

Christian Rählmann¹ · Ulrich W. Thonemann¹ 

Received: 22 June 2019 / Accepted: 15 May 2020 / Published online: 19 June 2020
© The Author(s) 2020

Abstract

We investigate the impact of coordinating the timetable and the crew schedule in an operational freight railway system. Usually, those problems are solved sequentially—resulting in suboptimal schedules for train drivers due to large idle times between two train rides. We seek to coordinate the timetable and the crew schedule on the operational level by adding flexibility to the timetable. We introduce small time windows that allow to shift entire trains forwards and backwards by discrete time periods. We present a mathematical model and solve it with a column generation heuristic. We test our model on three real datasets of a major European Freight Railway Operator and show that significant reduction in idle time and cost can be achieved.

Keywords Scheduling · Railway crew scheduling · Timetable · Transportation

1 Introduction

Freight railway operators must coordinate their resources (track capacity, engines and train drivers) to transport their customers' goods. Typically, the companies split their resource planning process into three phases (Kroon et al. 2009; Huisman et al. 2005b; Lusby et al. 2011): timetabling, engine scheduling and crew scheduling. In timetabling, the arrival and departure times of the trains are determined based upon the track availability. In engine scheduling, engines are assigned to each train. Finally in crew scheduling, different tasks are combined to duties into a crew schedule.

The companies solve these phases separately due to the complexity of the large railway networks. However, the integration of these problems will most likely free great potential because it can coordinate the resources simultaneously and therefore use them more efficiently. The full integration of the three phases for practical problem sizes

✉ Christian Rählmann
christian.raehlmann@uni-koeln.de

¹ Department of Supply Chain Management and Management Science, University of Cologne, 50923 Cologne, Germany

seems not feasible at the moment. Research focuses on integrating adjacent phases of the planning process, that is, integrating the timetabling with the engine scheduling (e.g. Bach et al. 2015; Schmid and Ehmke 2015) or integrating the engine scheduling with the crew scheduling (e.g. Kliwer et al. 2012).

These approaches cannot address one of the main problems in crew scheduling, low productivity of duties, that is, long idle time of train drivers when switching trains (see Jütte and Thonemann 2015 for a detailed analysis for waiting time per trip connection). At a major European Freight Railway Operator (EFRO), who initiated our research, train drivers frequently wait at train stations in order to drive the next train back to their home depot. These long waiting times can be attributed to the underlying timetables and not to the engine schedule, such that these cannot be reduced by integrating the engine scheduling problem with the crew scheduling problem, but require a coordination between the timetable and the crew schedule. This problem is addressed by Bach et al. (2016) for strategic planning problems, in which the annual timetable is coordinated with the crew schedule.

We focus on the operational problem to coordinate the timetable and the crew schedule. At EFRO, the operational coordination is of special importance, since 20% of all trains are scheduled at short notice and are not incorporated in the annual timetable. Customers requesting short notice trains usually require quick confirmations, such that these trains are inserted sequentially into the timetable over a period of up to several weeks. This large number of additional trains destroys the coordination between the annual timetable and crew schedule and makes an operational coordination necessary. We coordinate the timetable and the crew schedule by introducing time shifts that allow to shift entire trains forwards and backwards to improve crew scheduling, therefore adding flexibility to the timetable. At EFRO, small time shifts of up to 15 min typically do not affect the feasibility or the cost of the engine schedule, since cargo trains have substantial idle times between two engine duties.

We adjust the mathematical formulation of the crew scheduling problem for coordinating the timetable with the crew schedule and present a column generation heuristic that results in good solutions within reasonable computation time. In particular, we focus on deriving good integer solutions since the small time shifts of the trains in the operational setting increase the computational complexity compared to long time shifts in the strategic setting. We perform computational experiments based on three real-world datasets and show that small adjustments to the timetable significantly improve the crew scheduling cost.

The remainder of this paper is structured as follows. In Sect. 2, we review the relevant literature. In Sect. 3, we introduce the problem and explain railway crew scheduling related terms. In Sect. 4, we present our model and in Sect. 5 the solution method. In Sect. 6, we analyze the results on our computational experiments. In Sect. 7, we critically discuss our findings.

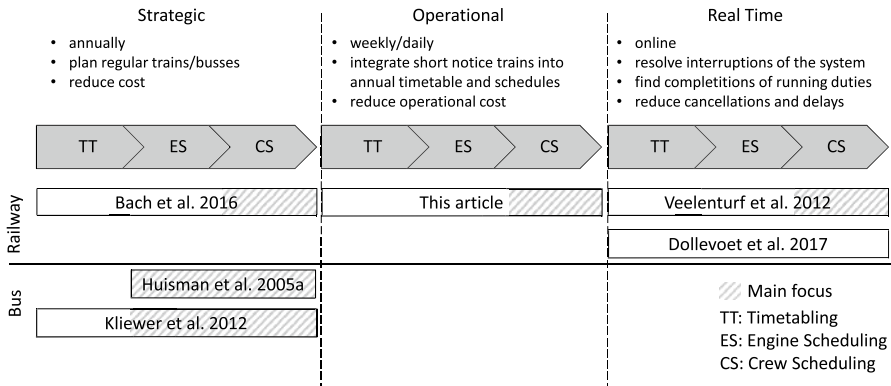


Fig. 1 Literature

2 Literature review

The resource planning process consists of timetabling, engine scheduling and crew scheduling problems. The process is executed strategically, operationally and in real time for different time horizons (Fig. 1).

At the strategic level, timetables, engine schedules and crew schedules are typically planned annually for trains that operate regularly. At the operational level, trains that must be planned on short notice are included in the annual schedules and timetables. Operational planning is conducted until about one week before execution. At real time level, the plans are adjusted to address disruptions during execution. Each planning level has its individual challenges. Models that work well for one level might be inappropriate for other levels.

The integration of the crew scheduling problem into other problems has been discussed in different streams of research, which we classify according to Fig. 1. We focus on crew scheduling models that allow to change the underlying timetable.

The first stream of research focuses on the strategic level and is divided into bus and railway systems. In suburban transit systems, the engine scheduling and the crew scheduling problem are addressed as a multiple depot vehicle and crew scheduling problem. Huisman et al. (2005a) propose a general mathematical formulation to address this problem that is based upon the single depot case of Freling et al. (2003). Kliwer et al. (2012) extend this model by implementing time windows for the scheduled trips and are able to solve instances of up to 661 trips in the single depot case for a bus system, resulting in significant savings. They implement the time windows by multiplying the corresponding trip arcs in the subproblem. In order to keep the graph small, they reduce the graph by deleting shifted arcs that do not enable additional connections. Kliwer et al. (2012) do not focus on operational cost reduction but on strategic resource planning. They use the time windows to reduce the number of necessary resources—buses and drivers.

In the railway industry, Bach et al. (2016) integrate the timetable with the crew scheduling problem, which is closely related to our research. They focus on the

strategic planning of the annual timetable. Based upon time windows for each customer demand, they create feasible timetables regarding the track capacity. In a second step, the crew scheduling problem may adjust the timetable if beneficial. Based upon free time slots for the tracks, they initiate discrete time stamps when customer demand can be fulfilled. However, the sequence of demand is predetermined and therefore the assignment of engines to trains is fixed. The considered time windows are rather large (6–9 h) and correspond on average to 7–10 possible starting times. They achieve significant savings considering both resources, tracks and train drivers. In contrast to Bach et al. (2016), we investigate to what degree coordinating the timetable with the crew schedules by small time shifts improves the overall performance in an operational environment. This focus implies that customers won't accept time shifts of several hours, since these would significantly impact their follow-up processes, for example production (compare Wieberneit 2007). Hence, operational planning must focus on small adjustments. To the best of our knowledge, the second stream of research about the operational coordination of the timetable and the crew schedule has not been addressed yet. We seek to close this gap.

The third stream of research focuses on real time railway rescheduling. Rescheduling deals with sudden disruptions in a running system and tries to find feasible solutions to further operate it with as few as possible interventions, such as delays or canceled trains. Veelenturf et al. (2012) seek to find feasible completions for already started duties which became infeasible due to disruptions. Their main objective is to find completions such that the train drivers return to their home depot while as few trips as possible are canceled or delayed. They use retiming to handle infeasibilities during the crew-rescheduling process. Infeasibilities might occur if not all trips can be covered by a train driver. Hence, their approach translates to exploring several timetables at once instead of a sequential trial and error process. In order to keep the computational effort small, they present two approaches to identify small subsets of critical trips that are allowed to be delayed if necessary. Other current research about rescheduling focuses on timetabling, rolling-stock and crew rescheduling exclusively. Dollevoet et al. (2017) present an iterative framework for real-time rescheduling that combines such approaches of Veelenturf et al. (2016), Nielsen et al. (2012) and Veelenturf et al. (2012). Although they use an iterative framework that is not based on integrated models, the overall performance is good and operationally applicable in disruptions management. In contrast to Veelenturf et al. (2012), we focus on the operational setting, that is, before any duty starts. Hence we do not seek to find feasible completions for already started duties but to coordinate the timetable proactively with the crew schedule in order to achieve higher productivities of the train drivers. This focus implies not only delays but also earlier departure times of trains.

We contribute by closing the gap between strategic planning and real-time operations management. In contrast to the strategic planning, we consider the specifics of an operational setting. At the strategic level, big changes of the timetable are allowed and help to coordinate the timetable, engine schedule and crew schedule for the annual plans. However, this coordination between the different resources is destroyed by the deviations of the annual schedule due to changes and the incorporation of short notice special trains into the system. The small time shifts by which we shift entire trains forwards or backwards in time increase the complexity of the

problem such that a different solution method is required to efficiently solve the crew scheduling problem in an operational setting. We present a column generation heuristic that is capable of solving the operational crew scheduling problem in an appropriate amount of time

In contrast to real-time planning, we optimize operational cost to increase profitability of the Freight Railway operators. At the real-time level, the focus is to find feasible completions for already started duties. Therefore, the transition from an interrupted to a stable system with as few as possible delays and cancellations is more important.

We contribute by introducing a mathematical model minimizing the operational cost by coordinating the timetable with the crew schedule in an operational setting, while keeping the engine schedule untouched. The time shifts in our model are small enough, such that customers and their follow-up processes are not affected. Furthermore, we present numerical results of three different real-world datasets, for which the timetable and the crew schedule were not coordinated at the strategic level. Our findings indicate that the coordination of the timetable and the crew schedule by small time shifts already results in significant idle time reductions of up to 27% and cost reductions of up to 9% depending on the specific dataset. Furthermore, the results suggest that the problem size can be reduced without losing much of the potential.

3 Problem description

Our research has been motivated by a planning problem that EFRO is facing. However, the planning processes and challenges are similar at other freight railway companies and the solution approach that we propose should be applicable also for them. At EFRO, train movements are split into to trips, which are pieces of work that must be performed by a single crew member. A sequence of trips being performed by one crew member is called a duty. The feasibility of a duty depends on working regulations such as maximum drive time, on operational requirements such as train related services and on the fact that a duty must start and end at the same depot (for a detailed description for regulations in Germany, see Jütte et al. 2011 p. 113). These requirements limit the crew scheduling flexibility such that idle times at relief points and dead-heading activities are often necessary to ensure that the duties are feasible.

During idle times, the train drivers wait for the next trip. Repositioning or dead-heading is necessary if there is no trip that can be assigned to the train drivers from their current positions. Dead-heading means that they sit as a ‘passenger’ in the train but do not drive it, whereas repositioning refers to other means of transportation such as buses, cabs or passenger trains.

The resource planning process at EFRO is sequential, and in each phase, only a local objective is optimized, which generally leads to suboptimal overall results, because dependencies between phases are not addressed. Consider, for instance, the crew scheduling example in Fig. 2: Assume that a crew member located at Station A is assigned to Trip 1. The crew member must return to Station A at the end of the duty. If trips cannot be moved, the crew member cannot be assigned to Trip 2, but must be assigned to

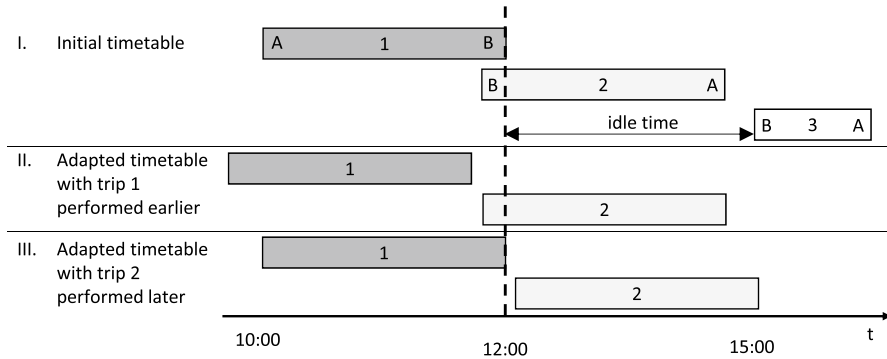


Fig. 2 Example of time shifts of the initial timetable

another trip, such as Trip 3. Such an assignment results in long idle time. If we could pull Trip 1 forward or postpone Trip 2, the crew member could be assigned to Trip 2 and the resulting duty would have little or no idle time.

These time shifts affect the underlying timetable and the utilization of the corresponding track infrastructure. Train operations and infrastructure management are activities that are typically conducted by different companies. EFRO, for instance, has to negotiate train schedules with the infrastructure provider. The negotiation process differs for long-term and short-term train planning (Caimi et al. 2017). For short-term planning, train operating companies may apply at all times for occasional traffic. In this process, spare track capacity is allocated to the train operating companies. However, the closer the planning gets to execution, the fewer options for track allocation are typically available, in particular for highly utilized tracks.

EFRO operates in Germany, where track utilization differs substantially between regions. Some regions in the west and southwest are highly utilized, but the majority of regions has slack capacity (see Sect. 6.1). In our analyses, we focus on regions that have slack capacity and assume that train schedules can be shifted by up to 15 minutes without violating capacity constraints. In situations where track capacity restrictions do not allow such shifts, we can constrain the model to trains that can be shifted.

The problem of coordinating the timetable and the crew schedule at an operational level then becomes to simultaneously find a solution at minimal cost that defines the departure and arrival times of each train while ensuring that each trip is covered by at least one train driver. Furthermore, the crew schedule must base upon the adjusted timetable to ensure operational feasibility between both.

4 Mathematical model

In this section, we formulate our Crew Scheduling Problem with Semi-flexible Timetables. First, we present the standard formulation of the crew scheduling problem and then extend this model to incorporate the time shifts for the trains. We apply a column generation approach to solve our model, which we present in Sect. 5.

We denote the set of duties by D , indexed by d . The cost of duty d is c_d and is determined in the subproblem. It contains all real cost associated with the duty such as wages for the train driver or fares for deadheads and artificial cost to penalize undesired properties of the duty, such as number of train changes. The set of trips is denoted by T , indexed by t . The parameter a_{td} defines whether duty d contains trip t ($a_{td} = 1$). x_d is a binary decision variable that indicates whether a duty d is selected for the schedule or not.

The binary master problem can be formulated as a set covering problem:

$$\min \sum_{d \in D} c_d x_d \tag{1}$$

$$\sum_{d \in D} a_{td} x_d \geq 1 \quad \forall t \in T \tag{2}$$

$$x_d \in \{0, 1\} \quad \forall d \in D \tag{3}$$

The objective function (1) minimizes the cost of the crew schedule, while constraints (2) ensure that all trips are covered by at least one duty.

In our model, we allow to shift entire trains forwards and backwards in time. Therefore, we extend the master problem by the decision of whether a train should be shifted or not. In addition to the notation of the standard formulation, we introduce a set of trains as K indexed by k . Each train k is allowed to have a set of time shifts Δ , indexed by δ . Furthermore, let $T(k)$ represent the set of trips that belong to train k . Since entire trains are shifted, each trip $t \in T(k)$ has the same time shifts $\delta \in \Delta$ as train k . Let $a_{td\delta}$ be 1, if duty d contains trip t with the time shift δ and 0 otherwise. Similarly, let \hat{a}_{tk} be 1 if trip t belongs to train k and 0 otherwise. Note that these coefficients are predefined or generated in the subproblem. Finally, let $y_{k\delta}$ be the binary decision variable defining whether train k is shifted by δ .

The Crew Scheduling Problem with Semi-flexible Timetables can be formulated as (4)– (9):

$$\min \sum_{d \in D} c_d x_d \tag{4}$$

$$\sum_{d \in D} \sum_{\delta \in \Delta} a_{td\delta} x_d \geq 1 \quad \forall t \in T \tag{5}$$

$$\sum_{d \in D} \sum_{t \in T} a_{td\delta} \hat{a}_{tk} x_d \leq M_k y_{k\delta} \quad \forall k \in K, \forall \delta \in \Delta \tag{6}$$

$$\sum_{\delta \in \Delta} y_{k\delta} = 1 \quad \forall k \in K \tag{7}$$

$$x_d \in \{0, 1\} \quad \forall d \in D \quad (8)$$

$$y_{k\delta} \in \{0, 1\} \quad \forall \delta \in \Delta, \forall k \in K \quad (9)$$

The objective function (4) remains the same. The trip covering constraints (5) are adapted, such that each trip t or one of its copies is covered by at least one duty. Constraints (6) enforce that a duty d can only be selected if each of its trains has the same time shift as the selected time shift δ for each particular train k . We set $M_k = 2|T(k)| \forall k$, that is, twice the number of trips performed by train k . We use $2|T(k)|$ in order to allow multiple train drivers to cover the same trip.¹ Choosing M_k as small as possible is important during the solution process, on which we will focus in Sect. 5. Constraints (7) ensure that each train has only one departure time. Hence in combination, Constraints (6) and (7) force the model to select for each train that time shift which results in the lowest cost for the crew schedule. Constraints (8) and (9) are the binary constraints for the decision variables.

5 Solution method

We apply a column generation approach to solve the Crew Scheduling Problem with Semi-flexible Timetables. The linearized problem can be solved efficiently, but finding an integer solution for the train variables $y_{k\delta}$ is challenging, since the time shifts increase the problem size. The existence of many close to optimal solutions complicates finding the optimal solution. The reasons are twofold: Firstly, many trains improve the solution when they are shifted. However, these improvements are often small for most of the trains (see Sect. 6.4). Secondly, when allowing multiple time shifts, the differences in the departure times are small and often result in only minor additional savings after a certain threshold, that is, a train is shifted such that a train switching option for the train driver is enabled (see Sect. 6.3).

We next present the general solution algorithm (Sect. 5.1), before providing more details to the subproblem (Sect. 5.2) and finally developing an efficient approach for generating integer solutions (Sect. 5.3).

5.1 General algorithm

The general algorithm (see Fig. 3) consists of six steps that can be grouped into three major phases: duty generation, timetable construction and duty fixation. The algorithm is taken from Jütte et al. (2011) and was adjusted by Phase 2.

The first phase is the 'standard' column generation scheme, with its initialization (step 1) and iteration between the subproblem (step 2) and the restricted linear

¹ Usually, more than two train drivers are allowed to cover the same trip. Technically, this formulation allows on average only two train drivers to cover the same trip. Test instances with $M_k = 4|T(k)|$ showed no significant changes regarding the over-coverage but the MIP performed slightly better for $M_k = 2|T(k)|$.

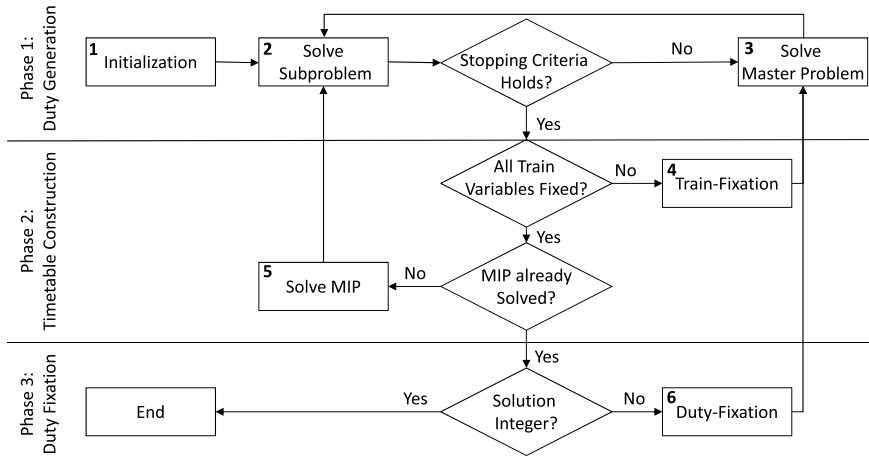


Fig. 3 Algorithm

master problem (step 3) with relaxed decision variables $x_d, y_{k\delta}$. In step 1, we insert dummy duties for the original trips with high cost to initialize the master problem. These dummy duties are not incorporated in Constraints (6), such that our model is always feasible. We go to the next phase, when no additional duties with negative reduced cost are found or when the improvement of the objective value is marginal over the previous iterations.

The second phase derives an integer timetable from the priorly generated duties. First, we heuristically fix the train variables $y_{k\delta}$ (step 4) while generating new duties. When all train variables are fixed, we remove the fixation constraints and solve the model *once* with binary $y_{k\delta}$ as a MIP (step 5).

In the third phase, an integer solution for duty variables x_d is heuristically derived (step 6), such that the timetable and the crew schedule are coordinated.

5.2 Solving the subproblem

The subproblem generates feasible duties concerning labor and operational requirements and adds them as variables to the master problem. We use a time-space network, in which each node represents a specific time and location and each arc represents a specific set of tasks with the corresponding cost and resource consumptions. For each depot, where a duty can start, we insert a start and an end node. The inserted node is time wise way before or after all possible trips that could be covered during the week from that particular depot. Besides dead-heading and repositioning, arcs also represent trips with the necessary activities to drive this particular trip, including among others walking and preparation tasks.

The Crew Scheduling Problem with Semi-flexible Timetables considers different departure times for each train. The subproblem represents these different departure times by copies of 'trip arcs' (compare with Mercier and Soumis 2007).

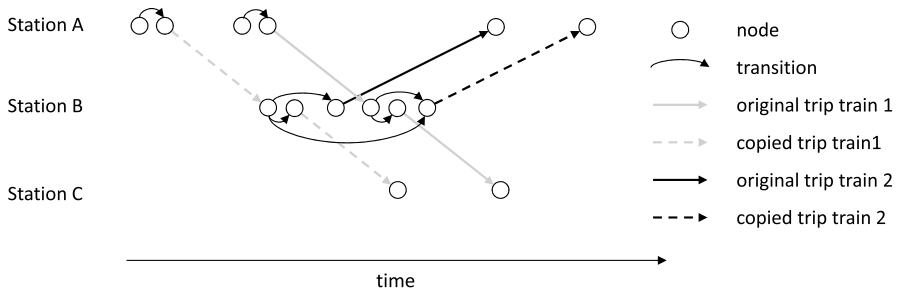


Fig. 4 Unconsolidated network

For the same train, only trip arcs with the same time shift δ are connected. Connecting arcs with different time shifts of the same train would result in infeasible duties in the master problem, hence by leaving them out we can reduce the graph. Please note that two non-consecutive arcs of the same train with different time shifts could be part of the same path through the network, resulting in an infeasible duty. These duties appear rarely and are automatically excluded from the final solution through the heuristic train fixation (Step 4) and solving the MIP (Step 5).

Figure 4 shows as an example the time-space network for three original trips of two different trains with a single time shift each. The continuous lines represent the original trips. Train 1 has two trips, driving from station A to station C via station B. Train 2 has only one trip going from station B to station A. Considering only the original trips, there is no train switching option for the train driver at station B because the first trip of train 1 arrives after the departure of train 2. Hence there is no transition arc between the two nodes. The dashed lines represent copied trips arcs. For simplicity, we consider only one time shift for each train. Train 1 is shifted backwards, whereas train 2 is shifted forward in time. These time shifts enable several transitions between the two trains. If only train 2 is shifted, then there exists one transition from the original trip of train 1 to the shifted trip of train 2. Additionally if train 1 is shifted as well, then there exist transitions to the original trip as well as to the shifted trip of train 2.

We consolidate trip arcs and transitions in order to reduce the network size. We combine trip arcs with associated transition arcs to a single arc, as it is shown in the network of Fig. 5.

The problem size of the subproblem and the master problem increases significantly by allowing additional time shifts for each train, that is, by adding copies of trips to the graph. Especially for quasi-continuous time windows with a resolution of one minute, the problem becomes intractable for any large time window. We discuss the impact of additional time shifts in Sect. 6 and show that for our real-world test cases a quasi-continuous resolution is not necessary.

We next show how we solve the subproblem to generate new variables (duties) for the restricted linearized master problem. In the subproblem, duties are represented as paths in a time-space network. We have to find those feasible paths

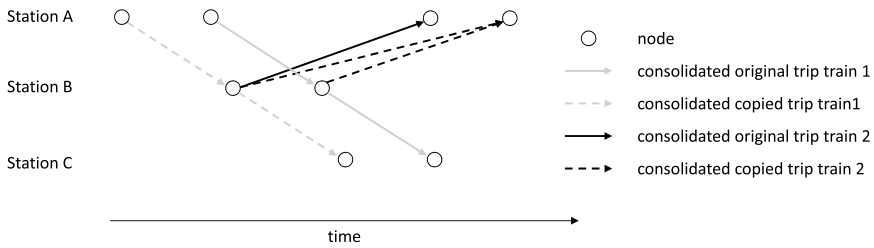


Fig. 5 Consolidated network

through the network that have the potential to reduce the objective function of the master problem.

Let $G = (V, E)$ be a directed graph with a set of nodes V and a set of edges E . We denote the subset of edges that belong to duty d by $E(d)$ and the subset of edges that represent trips in G by $E(T)$. Let $\pi_t, \sigma_{k\delta}$ and φ_k be the dual variables of Constraints (5)–(7), respectively. Then the reduced cost for duties can be written as

$$\begin{aligned} \tilde{c}_d = c_d & - \left[\sum_{t \in T} \sum_{\delta \in \Delta} a_{td\delta} \pi_t \right. \\ & \left. - \sum_{k \in K} \sum_{\delta \in \Delta} \sum_{t \in T} a_{td\delta} \hat{a}_{tk} \sigma_{k\delta} + 0 \cdot \varphi_k \right] \quad \forall d \in D, \end{aligned} \tag{10}$$

where $c_d = \sum_{e \in E(d)} c_e$ is the cost for all tasks that are performed in duty d . If $\tilde{c}_d < 0$, duties eventually improve the objective function of the master problem. Hence we search for shortest paths with negative reduced cost that comply with our scarce resources, such as maximum driving time. We use the permanent labeling algorithm of Desrochers and Soumis (1988) to solve this resource constrained shortest path problem.

For initializing the labeling algorithm, we update the cost for each arc that represents a trip to

$$\tilde{c}_e = c_e - (\pi_e - \sigma_e) \quad \forall e \in E(T) \tag{11}$$

where π_e is the dual value of the corresponding trip and σ_e is the dual value of the corresponding train with the time shift of that edge. The labeling algorithm begins at the start nodes of the depots. We then iterate through the network and set labels at each node to keep track of the consumed resources. At each node, we reduce memory consumption by eliminating dominated labels. Finally, we add the duties with the most negative reduced cost to the master problem. The exact number of added duties is dynamically defined and depends on the status of the optimization. Basically at the beginning, when many duties with negative reduced cost exist, we limit

the number of duties relatively strongly, whereas at the end all duties with negative reduced cost are added.

5.3 Timetable construction and duty fixation

During Phase 1 of the general algorithm, we generate duties as long as we do not meet one of the termination criteria. If one of the criteria is met, we go into Phase 2, where we derive a timetable that is coordinated with the crew schedule. The decision variables are relaxed, and we aim to find a binary solution for the timetable. Fixing a train variable $y_{k\delta}$ to a binary value means that a subset of duties becomes obsolete, since they violate Constraints (6). This is equivalent to a significant reduction in the solution space. Therefore, an optimal solution might require a different set of duties than those that have been generated. To compensate the reduction in the solution space, we fix the train variables step by step and continuously generate new duties. In terms of the optimality gap, the following heuristic performs poorly for any larger number of time shifts $|\Delta|$. However, our goal is not to find a good solution in this step of the algorithm but to generate additional duties.

Our fixation heuristic sorts all train variables that have not been fixed yet in ascending order by value. In each iteration, we fix a certain number of train variables to zero by adding an additional constraint and repeat until all train variables are fixed. We ensure that for every train one train variable is not fixed to zero such that the remaining $|K|$ variables correspond to the heuristically determined best coordinated timetable. By fixing train variables to zero instead of equal to one, we moderately reduce the solution space in each iteration and give the algorithm the opportunity to generate new duties.

In the next step, we improve the integer solution. Therefore, we remove all fixation constraints from the model added during the fixation and change the mathematical model to a mixed binary linear model, where only the train variables $y_{k\delta}$ are binary. We solve this model only once using Gurobi MIP solver, accepting an optimality gap of 0.1% or a maximum time. After fixing the train variables, our model corresponds to a normal crew scheduling problem with relaxed decision variables x_d .

Once we have an integer timetable, we go to Phase 3, in which we determine an integer crew schedule. We do so by fixing duty variables that are close to one while iterating between the subproblem and the master problem until we have a feasible model with integer values only. This heuristic normally performs well with gaps below 0.2%.

6 Numerical results

Adding time shifts to the crew scheduling problem raises four questions in an operational setting. Firstly, by how many minutes should we shift the trains? If the time shifts are too small, the additional flexibility might be insufficient to enable more train switching options for the train drivers. On the other hand, if the time shifts are

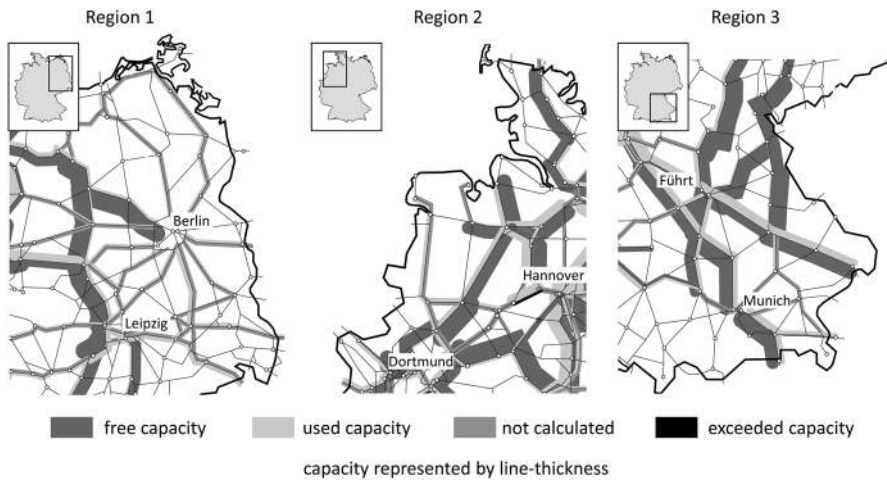


Fig. 6 Track utilization of the datasets (Umweltbundesamt 2010, p. 57)

too large, they might interfere with the engine schedule due to limited engine idle times. At EFRO, time shifts of less than or equal to 15 min can normally be implemented without interfering with the engine schedule.

Secondly, how many time shifts per train should we offer? Allowing only few alternative departure times limits the flexibility, whereas many alternative time shifts result in high complexity and long computation times.

Thirdly, how many trains do we have to shift to achieve significant savings? The best found solution might shift many trains. However, shifting a train means some effort for the planners, which is not considered in our model. Thus, it might be advisable to shift only some trains from a management standpoint. This question is of high relevance in operational environments.

Lastly, the small time shifts might impact the robustness of the duties, since we aim at reducing the idle time. Are our results still valid if we require buffer times when train drivers are changing trains?

To answer these four questions and to quantify the impact, we test our Crew Scheduling Problem with Semi-flexible Timetables on three real-world datasets.

6.1 Datasets and benchmark

The datasets are from the northeastern, the northwestern and the southeastern regions of Germany as shown in Fig. 6. The thickness of the tracks refers to the theoretical capacity in trains per day. All examined regions are characterized by a low track utilization of only 30–70%, and only one track between Hannover and Minden in Dataset 2 is overutilized. The overutilization is possible because the minimum time between trains was reduced to temporarily enlarge the capacity. We exclude trains running on this track from the optimization to ensure the feasibility of the track capacity. Table 1 provides additional information about the numbers of trips, trains and stations in the datasets. For all datasets, the timetable and the crew

Table 1 Dataset description

	Dataset		
	1	2	3
#trips	459	781	1281
#trains	103	149	339
#stations	55	36	95

schedule were not coordinated at the strategic level and the planning period covers one week.

We benchmark the Crew Scheduling Problem with Semi-flexible Timetables against a crew scheduling problem that does not allow time shifts. We solve the benchmark problem using the column generation heuristic described in Fig. 3 excluding the steps four and five. The solutions have an optimality gap of less than 0.2% based upon the root relaxation of the column generation scheme. Important performance indicators of the *planned* crew schedule are the total cost, the idle time, the total work time and the number of duties of the schedule. We normalize the benchmark equal to 100 for each dataset and each indicator, such that the impact of our model can be compared between the three datasets.

6.2 Maximum time shifts

We analyze the effect of the time shift length δ on the solution. We limit the number of time shifts $|\Delta|$ per train to three and increase the maximum time shift $\delta_{max} = \max(|\delta| \in \Delta)$ by multiples of 5 minutes from 5 to 30 min. For a train that originally departs at 12:00 and a time shift of $\delta_{max} = 5$ minutes, for instance the train can depart at 11:55, 12:00 or 12:05 and for $\delta_{max} = 10$ minutes the train can depart at 11:50, 12:00 or 12:10. At EFRO, time shifts of less than or equal to 15 minutes can normally be implemented without interfering with the engine schedule.

All presented results can be found in Table 5 in the Appendix and are based upon averages with at least six samples since the heuristic is not deterministic due to the parallel subproblem. Figure 7 shows how the cost depends on the time shift δ_{max} . The coordination between the timetable and the crew schedule reduces the total cost of the crew schedule. The savings increase in the length of the time shifts, but the slope is decreasing. An exception is Dataset 2, where the cost slightly increase for $\delta_{max} \geq 20$ since no additional connections are enabled for longer time shifts.

The savings are dependent on the specific dataset. For the first and second dataset, the savings for $\delta_{max} = 15$ are 4.5% and 5.6%, respectively, whereas the savings for the third dataset are 9.0%. The reasons why the savings are significantly larger in Dataset 3 are threefold. Firstly, without considering time shifts, each trip in Dataset 3 has on average less potential connections compared to the other datasets. Secondly, the times between the trips and the potential connections in Dataset 3 are longer; hence, the idle times for the train drivers are higher in the benchmarking case. Thirdly, time shifts increase the number of potential connections in Dataset 3 more strongly compared to the other datasets. Therefore, in total more connections

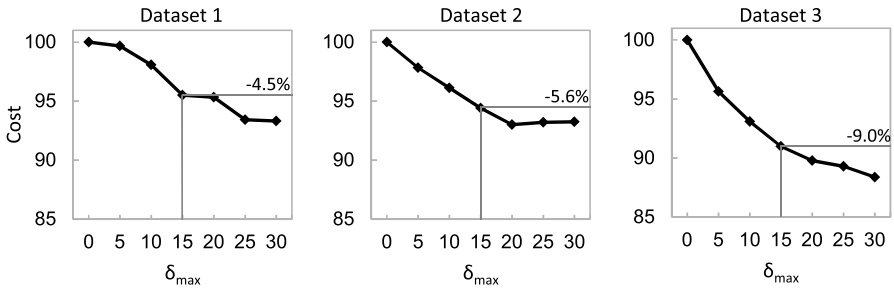


Fig. 7 Cost for increasing δ_{max}

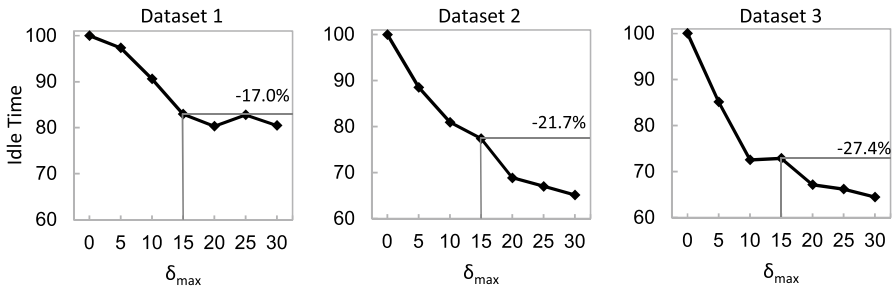


Fig. 8 Idle time for increasing δ_{max}

are enabled and the idle time reduction is higher compared to Datasets 1 and 2 since the original timetable was less coordinated with the crew schedule.

In general, the savings are caused by more efficient duties. Whereas the average length of a duty remains stable for increasing δ_{max} , the number of duties per schedule decreases. Figure 10 shows how the number of duties depends on the time shift δ_{max} . In all datasets, the number of duties reduces significantly. For $\delta_{max} = 15$, the reductions are 4.9%, 3.4% and 7.0%, respectively. Hence, the time shifts do not simply shorten the duty length, but the Crew Scheduling Problem with Semi-flexible Timetables constructs the duties differently than the standard crew scheduling problem could.

Different factors are responsible for the savings and an important one are the idle time reductions. The idle times are presented in Fig. 8. The idle time savings for $\delta_{max} = 15$ are between 17.0% and 27.4% and explain 56.5% to 72.5% of the cost savings.

The shape of the idle time is similar to the graph of the total cost; however, it is steeper and not monotonously decreasing. In Datasets 1 and 2, the shapes deviate for $\delta_{max} \geq 20$ min. In Dataset 1, the cost decreases, whereas the idle time stays stable. In Dataset 2, the opposite effect occurs: The cost remains stable whereas the idle time decreases. This underlines that the cost reduction is also driven by other factors than idle time. Figure 9 shows the total work time and supports this statement: The shape of the cost and the total work time are perfectly aligned. Thus, other activities

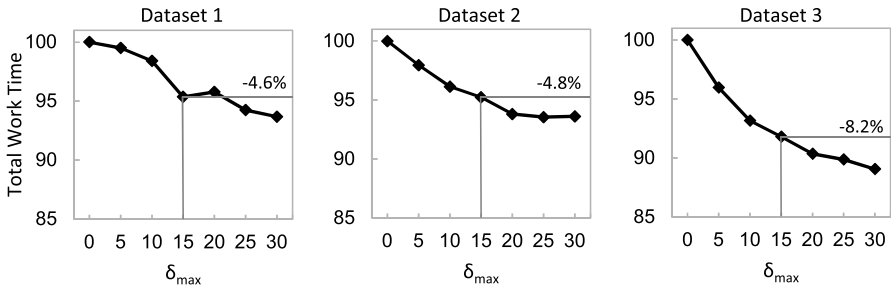


Fig. 9 Total work time for increasing δ_{max}

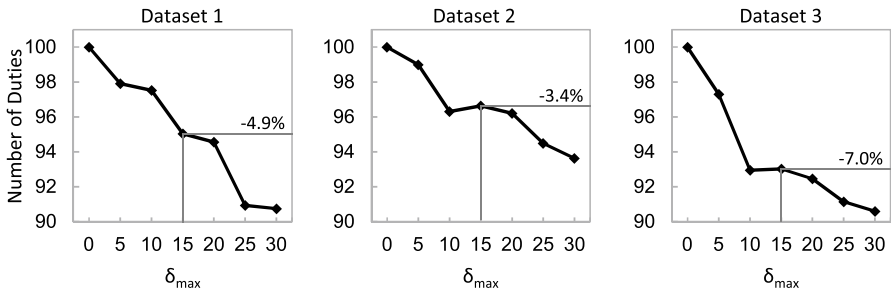


Fig. 10 Number of duties for increasing δ_{max}

must have impacted the work time. A more detailed analysis revealed that the model exchanged dead-heading and repositioning for idle time. Directly speaking, in Dataset 2 the train drivers spend the time in passenger trains or cabs instead of waiting in the stations for the next train to drive. However, the total work time and thus the cost do not change.

Our results suggest that the savings increase in δ_{max} . This is reasonable, since larger time shifts enable more additional connections. Hence, within the restriction of small time shifts, the largest feasible time shifts should be chosen in order to achieve the largest savings.

6.3 Multiple time shifts

Next we investigate the impact of additional time shifts, since the limitation to three possible time shifts is a restriction. We set $\delta_{max} = 15$ and increase the number of time shifts $|\Delta|$ from three to seven. This results in the following sets: $\Delta_3 = \{-15, 0, 15\}$, $\Delta_5 = \{-15, -7, 0, 7, 15\}$ and $\Delta_7 = \{-15, -10, -5, 0, 5, 10, 15\}$. For a train that originally departs at 12:00 and Δ_5 , the train can depart at 11:45, 11:53, 12:00, 12:07 and 12:15. All results are shown in Table 5 in the appendix.

Figure 11 shows the cost for our datasets. The cost changes only marginally in $|\Delta|$. On average, the cost reduces by 0.1 percent points for every additional

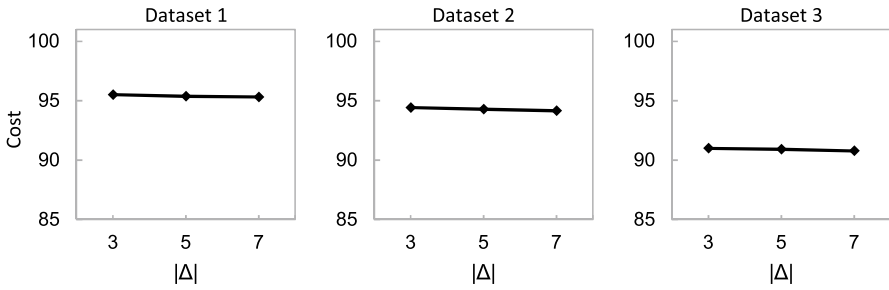


Fig. 11 Cost for increasing |Δ|

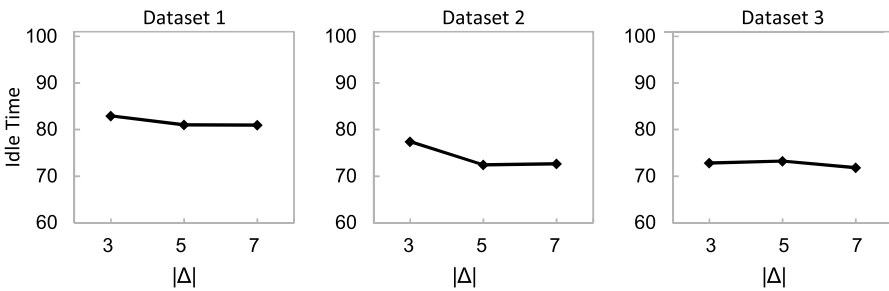


Fig. 12 Idle time for increasing |Δ|

two time shifts. The idle times for all three datasets vary (Fig. 12); however, they have no impact on the total work time (Fig. 13). This effect is explained by exchanging idle time with deadheading and repositioning as mentioned in the previous subsection. Similarly, the total work time and the number of duties remain stable for increasing |Δ| (see Figs. 13 and 14).

The results suggest that the cost reduction is driven primarily by the maximum time shift δ_{max} and can only be slightly improved by additional time shifts |Δ|. The larger δ_{max} , the larger is the potential time window and the more additional connections exist for enabling train switching processes. However, additional time shifts $|\delta| < \delta_{max}$ do not enable considerably more new switching options, but only if several trains are closely interconnected. If the schedule is sufficiently loose, then there exist buffer times between the trips of a duty and additional time shifts are irrelevant. If the schedule is tight, then additional time shifts are relevant and enable connecting several trains together. However, this is not the case in our datasets within a low utilized network, and hence, the savings are marginal. Besides closely interconnected trains, the additional reductions can be partly explained by tighter schedules. For instance, if a time shift of 5 min is enough, the additional savings compared to δ_{max} are then up to 10 min. However, these savings are rather small.

In an operational environment, this observation allows us to reduce the problem size significantly by considering only three time shifts in trade-off for slightly worse solutions. This is of special importance, since it directly influences the computation

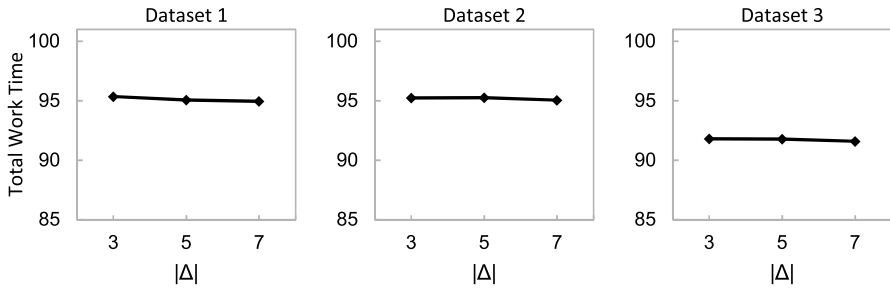


Fig. 13 Working time for increasing $|\Delta|$

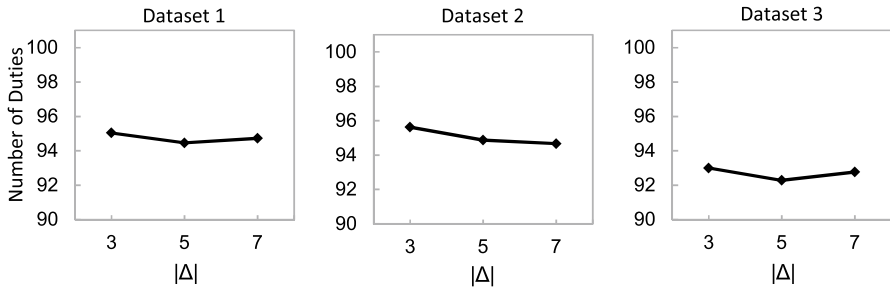


Fig. 14 Number of duties for increasing $|\Delta|$

time. Furthermore, it shows that models with continuous time windows most likely will not or not significantly improve the solution.

6.4 Limiting the number of shifted trains

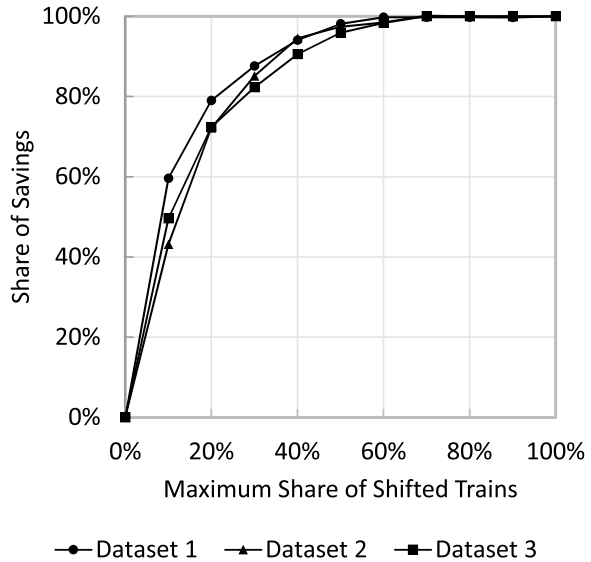
So far we have not limited the number of shifted trains. However, from an operational perspective it is important to shift only trains that contribute sufficiently large savings, since some effort is needed to shift a train.

Quantifying the cost of shifting trains is very difficult. In Bach et al. (2016), the cost is determined by the cost difference for the tracks, based upon the different times. For example during the day, the cost is higher than during the night. We consider small time shifts with negligible effect on track cost. However, additional effort might occur for planners, which has to be considered. We consider the additional effort by a management decision of how many trains should be shifted at most. Therefore, we extend our model (4)–(9) by

$$\sum_{\delta \in \Delta \setminus \{0\}} y_{k\delta} \leq \alpha |K| \tag{12}$$

where $\alpha \in [0, 1]$ is the share of trains that can be shifted at most.

Fig. 15 Share of maximum savings for increasing α



To analyze the effect of the number of shifted trains on savings, we restrict α from 0% to 100%. All results are shown in Table 6 in the Appendix. Figure 15 shows the results for $|\Delta| = 3$ and $\delta_{max} = 15$ for our datasets. The y-axis shows the share of the potential savings, relative to the savings if all trains could be shifted ($\alpha = 1$). The case $\alpha = 0$ does not allow to shift any trains and corresponds to the standard crew scheduling problem. The graph suggests that 43% - 60% of the savings can be gained by shifting only 10% of all trains. When shifting 20% of the trains, 72–79% of the potential savings can be gained. Almost all savings are gained by shifting half of the trains. This indicates that there are many train shifts adding only negligible savings. This observation increases the operational implementability of our Crew Scheduling Problem with Semi-flexible Timetables, since the effort can be reduced to a reasonable level while still achieving considerably large savings.

6.5 Robustness

Duty robustness plays a major role during operations. The literature on crew scheduling identifies train changes as a major source for delay propagation (Heil et al. 2019). When shifting trains by small amounts of time, we potentially enable very tight connections for the train drivers that change trains, resulting in less robust duties. The literature proposes to penalize the number of train changes (e.g. Alfieri et al. 2007; Freling et al. 2000) or to ensure minimal transfer or buffer times, whenever a train driver changes trains (e.g. Kroon and Fischetti 2000; Jütte et al. 2011).

We increase the buffer times by multiples of 15 min from 0 to 45 min and analyze the impact of the buffer times on the important performance indicators. The case of zero minutes buffer time corresponds to the results in the previous sections. Table 2 shows the results for $|\Delta| = 3$ and $\delta_{max} = 15$ for our datasets. All results are

Table 2 Results with buffer times

Dataset	δ_{max}	$ \Delta $	Buffer time	Cost	Idle time	Work time	#Duties
1	15	3	0	95.51	82.95	95.35	95.04
			15	94.98	78.51	94.52	93.38
			30	93.53	88.99	94.15	91.25
			45	94.40	89.08	94.56	90.88
2	15	3	0	94.42	77.43	95.24	96.63
			15	94.73	82.20	94.72	97.47
			30	93.21	80.77	94.11	94.68
			45	94.12	75.86	94.29	94.13
3	15	3	0	91.00	72.92	91.80	93.02
			15	88.07	74.13	92.46	91.54
			30	93.41	82.07	94.70	92.58
			45	93.09	76.42	93.62	94.94

benchmarked against a crew scheduling model that does not allow time shifts but requires the same amount of buffer times. All performance indicators vary slightly around the results with a buffer time of zero minutes but do not change the main findings of the previous sections.

Additionally, we analyzed the average number of train changes within one duty for $|\Delta| = 3$ and $\delta_{max} = 15$ without buffer times for all datasets. The average number of train changes increased from 1.50 to 1.53 for Dataset 1, from 2.07 to 2.13 for Dataset 2, and from 2.18 to 2.33 for Dataset 3. It is difficult to judge whether these increases result in less robust duties. We believe that the changes are rather small and that these small increases could be easily compensated by increased buffer times such that the duties in our crew schedules are similar robust as the duties in the crew schedules of the benchmark.

6.6 Computational complexity

We solved the model with a column generation-based heuristic. The computational complexity increases with the number of trips $|T|$, the number of trains $|K|$ and the number of possible time shifts $|\Delta|$. In this section, we first focus on the detailed computation times for the different phases of the algorithm. Next we analyze the impact of the time limit for the MIP solver, and finally, we investigate the impact of the small time shifts on the computational complexity. The tests were performed on a server with Intel Xeon E5-1650 processor and 8 CPUs with 32GB RAM. The sub-problem and the MIP are parallelized.

6.7 Detailed computation times

Table 3 shows the detailed computation times for the three datasets with $\delta_{max} = 15$ and $|\Delta| = 3, 5$ and 7 . We report the computation times for the root relaxation (Root),

Table 3 Detailed computation times

Dataset	$ T $	$ K $	$ \Delta $	Root [s]	TFH [s]	MIP [s]	DFH [s]	total [s]
1	459	103	3	170	73	26	32	300
			5	272	307	454	151	1184
			7	881	456	873	561	2770
2	781	149	3	1571	1306	960	16	3853
			5	4159	6426	7296	75	17,956
			7	15,998	9673	17,717	142	43,531
3	1281	339	3	1285	1186	4049	34	6554
			5	5914	4003	25,951	138	36,005
			7	15,597	9509	21,155	141	46,402

train fixation heuristic (TFH), MIP solver (MIP) and duty fixation heuristic (DFH). Several observations can be made:

Firstly, the total computation time increases with the number of trains $|K|$ and with the number of trips $|T|$ since both measures are closely related.

Secondly, the number of time shifts $|\Delta|$ has a large effect on the computation time. Adding two time shifts increases the computation time significantly. The reasons are the larger graph and therefore slower subproblem and a larger number of binary variables in the MIP.

Thirdly, a closer look at the detailed computation times reveals that the second dataset requires more time to reach an heuristic integer timetable than the third dataset although it is smaller in terms of trains and trips. One explanation might be that in Dataset 2 even without additional time shifts each trip has already on average more potential connections compared to Dataset 3. Hence, the time space network is more complex although Dataset 2 is smaller. In contrast the MIP computation time for Dataset 3 is longer compared to Dataset 2 since more binary variables exist such that the total computation time is still longer.

6.8 Impact of MIP time limit

For large problem sizes with many trains and time shifts, finding the optimal solution or achieving a target integrality gap of the MIP is time-consuming. To analyze the trade-off between computation time and optimality gap, we varied the maximum computation time for solving the MIP between 20 and 60,000 s (step size 10 for 20–50, step size 100 for 100–600, step size 1000 for 1000–7000, step size 10,000 for 10,000–60,000).

Figure 16 shows the cost and the total computation time for the different optimization runs. The larger the MIP time limit, the larger the total computation time. Furthermore, the lower bound and the average cost from Sect. 6.3 are included. As a lower bound, we use the column generation root relaxation. The results for all three datasets refer to $\delta_{max} = 15$ and $|\Delta| = 3, 5$ and 7 . Please note the different scales of the x-axis. Several observations can be made:

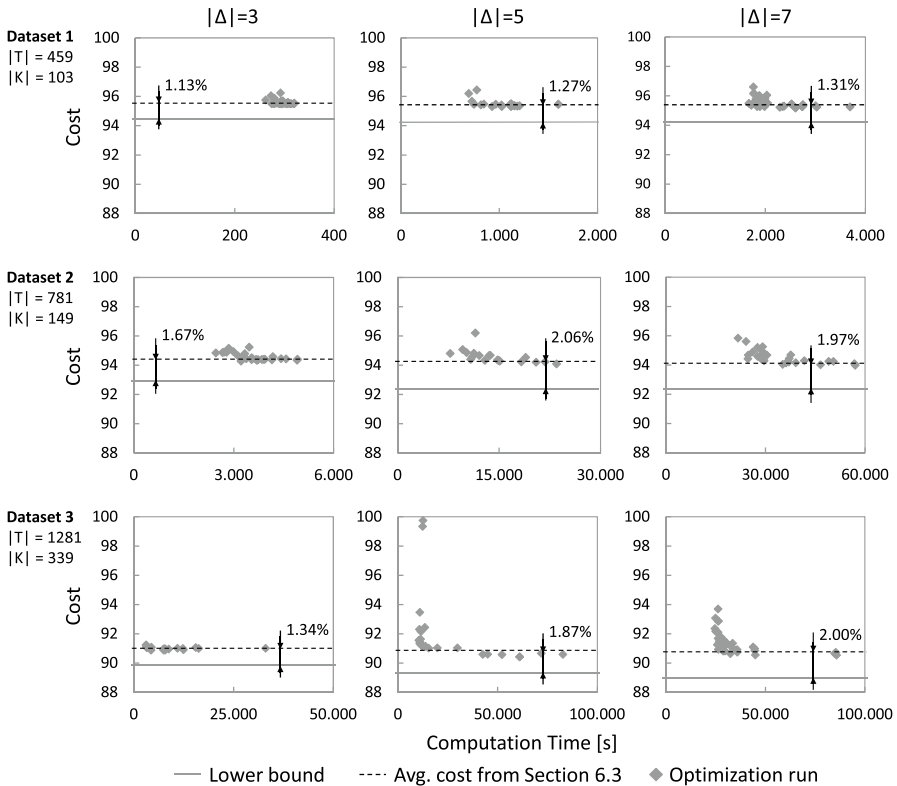


Fig. 16 Computation times for $\delta_{max} = 15$

Firstly, the MIP time limit plays a smaller role for the small test instances, since the solver very quickly returns solutions meeting the target integrality gap of the MIP (e.g. on average 26 seconds for Dataset 1 and $|\Delta| = 3$).

Secondly, the MIP time limit has a large impact on the integer solutions and on the resulting optimality gaps for the large test instances. Obviously, if the time limit is too short, the MIP solver does not have sufficient time to find good integer solutions with low gaps (e.g. see Fig. 16 for Dataset 3 and $|\Delta| = 5$). With increasing time limit, the solver quickly finds good solutions such that the integer solutions converge to a certain integer solution limit. Once this limit is reached, the MIP integrality gap decreases without significantly improving the integer solutions due to a slowly increasing lower bound for the MIP. However, this process does not decrease the root relaxation lower bound further and therefore shows the limitations of our algorithm.

Lastly, the necessary time limit to converge to the integer solution limit increases with the number of time shifts. For example in Dataset 3, the integer solution converges quickly for $|\Delta| = 3$, whereas for $|\Delta| = 5$ or 7 it takes significantly longer.

To conclude, we know from Section 6.3 that additional time shifts only result in marginal savings. This observation could be used to reduce the computation time significantly without losing much of the potential savings.

Table 4 Performance Comparison

Dataset	Δ	This paper		Bach et al. (2016)*	
		GAP [%]	CT [s]	GAP [%]	CT [s]
1	3	1.11	317	2.20	8961
	5	1.24	1196	1.76	24,306
	7	1.20	2899	2.54	96,413
2	3	1.67	3853	2.30	100,864
	5	2.06	17,956	3.09	483,706
	7	1.97	43,531	†	†
3	3	1.34	6554	1.97	259,011
	5	1.87	36,005	†	†
	7	2.00	46,402	†	†

† Exceeded maximum computation time of 144 h

*We implemented Bach et al.’s branch and price heuristic for the timetable variables including their branching scheme. Additionally, we adapted our subproblems, such that at each node in the search tree we solve the subproblems only for those days that are relevant for the variable that we are branching on

6.9 Impact of small time shifts

To show the effect of small time shifts in the operational setting on the computational complexity, we compare our solution approach to Bach et al.’s (2016) branch and price approach for the strategic setting. Table 4 shows the achieved gaps and computation times of our implementation of Bach et al.’s and our algorithms. Our proposed algorithm achieves gaps that are by a factor of 1.38–2.12 smaller in significantly shorter time.

The reason for this performance difference is that a branch and price approach requires solving the subproblems very frequently in the branching tree. However, solving the subproblem uses most of the computation time in our algorithm that is not spent for solving the MIP, e.g. 96.8% of the time for Dataset 3 with $|\Delta| = 7$ such that a branch and price is not a suitable approach for achieving integer solutions. In the operational setting, the subproblem is computationally more expensive because the complexity of time-space-networks is more prone to small time shifts compared to larger time shifts. The reason is that if a train is shifted by several hours, then the corresponding copied trips will be connected to different trips in the time-space-network than the original trip. In contrast, if a train is shifted only by a few minutes, then the original trip and the copied trip will have many connections in common. Furthermore, the connected trips have copies on their own, such that the number of possible connections increases very quickly. Therefore, branch and price approaches are not well applicable and we developed a solution method that behaves better with larger subproblems.

7 Conclusion

In this paper, we analyzed how the timetable and the crew schedule can be coordinated in an operational setting. We added flexibility to the timetable by allowing to shift entire trains by some minutes. We presented an adapted formulation of the

crew scheduling problem and introduced a column generation approach to solve this problem. In particular, we focused on deriving good integer solutions in a reasonable amount of time. The performance of the model was tested on three datasets from EFRO and benchmarked against the standard crew scheduling problem. The results indicate that significant reductions can be achieved by coordinating the timetable and the crew schedule. The specific reductions depend on the dataset and the maximum amount of time by which the trains are allowed to be shifted. Allowing 15 minutes time shifts reduces total cost between 4.5% and 9.0%. These savings were primarily achieved by reducing the idle time of the train drivers through additional train switching options. The idle time was reduced between 17.0% and 27.4%. Furthermore, we analyzed whether multiple time shifts further reduce the cost. Our results suggest that additional time shifts can further reduce cost, but that the potential cost reductions are small. Additionally, we analyzed how restricting the fraction of trains that can be shifted affects the savings. Many trains that were shifted added only little to the savings: By shifting only 20% of all trains, 72% to 79% of the maximum savings can be realized. Finally, we analyzed the robustness of the duties and found that our main findings still hold true if we require substantial buffer times for train drivers switching trains.

In our research, we focused on low utilized networks, since shifting trains in a highly utilized network is difficult due to track capacity. In a highly utilized network, the track capacity must be incorporated into the model. In future research, it would be interesting to see whether in highly utilized networks similar results are achievable by small time shifts, since the higher utilization by itself provides more train switching options, such that a coordination of the timetable and the crew schedule might not be necessary. Furthermore, shifting trains by small amounts of time shortly before operations might be only possible if the work council agrees on these changes, which was the situation at the company when we conducted this research. Future research could integrate additional requirements such as identical start and end times of the duties to increase the likelihood of crews and work councils accepting changes shortly before operations.

Acknowledgements Open Access funding provided by Projekt DEAL.

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

Appendix: Detailed numerical results

See Tables 5 and 6.

Table 5 Results for maximum and multiple time shifts

Dataset	δ_{\max}	$ \Delta $	Cost	Idle time	Work time	#Duties	CT [s]	GAP [%]	$ \Gamma $	$ K_{-3} $	$ K_{-2} $	$ K_{-1} $	$ K_0 $	$ K_1 $	$ K_2 $	$ K_3 $	
1	Normal	1	100.00	100.00	100.00	100	79	0.10	459	-	-	-	103.0	-	-	-	-
	5	3	99.68	97.36	99.51	97.90	276	0.97	1377	-	-	23.0	58.3	21.7	-	-	-
	10		98.07	90.58	98.41	97.52	298	0.94	1377	-	-	43.9	24.3	34.8	-	-	-
	15		95.51	82.95	95.35	95.04	300	1.13	1377	-	-	40.2	27.3	35.5	-	-	-
	20		95.34	80.31	95.78	94.56	353	1.70	1377	-	-	35.8	27.0	40.2	-	-	-
	25		93.42	82.77	94.25	90.94	424	2.38	1377	-	-	34.5	25.7	39.0	-	-	-
	30		93.31	80.47	93.68	90.74	501	2.83	1377	-	-	39.0	29.3	34.8	-	-	-
	15	3	95.51	82.95	95.35	95.04	300	1.13	1377	-	-	40.2	27.3	35.5	-	-	-
		5	95.38	81.01	95.06	94.46	1184	1.27	2295	-	35.2	9.3	19.0	12.0	27.5	-	-
		7	95.31	80.96	94.95	94.72	2770	1.31	3213	32.0	8.1	7.3	11.5	9.3	10.1	24.8	-
2	Normal	1	100.00	100.00	100.00	100	234	0.05	781	-	-	-	149.0	-	-	-	-
	5	3	97.84	88.53	97.96	98.98	3149	0.66	2343	-	-	89.7	30.3	29.0	-	-	-
	10		96.13	80.92	96.13	96.31	3498	1.17	2343	-	-	95.3	36.6	17.1	-	-	-
	15		94.42	77.43	95.24	96.63	3853	1.67	2343	-	-	95.3	39.7	14.0	-	-	-
	20		93.01	68.89	93.80	96.20	3756	1.52	2343	-	-	85.3	35.2	28.5	-	-	-
	25		93.20	67.03	93.55	94.49	4418	2.12	2343	-	-	82.7	37.2	29.2	-	-	-
	30		93.24	65.16	93.61	93.63	4612	2.70	2343	-	-	72.5	47.3	29.2	-	-	-
	15	3	94.42	77.43	95.24	96.63	3853	1.67	2343	-	-	95.3	39.7	14.0	-	-	-
		5	94.29	72.44	95.26	94.87	17956	2.06	3905	-	49.7	21.2	20.2	20.2	12.8	45.1	-
		7	94.15	72.65	95.05	94.67	43531	1.97	5467	48.4	17.0	8.2	13.7	9.5	10.4	41.8	-

Table 5 (continued)

Dataset	δ_{\max}	$ \Delta $	Cost	Idle time	Work time	#Duties	CT [s]	GAP [%]	$ \Gamma $	$ K_{-3} $	$ K_{-2} $	$ K_{-1} $	$ K_0 $	$ K_1 $	$ K_2 $	$ K_3 $
3	Normal	1	100.00	100.00	100.00	100	118	0.02	1279	-	-	-	338.0	-	-	-
	5	3	95.64	85.13	95.97	97.31	4156	0.39	3837	-	-	188.7	56.3	93.0	-	-
	10		93.08	72.52	93.16	92.95	7610	0.95	3837	-	-	135.5	66.3	136.2	-	-
	15		91.00	72.92	91.80	93.02	6554	1.34	3837	-	-	129.4	82.8	125.8	-	-
	20		89.78	67.15	90.35	92.46	5441	1.53	3837	-	-	133.0	91.5	113.5	-	-
	25		89.29	66.17	89.86	91.13	8326	2.00	3837	-	-	194.0	86.5	57.5	-	-
	30		88.38	64.45	89.05	90.59	8676	1.95	3837	-	-	134.2	88.83	115.0	-	-
	15	3	91.00	72.92	91.80	93.02	6554	1.34	3837	-	-	129.4	82.8	125.8	-	-
	5	5	90.92	73.25	91.77	92.28	36005	1.87	6395	-	127.0	37.6	31.0	44.9	97.5	-
	7	7	90.77	71.82	91.59	92.77	46402	2.00	8953	123.5	29.1	23.5	19.3	23.2	31.7	87.7

Table 6 Results for limiting the number of shifted trains ($\delta_{max} = 15$)

Dataset	α	$ \Delta $	Cost	Idle time	Work time	Savings [%]	$ K_{-1} $	$ K_0 $	$ K_1 $
1	0.0	3	100.00	100.00	100.00	0	–	103.00	–
	0.1	3	97.31	94.74	97.23	60	5.00	93.00	5.00
	0.2	3	96.44	89.50	96.35	79	10.47	82.00	10.53
	0.3	3	96.06	87.05	95.91	88	15.93	72.00	14.93
	0.4	3	95.76	84.83	95.77	94	23.47	62.00	17.53
	0.5	3	95.58	83.54	95.48	98	29.27	51.00	22.73
	0.6	3	95.51	82.75	95.36	100	34.33	41.60	27.07
	0.7	3	95.51	82.91	95.33	100	37.67	34.67	30.67
	0.8	3	95.51	82.95	95.33	100	43.00	25.13	34.87
	0.9	3	95.51	82.37	95.36	100	45.33	22.80	34.87
2	1.0	3	95.51	82.95	95.35	100	40.23	27.31	35.46
	0.0	3	100.00	100.00	100.00	0	–	149.00	–
	0.1	3	97.59	94.98	97.99	43	12.33	134.00	2.67
	0.2	3	95.97	88.94	96.26	72	22.50	119.00	7.50
	0.3	3	95.24	84.09	95.78	85	31.50	104.00	13.50
	0.4	3	94.72	79.09	95.62	95	42.33	89.00	17.67
	0.5	3	94.56	77.89	95.25	97	55.29	69.00	24.71
	0.6	3	94.50	76.95	95.21	98	66.67	60.00	22.33
	0.7	3	94.40	77.24	95.12	100	77.50	46.50	25.00
	0.8	3	94.44	77.92	95.12	100	80.83	40.67	27.50
3	0.9	3	94.49	76.87	95.25	100	79.50	42.17	27.33
	1.0	3	94.41	77.43	95.16	100	88.50	40.44	20.06
	0.0	3	100.00	100.00	100.00	0	–	338.00	–
	0.1	3	95.50	87.84	95.82	50	24.40	304.00	9.60
	0.2	3	93.43	82.25	93.80	72	49.00	270.00	19.00
	0.3	3	92.53	79.49	93.32	82	70.67	237.00	30.33
	0.4	3	91.78	76.84	92.63	91	92.67	203.00	42.33
	0.5	3	91.30	75.08	92.18	96	115.00	169.00	53.67
	0.6	3	91.07	73.63	91.89	98	135.50	135.00	67.50
	0.7	3	90.98	73.59	91.97	99	156.17	101.67	80.17
0.8	3	90.94	72.72	91.73	100	172.17	84.67	81.17	
0.9	3	90.91	72.56	91.80	100	174.83	79.67	83.50	
1.0	3	90.96	72.86	91.76	100	177.25	82.13	78.63	

References

- Alfieri A, Kroon L, van de Velde S (2007) Personnel scheduling in a complex logistic system: a railway application case. *Journal of Intelligent Manufacturing* 18(2):223–232. <https://doi.org/10.1007/s10845-007-0017-9>
- Bach L, Gendreau M, Wøhlk S (2015) Freight railway operator timetabling and engine scheduling. *European Journal of Operational Research* 241(2):309–319. <https://doi.org/10.1016/j.ejor.2014.08.036>

- Bach L, Dollevoet T, Huisman D (2016) Integrating timetabling and crew scheduling at a freight railway operator. *Transportation Science* 50(3):878–891. <https://doi.org/10.1287/trsc.2015.0648>
- Caimi G, Kroon L, Liebchen C (2017) Models for railway timetable optimization: Applicability and applications in practice. *Journal of Rail Transport Planning & Management* 6(4):285–312. <https://doi.org/10.1016/j.jrtpm.2016.11.002>
- Desrochers M, Soumis F (1988) A generalized permanent labelling algorithm for the shortest path problem with time windows. *INFOR: Information Systems and Operational Research* 26(3):191–212. <https://doi.org/10.1080/03155986.1988.11732063>
- Dollevoet T, Huisman D, Kroon LG, Veelenturf LP, Wagenaar JC (2017) Application of an iterative framework for real-time railway rescheduling. *Computers & Operations Research* 78:203–217. <https://doi.org/10.1016/j.cor.2016.08.011>
- Freling R, Lentink RM, Odijk M (2000) Scheduling train crews: a case study for the dutch railways
- Freling R, Huisman D, Wagelmans AP (2003) Models and algorithms for integration of vehicle and crew scheduling. *Journal of Scheduling* 1:63–85
- Heil J, Hoffmann K, Buscher U (2019) Railway crew scheduling: Models, methods and applications. *European Journal of Operational Research*. <https://doi.org/10.1016/j.ejor.2019.06.016>
- Huisman D, Freling R, Wagelmans APM (2005a) Multiple-depot integrated vehicle and crew scheduling. *Transportation Science* 39(4):491–502. <https://doi.org/10.1287/trsc.1040.0104>
- Huisman D, Kroon LG, Lentink RM, Vromans MJCM (2005b) Operations research in passenger railway transportation. *Statistica Neerlandica* 59:467–497
- Jütte S, Thonemann UW (2015) A graph partitioning strategy for solving large-scale crew scheduling problems. *OR Spectrum* 37(1):137–170. <https://doi.org/10.1007/s00291-014-0381-8>
- Jütte S, Albers M, Thonemann UW, Haase K (2011) Optimizing railway crew scheduling at db schenker. *Interfaces* 41(2):109–122. <https://doi.org/10.1287/inte.1100.0549>
- Kliewer N, Amberg B, Amberg B (2012) Multiple depot vehicle and crew scheduling with time windows for scheduled trips. *Public Transport* 3(3):213–244. <https://doi.org/10.1007/s12469-011-0049-6>
- Kroon L, Fischetti M (2000) System sciences 2000: Proceedings of the 33rd annual hawaii international conference, maui, 2000 <http://ieeexplore.ieee.org/servlet/opac?punumber=6709>
- Kroon L, Huisman D, Abbink E, Fioole PJ, Fischetti M, Maróti G, Schrijver A, Steenbeek A, Ybema R (2009) The new dutch timetable: The or revolution. *Interfaces* 39(1):6–17. <https://doi.org/10.1287/inte.1080.0409>
- Lusby RM, Larsen J, Ehrgott M, Ryan D (2011) Railway track allocation: Models and methods. *OR Spectrum* 33(4):843–883. <https://doi.org/10.1007/s00291-009-0189-0>
- Mercier A, Soumis F (2007) An integrated aircraft routing, crew scheduling and flight retiming model. *Computers & Operations Research* 34(8):2251–2265. <https://doi.org/10.1016/j.cor.2005.09.001>
- Nielsen LK, Kroon L, Maróti G (2012) A rolling horizon approach for disruption management of railway rolling stock. *European Journal of Operational Research* 220(2):496–509
- Schmid V, Ehmke JF (2015) Integrated timetabling and vehicle scheduling with balanced departure times. *OR Spectrum* 37(4):903–928. <https://doi.org/10.1007/s00291-015-0398-7>
- Umweltbundesamt (ed) (2010) Schienennetz 2025/2030: Ausbaukonzeption für einen leistungsfähigen Schienengüterverkehr in Deutschland. 42/2010, Dessau-Roßlau
- Veelenturf LP, Potthoff D, Huisman D, Kroon LG (2012) Railway crew rescheduling with retiming. *Transportation Research Part C: Emerging Technologies* 20(1):95–110. <https://doi.org/10.1016/j.trc.2010.09.008>
- Veelenturf LP, Kidd MP, Cacchiani V, Kroon LG, Toth P (2016) A railway timetable rescheduling approach for handling large-scale disruptions. *Transportation Science* 50(3):841–862
- Wieberneit N (2007) Service network design for freight transportation: a review. *OR Spectrum* 30(1):77–112. <https://doi.org/10.1007/s00291-007-0079-2>

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