Janis S. Neufeld, Martin Scheffler, Felix Tamke, Kirsten Hoffmann, Udo Buscher

An efficient column generation approach for practical railway crew scheduling with attendance rates

Erstveröffentlichung in / First published in:

European Journal of Operational Research. 2021. 293 (3), S. 1113-1130. Elsevier. ISSN 0377-2217.

DOI: https://doi.org/10.1016/j.ejor.2020.12.058

An Efficient Column Generation Approach for Practical Railway Crew Scheduling with Attendance Rates

Janis S. Neufeld[a,b], Martin Scheffler[a,*], Felix Tamke[a], Kirsten Hoffmann[a], Udo Buscher[a]

[a] *Faculty of Business and Economics, TU Dresden, 01062 Dresden, Germany*
[b] *Faculty of Economics and Management, Otto von Guericke University Magdeburg, 39106 Magdeburg, Germany*

Abstract

The crew scheduling problem with attendance rates is highly relevant for regional passenger rail transport in Germany. Its major characteristic is that only a certain percentage of trains have to be covered by crew members or conductors, causing a significant increase in complexity. Despite being commonly found in regional transport networks, discussions regarding this issue remain relatively rare in the literature. We propose a novel hybrid column generation approach for a real-world problem in railway passenger transport. To the best of our knowledge, several realistic requirements that are necessary for successful application of generated schedules in practice have been integrated for the first time in this study. A mixed integer programming model is used to solve the master problem, whereas a genetic algorithm is applied for the pricing problem. Several improvement strategies are applied to accelerate the solution process; these strategies are analyzed in detail and are exemplified. The effectiveness of the proposed algorithm is proven by a comprehensive computational study using real-world instances, which are made publicly available. Further we provide real optimality gaps on average less than 10% based on lower bounds generated by solving an arc flow formulation. The developed approach is successfully used in practice by DB Regio AG.

Keywords: Transportation; Railway Crew Scheduling; Attendance Rates; Column Generation; Real-world Application

*Corresponding author

Email addresses: `janis_sebastian.neufeld@tu-dresden.de` (Janis S. Neufeld), `martin.scheffler@tu-dresden.de` (Martin Scheffler), `felix.tamke@tu-dresden.de` (Felix Tamke), `kirsten.hoffmann@tu-dresden.de` (Kirsten Hoffmann), `udo.buscher@tu-dresden.de` (Udo Buscher)

February 15, 2023

1. Introduction

In Germany, federal states or subsidiary transport associations are responsible for organizing and implementing regional passenger rail transport. Thus, they define lines and timetables for the regional railway networks. Furthermore, specific requirements are detailed, such as the type of vehicles and pricing systems. These conditions have to be met by railway companies that apply for network operation. The liberalization of German regional passenger rail transport has led to increasing competition between the tendering processes of different railway companies. As a result of high cost pressure, efficient deployment of personnel, vehicles, and resources by the railway companies is crucial for their success. This holds true across all levels of the planning process in regional passenger rail transport. Based on the conditions that are established by the transport association, rolling stock scheduling, maintenance planning, and crew scheduling have to be carried out by the railway company before the generated schedules are assigned to specific vehicles and personnel (rostering) (Hoffmann et al., 2017). In particular, crew scheduling has a substantial influence on total costs. It is a part of tactical planning and results in an anonymous crew schedule, i.e., a set of duties that have not been assigned to particular employees. A crew on a train consists of a train driver and one or several conductors who are responsible for checking tickets, customer service, and certain operational tasks.

A special challenge in the crew scheduling of conductors is the common requirement of attendance rates, which means that only a defined rate of trains needs to be covered. Attendance rates are set by superordinate transport associations and were introduced to save costs. If attendance rates are not met by the employed crew schedule, the liable railway company must pay a contractual penalty. For the underlying planning problem, attendance rates result in an additional degree of freedom compared to the crew scheduling problem (CSP), i.e., in addition to the assignment of conductors to trains, the trains that are attended have to be selected first.

The crew scheduling problem with attendance rates (CSPAR) has rarely been studied in the literature to date, and research has been limited to one conductor per trip at most (Heil et al., 2020). Nevertheless, it constitutes a major planning challenge for practical crew scheduling. Thus, the goal of our work is to present a novel hybrid column generation approach for solving the CSPAR that was developed and implemented as a client-server program during a long-term project with DB Regio AG (Neufeld, 2019).

In this paper, we provide four major contributions. First, several real-world specifications, such as multi-manning and part-time employees, are considered for the first time. Although these specifications have not been considered in the literature to date, they are required by transport associations and planners and are therefore vital for a successful application in practice. To bridge this gap, we present a new overlapping multi-period railway crew scheduling problem with attendance rates (OMCSPAR) that can be extended by various restrictions.

Second, based on the problem description and basic algorithm from the literature, sophisticated methodological enhancements are presented to enable a solution of practical instances within a reasonable time. This includes a novel three-phase solution procedure for generating initial solutions. Additionally, we quicken the subsequent column generation process by integrating various improvements. Our algorithmic contributions are analyzed using several real examples that are based on 14 German regional railway networks. We show that these improvements allow us to solve many previously intractable instances and provide decision support for considerably large networks for the first time. Moreover, we demonstrate that the presented approach is able to generate optimal solutions for small real-world instances and we provide lower bounds for larger networks based on solving an arc flow formulation.

Third, we discuss the cost effects of attendance rates and some other requirements established by federal states or subsidiary transport associations in the tender process. Thus, we not only consider the perspective of railway planners but also provide some managerial insights for decision makers in federal states and transport associations.

Finally, we make the considered real-world instances publicly available in a xml-based file format. In addition, we have provided and published a test script that contains all considered rules for the duty generation. It can be used to easily check feasibility of a newly generated schedule and serves as explicit definition of the considered requirements. This allows reproducibility of our results as well as comparison of different crew scheduling approaches. The provided instances can also be used for testing other crew scheduling approaches without attendance rates.

The remainder of the paper is structured as follows: Section 2 gives an overview of the relevant literature on railway crew scheduling. The studied problem is defined in detail in Section 3, and various practical requirements are described. These requirements form the basis for the mixed integer programming formulation of the OMCSPAR. The applied hybrid column generation approach is presented in Section 4. Special attention is paid to the initial solution, which has a substantial influence on the performance of the algorithms, and to the genetic algorithm for solving the pricing problem. A comprehensive computational study based on several German real-world instances is presented in Section 5. Managerial insights into the effects of attendance rates are provided in Section 6. Section 7 closes with concluding remarks and constructive directions for future research.

2. Related Literature

The CSP first arose in the airline and bus industries (Arabeyre et al., 1969; Carraresi and Gallo, 1984; Van den Bergh et al., 2013; Ibarra-Rojas et al., 2015; Kasirzadeh et al., 2017). Since then, it has been applied to other transportation sectors; in particular, several approaches in the railway sector were published after 1995. For detailed overviews of models and methods for the various planning tasks in the railway industry, we refer to Caprara et al. (2007); Huisman et al. (2005); Narayanaswami and Rangaraj (2011); Teodorović and Janić (2017); Heil et al. (2020). Usually, crew scheduling models have been proposed for practical problems; consequently, such models often comprise specific characteristics and challenges (Barnhart et al., 2003). At the same time, a common property is that large-scale problems have to be solved.

Two prevalent modeling approaches have evolved (Suyabatmaz and Şahin, 2015): network flow models and set covering or set partitioning formulations. All in all, network flow models are seldom used (e.g., Şahin and Yüceoğlu (2011); Vaidyanathan et al. (2007); Fuentes et al. (2019)), whereas set covering/partitioning approaches form the majority of publications. Column generation, in particular, has been proven to be suitable for solving practical instances by exerting a reasonable computational effort (Caprara et al., 1997, 2007; Ernst et al., 2001; Jütte et al., 2011; Shen and Chen, 2014). Bengtsson et al. (2007) present an algorithm for a problem similar to the one discussed herein but without attendance rates. A column generation approach is applied to solve the pricing problem through the k-shortest path enumeration. Nishi et al. (2011) present dual inequalities that accelerate column generation and reduce the number of iterations. Given the NP-hard nature of the CSP (Kwan, 2011), metaheuristics have also been developed. Among these are tabu search and genetic algorithms (Shen et al., 2013). Yaghini et al. (2015) propose a train driver CSP through a combined metaheuristic and mathematical programming approach. Recently, decomposition techniques were applied to CSPs in rail freight transport as well, leading to considerably promising results (Jütte and Thonemann, 2012, 2015). Janacek et al. (2017) use a column generation approach to generate periodic crew schedules.

Furthermore, the literature has also discussed integrated crew-scheduling approaches combined with timetabling (Bach et al., 2016) or vehicle scheduling (Dauzère-Pérès et al., 2015; Steinzen et al., 2009) as well as rescheduling problems (Veelenturf et al., 2014) in recent years. To the best of our knowledge, attendance rates have only been considered by Hoffmann et al. (2017) and Hoffmann and Buscher (2019). Such an approach is elaborated upon in the following text in greater detail.

3. Problem Definition

3.1. Problem Description and Practical Requirements

To generate crew schedules that are applicable in real-world railway networks, various restrictions and practical requirements must be considered. The objective is to find a schedule that satisfies these requirements with minimal costs. Scientifically developed algorithms may lead to very good solutions regarding a defined objective function; nevertheless, at the same time, the generated schedules are not satisfactory from a planner's view or are not viable at all. The application of the proposed solution approach in practice showed that the consideration of the following requirements is crucial for fulfilling regionally differing conditions in regional transport. However, several of these requirements have not been mentioned in the existing literature. In the following section, we address the differences in the literature in a more detailed manner. All the requirements described by Jütte et al. (2011) and Hoffmann et al. (2017) are taken into consideration.

3.1.1. Operating Conditions

Operating conditions specify the general structure of duties and guarantee a trouble-free realization. A *duty* is defined as a combination of consecutive trips covered by a certain conductor on a given day. Each *trip* is characterized by a designated departure time, departure station, arrival time, and arrival station and represents the smallest planning entity. On a superordinate level, a *train* can consist of several trips. Because a change of trains is not possible at every stop, a limited number of stations, so-called *relief points*, is usually defined at which changeovers are possible.

Apart from relief points, *crew bases* are important nodes in regional railway transport networks. A crew base is associated with a certain station, and each duty of a conductor has to start and end at the same crew base. Hence, conductors are assigned to crew bases; each crew base can only have a maximum number of employees assigned to it. In contrast to Hoffmann et al. (2017), we support the separation between full-time employees and part-time employees, who usually perform shorter duties. This distinction is important for planners because not all current conductors are full-time employees. In addition, recruiting new conductors for regional railway companies is difficult, and working part-time is an appealing option for prospective conductors.

Duties are usually created on a daily basis, i.e., a time period from the start of the first trip in the morning until the end of operations at night is considered. In particular, for city trains in larger urban regions and during weekends, there is often no end of operations. Thus, extending the considered time span for generating duties is inevitable to ensure that trips at night can be integrated into valid duties. As a result, duties may consist of trips of two consecutive days; therefore, we must consider overlapping duties similar to Abbink et al. (2011) in the pricing problem.

Furthermore, planners may desire to control the number and daily distribution of morning, day, evening, and night duties for each crew base. These categories are dependent on the starting times of the duties and can represent the preferences of conductors. For example, if morning and night duties are less popular, the distribution has higher percentages for day and evening duties. However, such patterns can lead to competing goals, particularly if attendance rates differ by the time of day because a higher rate at a certain time correlates with a higher number of duties.

3.1.2. Legal Requirements and Labor Contracts Regulations

Labor contracts and legal regulations specify several characteristics of a feasible duty. According to the German Working Hours Act, three types of working time can be distinguished. First, *duty time* is the time from signing on at the beginning of a duty to signing off at the end of the duty. Second, *protected working time* is defined as duty time excluding all breaks, deadhead times, and idle times. Finally, *paid time* is specified as the duty time excluding breaks. Because full-time conductors are supposed to have five workdays (i.e., five duties) per week on average, the average paid time of all duties must be restricted within certain bounds. For a detailed description of the legal requirements considered, we refer to Jütte et al. (2011) and Hoffmann et al. (2017), although the concrete values may vary depending on the context.

3.1.3. Transportation Contract

The third category of conditions is caused by the transportation contract of the respective transport network, which is announced by the transport associations. From this contract, attendance rates in particular have a major influence on the arising CSP. Attendance rates are defined as the percentage of kilometers of all trips with a common rate that must be covered by conductors. The rates can depend on certain lines, product types, track sections, train numbers, or the time of the day and usually range between 0 % (i.e., no conductor is necessary) and 100 %. The latter indicates that the trip must always be accompanied by a conductor. If the attendance rate is

100 % for all trips, the considered problem equals the crew scheduling of train drivers studied in the literature. As an extension of the known literature, we consider rates higher than 100 % that are required in some regional railway networks. Therefore, multiple conductors must cover the same trip (multi-manning). Multi-manning is particularly important for rush hour trips in which a solitary conductor cannot control all the passengers or for the evening to provide a greater sense of security.

Finally, a uniform distribution of the attended trips over the planning period can be claimed to avoid a predictable or imbalanced appearance of conductors on trains. The uniform distribution is typically ensured by conducting each trip at least once within a period of two weeks. In other transport networks, a weaker variant is demanded, and accompanying at least one trip by each train (i.e., train number) within the requested period is sufficient. Thus, both definitions of uniform distribution must be integrated, and a planning horizon of 14 days is usually chosen for the tactical railway CSP.

To provide a brief summary, Table 1 presents the additional requirements for railway CSPs with attendance rates that are considered in the present research compared to those considered in the known literature.

Table 1: Comparison of Considered Requirements to the Known Literature on Railway Crew Scheduling Problems with Attendance Rates

	legal requirements	operational requirements	attendance rates	overlapping duties	multi-manning	uniform distribution trips	uniform distribution trains	crew base capacities	part-time employees	daily duty distribution
Literature	•	•	•	•		•		•		
This paper	•	•	•	•	•	•	•	•	•	•

3.2. Mathematical Problem Formulation

3.2.1. Notation for Sets, Parameters, and Decision Variables

In the following section, we extend the multi-period railway crew scheduling model with variable attendance rates presented in Hoffmann et al. (2017) by the various aforementioned requirements. We distinguish between the basic OMCSPAR, which takes the coverage of attendance rates into account, and additional requirements demanded by the transportation contract or railway planners.

OMCSPAR aims to find a minimal cost combination of duties selected from a set of feasible duties N. The planning horizon consists of $|K|$ days with K as a set of days of the week, and each duty $j \in N$ begins on a specific day $k \in K$. Thus, we define set N_k as a set of duties starting on day k. Moreover, a duty covers a subset of trips $i \in M$, with M representing the set of all trips in the transportation network. Hence, a duty can be represented by a column in matrix $A \in \{0,1\}^{|M|\times|N|}$ with $a_{ij} = 1$ if duty j covers trip i and 0 otherwise. A trip i can exist on a single day $k \in K$ or on several days of the planning horizon K. As a result, M_k can be defined as a subset of M that contains all the trips $i \in M$ that take place on day k. Additionally, the planner can specify trips that must be checked regardless of their attendance rate. To this end, we define the set of mandatory trips O and add trip $i \in M$ on day k as pair (i, k) if it is marked by the planner.

Further, creating overlapping duties may be beneficial or even necessary for practical applications. Figure 1 shows the timespan from which trips are considered for each day of the planning horizon. A trip $i \in M_k$, which is operated prior to a certain time limit on day k, may be covered not only by duties starting on day k but also by duties beginning the day before. For example, if a trip starts on a Tuesday between the start of day and the time limit (e.g., 12 a.m.), this trip may be integrated in a duty from Tuesday ($k = 1$), but also a duty that starts on a Monday ($k = 0$). In other words, the days of our planning horizon overlap. In addition, we consider a cyclic planning horizon (one week or two weeks), as is also shown in Figure 1. Hence, the day previous to Monday ($k = 0$) is Sunday ($k = 6$); consequently, $k - 1$ is an invalid general representation of the day before k. Thus, we apply $\bar{k} = (k - 1) \bmod |K|$ to determine the day before k correctly. Note that enabling overlapping duties only on

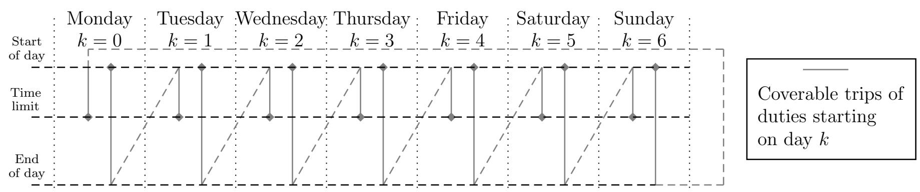


Figure 1: Representation of Duties Across Different Days

certain nights (e.g., weekends) is also possible. However, this occurs in the pricing problem because only the set of available trips for each of the $|K|$ sub-problems (see Section 4.3) has to be adjusted accordingly.

Furthermore, let G be the set of all attendance rates $g \in \mathbb{R}_0^+$ defined in the transportation contract. We can determine d_{ig} as the distance of trip $i \in M$ with attendance rate $g \in G$. Note that index g is necessary because one trip may consist of several sections with varying attendance rates.

The costs c_j of a feasible duty $j \in N$ consist of two parts. First, fixed costs c^{fix} occur for every duty. Second, every minute of the paid working time τ_j of duty j is rated with variable costs c^{var}, yielding $c_j = c^{\text{fix}} + c^{\text{var}} \cdot \tau_j$. The paid working time is calculated in accordance with the operating conditions and legal requirements described in Section 3.1. If a duty does not meet an operating condition or a legal requirement, we penalize the use of this duty with costs c^{pen}.

In addition to the sets and parameters described previously, we introduce the following decision variables. Integer variable x_j corresponds to the frequency of duty $j \in N$ in the solution. Owing to potential multi-manning, x_j is not always a binary variable, as it is in most crew scheduling approaches, but an integer variable. For example, if two conductors are assigned to a duty j, $x_j = 2$, i.e. j is selected two times in the solution.The maximum frequency of duty $j \in N$ is defined by the highest attendance rate of all trips included in this duty. If a duty would be selected more often than this highest attendance rate, at least one of these duties would only increase costs without improving the stipulated coverage of trips. Hence, the upper bound λ_j^{u} of x_j can be determined with $\lambda_j^{\mathrm{u}} = \lceil \max_{i \in M, g \in G} (\{a_{ij} g \,|\, d_{ig} > 0\}) \rceil$. The check for $d_{ig} > 0$ is necessary because attendance rate $g \in G$ only needs to be considered if trip $i \in M$ requires an attendance rate of g. Furthermore, we use integer variables y_{ik} to model the number of conductors attending trip $i \in M$ on day $k \in K$ in the solution. Similar to the variable x_j, the frequency depends on the attendance rates of the trip. Therefore, we can define a lower (upper) bound μ_i^{l} (μ_i^{u}) as follows: $\mu_i^{\mathrm{l}} = \lfloor \max_{g \in G} (\{g \,|\, d_{ig} > 0\}) \rfloor$, $\mu_i^{\mathrm{u}} = \lceil \max_{g \in G} (\{g \,|\, d_{ig} > 0\}) \rceil$. Thus, for example, if the attendance rate of a trip is 1.5, the trip should comprise at least one duty and at most two duties.

3.2.2. Basic OMCSPAR set covering model

Using the notation presented earlier, we introduce the basic OMCSPAR as follows.

$$\text{[OMCSPAR]:} \quad \min \sum_{j \in N} c_j x_j \tag{1}$$

$$\text{s.t.} \quad \sum_{k \in K} \sum_{i \in M_k} d_{ig} y_{ik} \geq g \sum_{k \in K} \sum_{i \in M_k} d_{ig} \qquad \forall\, g \in G \tag{2}$$

$$\sum_{j \in N_{\bar{k}}} a_{ij} x_j + \sum_{j \in N_k} a_{ij} x_j \geq y_{ik} \qquad \forall\, k \in K, i \in M_k \tag{3}$$

$$y_{ik} = \mu_i^{\mathrm{u}} \qquad \forall\, (i,k) \in O \tag{4}$$

$$x_j \leq \lambda_j^{\mathrm{u}} \qquad \forall\, j \in N \tag{5}$$

$$\mu_i^{\mathrm{l}} \leq y_{ik} \leq \mu_i^{\mathrm{u}} \qquad \forall\, k \in K, i \in M_k \tag{6}$$

$$x_j \in \mathbb{N} \qquad \forall\, j \in N \tag{7}$$

$$y_{ik} \in \mathbb{N} \qquad \forall\, k \in K, i \in M_k \tag{8}$$

The objective function (1) minimizes the total operating costs. Constraints (2) ensure compliance with the required attendance rates. This compliance is achieved by forcing the accumulated distance of the covered trips of each attendance rate in the solution schedule to be greater than or equal to the requested percentage of the total distance assigned to this rate. Constraints (3) are linking variables x_j and y_{ik}. Hence, there has to be at least μ_i^{l} duty $j \in N_k$ or $j \in N_{\bar{k}}$ in the solution schedule covering trip i on day k if trip i on day k is in the solution. Note that trip i can be covered by a duty starting on day k or $\bar{k}$. Furthermore, deadheads are possible because of the inequality relation. The inclusion of all mandatory trips in the final schedule is modeled by constraints (4). This constraint has been slightly modified to meet attendance rates higher than 100 %. Finally, constraints (5)–(8) set the aforementioned bounds and state the domains to enable attendance rates of more than 100 %.

In the following section, we present the additional requirements that can be necessary to generate valid and accepted crew schedules.

3.2.3. Average paid time

As explained earlier, balancing the paid working time of duties across the week is necessary. In our approach, we define that the average paid working time of all duties of the planning horizon must be between a lower bound $\tau^{\min}$ and an upper bound $\tau^{\max}$. Hence, we introduce the following constraints:

$$\sum_{j \in N} \tau_j x_j \geq \tau^{\min} \sum_{j \in N} x_j \tag{9}$$

$$\sum_{j \in N} \tau_j x_j \leq \tau^{\max} \sum_{j \in N} x_j. \tag{10}$$

Constraints (9) guarantee that the average paid time over all duties in the solution schedule is either longer than or equal to the permitted lower bound, and constraints (10) ensure compliance with the upper bound.

3.2.4. Uniform distribution

The uniform distribution should ensure that a variety of trips is checked. We provide two different approaches to model this requirement. The first variant

$$\sum_{k \in K \mid i \in M_k} y_{ik} \geq 1 \qquad \forall\ i \in M \tag{11}$$

guarantees that each trip is covered at least once in the planning horizon. Note that this corresponds to the trip-based uniform distribution in Section 3.1.

In addition, we present a new alternative variant called train-based uniform distribution. For this purpose, we define set M_{kz} as a set of all trips on day k associated with train number $z \in Z$, where Z is the set of all train numbers. This definition enables the train-based uniform distribution to be modeled as follows:

$$\sum_{k \in K} \sum_{i \in M_{kz}} y_{ik} \geq 1 \qquad \forall\ z \in Z. \tag{12}$$

Note that one variant can be used at most, i.e., a uniform distribution can also be deactivated.

3.2.5. Crew base capacity

Another practical requirement introduced in Section 3.1 is the maximum number of duties starting at a crew base. Let E be the set of all crew bases in the network; subsequently, parameter b_{je} equals one if duty j starts at crew base e and zero otherwise. The capacity of each crew base $e \in E$ may vary depending on the day $k \in K$ and is denoted by Q_{ek}. We can now introduce

$$\sum_{j \in N_k} b_{je} x_j \leq Q_{ek} \qquad \forall\ e \in E, k \in K \tag{13}$$

as crew base capacity constraints.

In some scenarios, however, planners must distinguish between full-time and part-time employees. For this purpose, the notation will be extended again. First, we define the maximum number of full-time duties starting at e on day k by Q^{FT}_{ek}. Second, we set binary parameter w_j to one if duty j must be performed by a full-time employee and to zero otherwise. A duty is considered invalid for a part-time employee if its duty time is longer than a predefined but variable threshold. Note that all part-time duties are included in the general crew base capacity Q_{ek} because shorter duties can be operated by part-time and full-time employees. Hence, considering them separately is unnecessary, and constraints

$$\sum_{j \in N_k} b_{je} w_j x_j \leq Q_{ek}^{\mathrm{FT}} \qquad \forall\, e \in E, k \in K \tag{14}$$

are used to restrict the maximum number of full-time employees per day for each crew base.

3.2.6. Daily duty distribution

Finally, we consider the daily distribution of duties as a further requirement that arises in railway CSPs, which may vary between different crew bases. To this end, we define a set of daytimes T for categorizing duties as early, day, late, and night and parameter l_{jet}. This parameter equals one if duty $j \in N$ starts at crew base $e \in E$ during daytime $t \in T$ and zero otherwise. Hence, only the start time of a duty is decisive for the daytime category. In addition, each crew base e has a desired percentage p_{et} of duties starting there at time of day $t \in T$. Note that we cannot apply a fixed number of duties for each daytime because we do not know how many duties begin at crew base e.

In most cases, however, meeting this quota exactly is not possible. Therefore, we introduce continuous variables $v_{et} \in \mathbb{R}$ and $u_{et} \in \mathbb{R}$ as the lower and upper deviations from the desired total number of duties starting at crew base $e \in E$ during daytime $t \in T$ and implement the daily duty distribution as the following soft constraints:

$$\sum_{j \in N} l_{jet} x_j \geq p_{et} \sum_{j \in N} b_{je} x_j - v_{et} \qquad \forall\, e \in E, t \in T \tag{15}$$

$$\sum_{j \in N} l_{jet} x_j \leq p_{et} \sum_{j \in N} b_{je} x_j + u_{et} \qquad \forall\, e \in E, t \in T. \tag{16}$$

Constraints (15) allow the number of duties that start during t at base e to remain under the desired percentage p_{et} of all duties starting at crew base e. Conversely, constraints (16) permit the number of duties that begin during t at base e to exceed the desired percentage p_{et} of all duties beginning at base e.

However, variables v_{et} and u_{et} must be penalized to control the extent of deviation. We evaluate the deviation from the desired number of duties with variable penalty costs s. Thus, the original objective function (1) must be extended by a penalty term, and we obtain the following new objective function:

$$\min \sum_{j \in N} c_j x_j + s \sum_{e \in E} \sum_{t \in T} \left(v_{et} + u_{et} \right). \tag{17}$$

Here, the higher the value of s, the greater the enforced compliance with the daily duty distribution.

4. Solution Approach for OMCSPAR

4.1. Column Generation Framework

Set covering problems in large-scale crew-scheduling applications are usually tackled by column generation approaches because the set of all feasible duties N is considerably large. Hence, a complete creation of N would be too consuming in terms of both time and memory. By contrast, column generation operates with a restricted set of duties $\bar{N}$ and successively adds new duties in an iterative process. Thus, two iteratively connected problems, called restricted master problem (RMP) and pricing problem, are applied herein. The RMP is equivalent to OMCSPAR but with the restricted set of duties $\bar{N}$ instead of N. Solving the linear relaxation of the RMP (rRMP) yields dual values that are used in the pricing problem to generate new columns with negative reduced costs, i.e., duties that may reduce the objective function value. Because our planning horizon consists of $|K|$ days, we can decompose the pricing problem in K independent problems. Thus, we create new duties for a specific day $k \in K$ and solve the rRMP with a new set of duties $\bar{N}$ in each iteration. This procedure is adapted from the cyclic generation strategy introduced in Mourgaya and Vanderbeck (2007) for a multi-period vehicle routing problem. Another approach might be to generate new duties for the entire planning horizon first and subsequently solve the rRMP. However, this approach could lead to the generation of many unused duties, which needlessly inflates the RMP.

The general flow of our column generation procedure is presented in Figure 2. The procedure will be described in detail below. As with all column generation

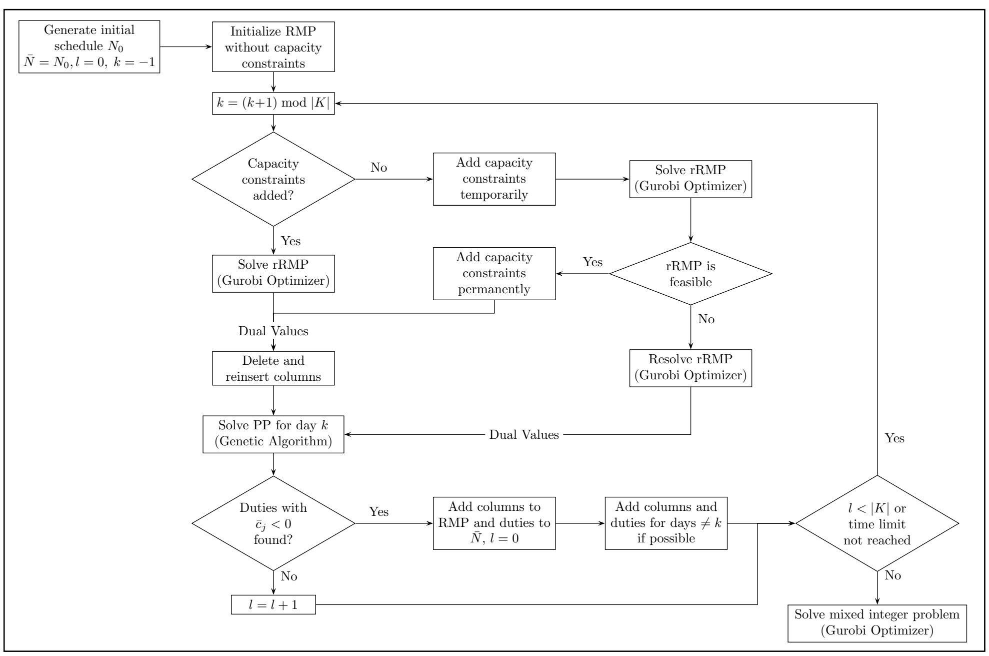


Figure 2: Flowchart of the Proposed Multi-Period Column Generation Algorithm

approaches, our algorithm starts by generating an initial set of duties N_0. We apply different strategies to determine feasible initial solutions within a short processing time. However, it is important to note that the rRMP should be able to generate a feasible solution with N_0, but N_0 itself may contain infeasible duties. We refer to Section 4.2 for an in-depth description of our approach. Subsequently, the restricted set of duties $\bar{N}$ is initialized with N_0, and control variables l and k are introduced. Variable k represents the currently considered day, whereas variable l counts the number of contiguous iterations without newly found columns. Moreover, the RMP is initialized. However, if capacity constraints ((13) and (14)) are considered for a network, these constraints are first omitted because a feasible initial solution is not guaranteed with tight crew base capacities.

Subsequently, the iterative procedure of generating new duties begins. As described previously, we iterate all days of the planning horizon using variable k and create new columns for each day. Because we omit constraints (13) and (14) during initialization, we have to add them manually. To do so, we first check whether they have been added in a previous iteration. If this is the case, we solve the linear relaxation of the RMP; if not, we add them temporarily and then solve the linear relaxation. In the case of a feasible relaxation, the capacity constraints are added permanently. Otherwise, we solve the rRMP again without the constraints. Thus, in any case, we achieve a feasible solution of the rRMP and can obtain the dual values of all constraints related to the variables x_j.

Furthermore, if the crew capacity constraints (13) and (14) are already added permanently, we attempt to remove unnecessary columns from the RMP. This should accelerate the solution of the rRMP as well as reduce memory consumption. A column is marked as unnecessary if, first, it is not a basic variable for a number of contiguous iterations (*maxAgeofDuties*) and, second, if its positive reduced costs are smaller than a predefined threshold (*reducedCostsThreshold*). Moreover, because columns are solely removed from the RMP but duties from set $\bar{N}$ are not, we can also reinsert already deleted columns with the now negative reduced costs. Consequently, favorable duties are not erroneously excluded from the final solution.

In the next step, we attempt to determine new duties for day k that may improve the objective function value, i.e., have negative reduced costs. Solving the pricing problem in an efficient manner is a crucial aspect of every column generation approach. In contrast to the initial solution procedure, we apply a genetic algorithm that only generates feasible duties. This solution approach for the pricing problem is explained in further detail in Section 4.3. If the set of new duties with negative reduced costs is not empty, we add every new duty j to $\bar{N}$ and the corresponding new column x_j to the RMP. To quicken the solution process, for all new duties, we also check whether it is possible and beneficial to add similar duties on other days of the planning horizon. Because trips do not occur every day, the feasibility of duties on all days is not guaranteed. Additionally, it is only beneficial to add a duty with negative reduced costs. If either is true, we add a similar but new duty and column. If no new duties with negative reduced costs are found by solving the pricing problem, we increase l by one.

Finally, variable k is updated, and the next iteration starts if no termination criterion is reached. We apply two different termination criteria to stop the generation of new columns. First, we use the criterion introduced in Mourgaya and Vanderbeck (2007). There, the loop stops if $i = |K|$, meaning that no new duties with reduced costs were created for K consecutive iterations. However, this approach may lead to considerably long computing times because we deal with extremely large transportation networks. Therefore, we apply a time limit as a second termination criterion.

If new columns have stopped being generated, we solve the RMP with all current columns in $\bar{N}$ as a mixed integer linear program to obtain a feasible schedule. This approach is called restricted master heuristic (Joncour et al., 2010) or price-and-branch (Desrosiers and Lübbecke, 2011) and leads to good solutions in reasonable computation times. As mentioned in cite Joncour et al. (2010), the restricted master heuristic can result in infeasible problems since the generated columns might be feasible for the rRMP, but not for the RMP. However, infeasibility is not an issue here because the set $\bar{N}$ is quite large. In contrast, the column generation method could be integrated into a branch-and-price framework to obtain optimal solutions. Unfortunately, this is not viable for the considered problem sizes as solving one node in the branch-and-bound tree with column generation could take several hours and many nodes might have to be processed. Therefore, this approach would exceed reasonable computation times.

4.2. Initial Solution

4.2.1. General Procedure

Generating an initial solution related to a column generation approach has yet to be described exhaustively. Chen and Shen (2013) use a vehicle-based approach to generate sets of potential duties. We will refer to this procedure as a vehicle-based block generator (VBBG). Hoffmann et al. (2017) describe a trip-based depth-first search within heuristic limits to create an initial solution. We refer to this procedure as a block generator (BG). Both procedures are two-stage algorithms consisting of a creating and a combining stage. For practical applications, a feasible solution that covers each trip at least once is difficult to find. Therefore, Shen and Chen (2014) use artificial variables. However, an artificial solution can be assumed to decelerate the solution process. Finally, Janacek et al. (2017) use shortest path information based on a frame concept for small problems (less than 100 trips). Generating reasonable-sized sets of potential duties has not been discussed extensively in the literature for large scale crew scheduling.

Older approaches directly discuss the enumeration of all feasible duties, which is followed by solving the RMP. Alefragis et al. (1998) use a straightforward depth-first enumeration. Caprara et al. (2001) combine the enumeration with the use of time-related shortest path information between nodes in the underlying temporal and spatial network for improving branching strategies. Goumopoulos and Housos (2004) focus on the efficiency of the feasibility checks needed for an enumeration and use shortest path-based information to generate bounds as pruning of the branching tree. Koniorczyk et al. (2015) extend this approach by heuristic limits. Clearly, an enumeration requires an accurate handling of infeasibility, whereas an initial solution can handle this more generously. Therefore, we first need to clarify different types of infeasibility and their impact on the algorithm.

An initial solution can be infeasible because of three reasons: trip infeasibility, constraint infeasibility, and duty infeasibility. Trip infeasibility (t-inf) is caused by missing or uncovered trips. For example, a trip with $g \geq 1$ (i.e. it must be attended) which is not part of any duty in N_0 causes t-inf. If an initial solution does not fulfill constraints (9) and (10) of the rRMP, it falls under constraint infeasibility (c-inf). This infeasibility is also applicable to constraints (13) and (14), but as described in Section 4.1, we treat these separately in the subsequent column generation process. We do not consider these for generating the initial solution. The duties of the initial solution are referred to as *blocks*. A block represents a symmetrical (i.e., starting and ending at the same crew base) and ordered list of trips without taking legal requirements into account, such as *maximum working time* or other time restrictions. Finally, duty infeasibility (d-inf) describes blocks that violate one of these restrictions but could theoretically be attended by a conductor. c-inf and d-inf can be fixed during the column generation approach, whereas t-inf prevents the start of this, because constraints (2) and (3) cannot be fulfilled and even the rRMP is infeasible. Thus, we extend the initial solution approach using a repair procedure (RP). The result is a three stage procedure consisting of creating, repairing, and combining, as illustrated in Figure 3.

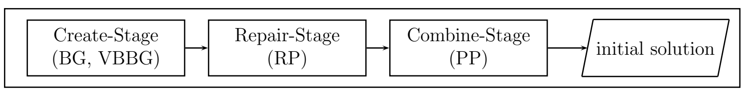


Figure 3: Flowchart of the Proposed Initial Solution Procedure

For the Combine-Stage, we have used a simple pre-processing (PP). A set of blocks is chosen randomly from the solution pool. For each of these blocks, a matching downstream block is searched for by requiring a break between both. This strategy is suitable for the considered problem sizes in regional transport.

4.2.2. Improved Create-Stage

In the first step, we performed several tests for BG described by Hoffmann et al. (2017) using different settings for the parameters *minD*, *maxD* (minimum and maximum duration of a block in minutes; generated by the BG), *maxT* (maximum accepted transition time between two subsequent trips in a block), and *maxS* (number of subsequent trips; limits the number of possible branches at each vertex of the branching tree). These preliminary tests showed that each network requires a different parameter setting for a suitable initial solution. Determining the appropriate setting is occasionally very time consuming.

To avoid this, we extend the BG by introducing three levels for the Create-Stage, i.e. three different search strategies are used for the BG. Therefore, the generator is called three times for each trip $i \in M_E$, using $M_E \subseteq M$, which contains all trips starting at any crew base. Hoffmann et al. (2017) set several network-specific limits to reduce the branching tree used for the depth-first search. We use the fixed setting $BG^{minD-maxD}_{maxS-maxT} = BG^{120-360}_{6-120}$ for each level. Note that using this setting for the original BG would result in an initial solution that is far too large. To prevent this, we introduce variables *BlockLimit*, *Depth*, and *Random*, which focus on the branching

Algorithm 1 Extend(*oldBlock*, *newTrip*)

```
parameters: minD, maxD, maxS, maxT        global variables: ct, BlockLimit, Depth, Random
 1: currentBlock = Copy(oldBlock);
 2: Add newTrip at the end of currentBlock
 3: if currentBlock is symmetrical & minD≤Duration(currentBlock)≤maxD then
 4:     Add currentBlock to N_0
 5:     increment ct by one
 6: end if
 7: if Duration(currentBlock)≤maxD & ct<BlockLimit & TripCount(currentBlock)<Depth then
 8:     determine maxS subsequent trips of newtrip with transition time ≤ maxT
 9:     if Random then
10:         for all determined subsequent trips t of newTrip in random order do
11:             Extend(currentBlock, t)
12:         end for
13:     else
14:         for all determined subsequent trips t of newTrip sorted by departure time do
15:             Extend(currentBlock, t)
16:         end for
17:     end if
18: end if
```

tree itself. The BG is implemented recursively and is based on the Extend method shown in Algorithm 1. On each level, the method is called for the first time for each trip $i \in M_E$ with different values for *Random* and *Depth*. Variable *ct* is initialized with 0 for each trip on each level. The first if branch (line 3-6) adds appropriate blocks to the initial solution N_0. The second if branch (line 7-18) is used to recursively extend the blocks with different strategies for each level. For levels one and two the else branch (line 13-17) is used. For level three the if-branch is used (line 9-12).

The underlying idea of *Depth* is quite similar to the maximum distance to the depot introduced by Koniorczyk et al. (2015). The value of *Depth* in Level 1 is based on the average duration of a trip $\bar{l}$ and is calculated by $360/\bar{l}$. For Level 2 and Level 3, the average number of trips in a block as a result of Level 1 is used. To avoid outliers on certain special networks, the value of *Depth* in Level 1 is fixed in the range $[10, 25]$, which is suitable for all the considered networks in this paper. The value of *Random* is false for Level 1 and Level 2 but is true for Level 3.

The value for *BlockLimit* is calculated by $\frac{|M|\cdot|E|\cdot 2}{|M_E|}$ and is constant for each level. This equation ensures that the total number of blocks generated by each level of the generator is controllable and network-specific. In addition, an equal distribution of blocks over the underlying temporal and spatial network is achieved.

Figure 4 illustrates a branching tree and the explored solution space for each level. To provide a clear presentation, contrary to the implementation, $maxS$ is set to two. This setting reduces the tree to a binary structure. We assume a value *Depth* of six in Level 1. In approximate terms, $maxS$ limits the width of the branching tree, and *Depth* limits the height. Considering $maxT$ and $maxD$, some branches can be ignored (dotted arrows). Note that the length of an arrow is not related to the length of the corresponding trip. Arrows (i.e., trips) leaving a node are sorted from left to right by increasing order of transition times. Further, we assume a value of seven for *BlockLimit*. Finally, each connection between two crew base nodes is assumed to have a duration that is longer than 120 minutes. For practical instances, *minD* prevents the generation of overly short blocks.

The result of Level 1 is a set of seven blocks with an average trip count of five

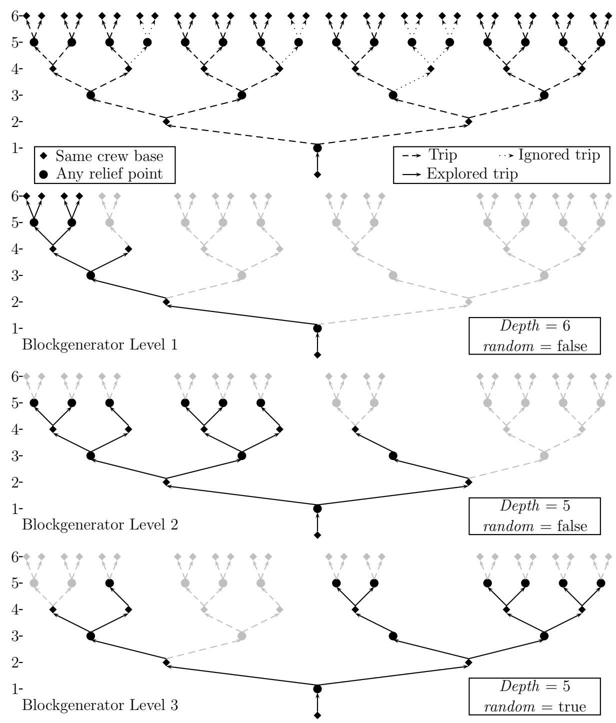


Figure 4: Depth-First Branching Trees (*BlockLimit* = 7, *maxS* = 2)

(4.857; rounded to the nearest integer), which is why the value of *Depth* is reduced to five for the following levels and the upper arrows and nodes become irrelevant (gray). Since *Random* is false in Level 1 and Level 2, the explored search space is on the left side of the tree because subsequent trips are chosen by a minimum transition time. By contrast, Level 3 is random based, and the exploration space becomes less organized. For Level 1 and Level 2, the illustrated results are deterministic; for Level 3, the result is merely an example.

4.2.3. Repair-Stage

To ensure the feasibility of the RMP, the solution pool generated by the creating stage has to be checked for uncovered trips to fulfill constraints (2), (3), (4) and (11) or (12). For each uncovered trip, finding a single block that includes the trip is sufficient. Avoiding d-inf by creating only feasible duties is unnecessary in this stage; this is assumed to be achieved by the genetic algorithm (see Section 4.3). As observed by Caprara et al. (2001) and Goumopoulos and Housos (2004) in relation to their enumeration approaches, using the shortest paths is a suitable method for connecting a sequence of trips to a crew base. Extending this idea, we use the algorithm introduced by Dijkstra (1959) to find two paths for each uncovered trip. To this end, we use the underlying space-time network, as shown in Figure 5. Each node represents a distinct

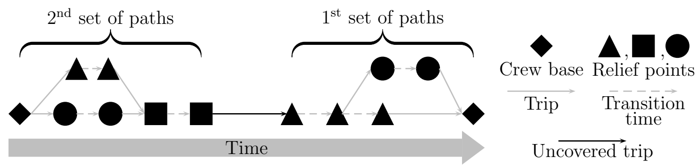


Figure 5: Spatial and Temporal Network for the Shortest Path Repair Procedure

combination of time and a relief point or a crew base. Trips (change the place) and transition times (stay in one place) are represented by arcs and are weighted by the length of the travel or transition time. Thus, all possible paths between two nodes have the same duration. To avoid t-inf, the path that is chosen by the procedure does not matter. To obtain productive paths, each transition time longer than one hour is penalized by high weight. Starting from the arrival node of an uncovered trip, we search for the shortest path to each crew base node later in time. Analogously, the departure node of an uncovered trip must be reached from an earlier crew base node. Therefore, this search is carried out backward in time. Each arc is reversed, and starting at the departure node of an uncovered trip, we search for the shortest path to each crew base earlier in time. For each crew base, this process results in two sets of paths (there and back). One path is chosen randomly from each set for each crew base, and the resulting block of joining both paths and the uncovered trip is added to N_0. Note that the presented initialization of Janacek et al. (2017) aims at choosing more than one set from each path. However, the instance sizes considered in this paper are too large to use this approach.

If at least one set is empty, the trip cannot be covered by a duty beginning at the concerned crew base; if this applies for all crew bases, the trip is not coverable. For practical application, this approach provides essential information for crew planners, e.g., the need for additional deadheads. By using this procedure, the validation of real-world data is simplified, and identifying critical trips or other issues in the network is possible.

4.2.4. Preliminary Tests

A suitable initial solution needs to be of a reasonable size and quality and simultaneously generated within an acceptable computing time. This trade-off requires a detailed view of different strategies on the Create-Stage. Furthermore, following Chen and Shen (2013), we implemented and tested a VBBG. The VBBG creates all symmetrical blocks without a change in vehicle and a working time lower than the given maximum. For a detailed description of the used networks, refer to Section 5.

Table 2 provides the results of different initial solution procedures. The first column indicates the used procedure and parameter in the Create-Stage and the additional stages that were carried out. Because the Create-Stage is essential, the table is structured in groups of three rows that used the same creating parameters. In the first row of a group, only the Create-Stage was used. In the second and third row, RP was additionally carried out. PP was only used as an additional step in the last row (all three stages are carried out). The computing time required to solve the RMP first, t_0, is a suitable indicator for the expected time needed for one iteration of the following column generation approach and increases proportionally with the size (i.e. column *size* in Table 2) of the initial solution N_0. The corresponding objective value obj_0 equals the total costs of the crew schedule. In the first group of rows, the BG setting of Hoffmann et al. (2017) is used. The second group shows the results for using the VBBG as the Create-Stage. Note that for the first two groups in each case, only the approach used in the first line is identical to the literature. In the third group (1lvlBG), only Level 1 is used. In the fourth group (2lvlBG), Level 1 and Level 2 are carried out. In the last group (3lvlBG), all three levels are used.

Table 2: Computational Results for different Initial Solution procedures (14 Days)

	N_0		Network				N_0		Network				N_0		Network			
			I		I*				II		II*				III		III*	
	size	t	t_0	obj_0	t_0	obj_0	size	t	t_0	obj_0	t_0	obj_0	size	t	t_0	obj_0	t_0	obj_0
$BG_{4-60}^{120-180}$	0.1	2	c-inf		t-inf		0.3	3	c-inf		c-inf		0.2	3	t-inf		t-inf	
+RP	0.1	4	2	73.2	3	85.3	0.3	6	c-inf		c-inf		0.2	15	3	93.4	5	109.7
+RP+PP	0.2	34	4	8.8	8	10.3	0.3	92	4	45.2	6	45.2	0.2	68	6	41.8	13	51.5
VBBG	6.3	28	267	5.7	t-inf		3.5	17	97	3.1	t-inf		0.9	6	t-inf		t-inf	
+RP	6.3	176	216	5.5	255	9.5	3.5	75	120	3.1	146	15.0	0.9	32	30	6.7	34	15.5
+RP+PP	6.3	312	254	5.5	283	9.4	3.5	131	98	3.1	96	11.0	1.0	45	32	6.7	38	15.5
1lvlBG	0.2	7	c-inf		t-inf		0.1	3	t-inf		t-inf		0.1	26	t-inf		t-inf	
+RP	0.2	9	c-inf		c-inf		0.1	5	c-inf		c-inf		0.1	31	2	6.0	4	12.1
+RP+PP	0.2	53	14	5.3	28	6.1	0.1	11	4	11.5		13.6	0.1	46	7	3.9	8	8.8
2lvlBG	0.3	9	c-inf		t-inf		0.1	4	t-inf		t-inf		0.1	42	t-inf		t-inf	
+RP	0.3	13	c-inf		c-inf		0.1	6	c-inf		c-inf		0.1	49	3	6.3	5	12.0
+RP+PP	0.3	103	25	5.3	54	6.6	0.1	14	5	11.4	10	13.5	0.1	57	9	3.9	11	8.3
3lvlBG	0.4	12	c-inf		c-inf		0.2	5	c-inf		t-inf		0.2	63	t-inf		t-inf	
+RP	0.4	18	c-inf		c-inf		0.2	7	c-inf		c-inf		0.2	68	6	16.4	10	19.7
+RP+PP	0.4	104	27	5.3	42	5.6	0.2	19	7	4.8	11	5.5	0.2	108	13	3.9	18	4.6

Notation: $BG_{maxS\ \text{-}\ maxT}^{minD\ \text{-}\ maxD}$; *size*: # duties in millions; t: CPU time in sec.; t_0: CPU time first rRMP in sec.; obj_0: objective value in millions

The function of each stage can be seen exemplarily for row-group 1lvlBG on network II. Applying RP avoids t-inf and ensures usability for real-world applications. Analogously, the need for PP to avoid c-inf can be acknowledged. Note that the computing time of RP and PP depends on the result of the first stage. The process of searching uncovered trips (RP) and choosing suitable blocks (PP) slows down with increasing block number as a result of stage one.

Certainly, using all three levels of the BG yields the best results and outperforms all procedures presented in Table 2 on networks I, I*, II*, III, and III*. On the one hand, the objective value is adequate for each network. On the other hand, the size of N_0 and the time needed for the complete three-stage-approach are reasonably small for creating an initial solution. The best objective values for network II are achieved by VBBG. However, by considering t and particularly t_0 in combination with the related size of N_0, the setting 3lvlBG evidently deals best with the underlying trade-off among feasibility, quality, and size (computing time). Hence, this setting is used for all the considered networks in this paper. The used parameter values are apparently suitable for a wide range of networks. Based on these improvements, the following column generation approach can be assumed to be accelerated by this as well (see Section 5.2).

4.3. Solving the Pricing Problem

Solving the pricing problem is one of the most challenging aspects of every column generation approach. As described in Section 4.1, $|K|$ different problems have to be solved during the algorithm. Based on Hoffmann et al. (2017) and the additional constraints introduced in Section 3.2, the reduced costs for a duty j that starts on day k are given by

$$\begin{aligned}\bar{c}_j = c_j &- \sum_{i \in M_k} a_{ij}\pi_{ik} - \sum_{i \in M_{k'}} a_{ij}\pi_{ik'} + \sum_{e \in E} b_{je}\sigma_{ek} + \sum_{e \in E} b_{je} w_j \sigma_{ek}^{\mathrm{FT}} \\ &- (\tau_j - \tau^{\min}) \cdot \rho^{\min} - (\tau^{\max} - \tau_j) \cdot \rho^{\max} \\ &- \sum_{e \in E} \sum_{t \in T} \left[(l_{jet} - b_{je} p_{et}) \cdot \gamma_{et}^{\mathrm{l}} + (b_{je} p_{et} - l_{jet}) \cdot \gamma_{et}^{\mathrm{u}} \right]\end{aligned} \tag{18}$$

using $k' = (k+1) \bmod |K|$ as the day after k, π_{ik} as the dual value of constraints (3), $\rho^{\min}$ and $\rho^{\max}$ of (9) and (10), σ_{ek} of (13), $\sigma_{ek}^{\mathrm{FT}}$ of (14), and γ_{et}^{l} as well as γ_{et}^{u} of (15) and (16). Finding duties with negative reduced costs under consideration of all requirements described in Section 3.1 represents the complete pricing problem.

In general, the pricing problem can be modeled as a resource constrained shortest path problem (RCSPP). Irnich and Desaulniers (2005) provide a detailed overview on several solution approaches for this issue. Because this problem is already an NP-hard optimization problem, a solution might be considerably time consuming. Furthermore, they note that an optimal RCSPP solution is merely required in the last pricing step. Based on the results of Albers (2009) summarized by Hoffmann et al. (2017) in the context of railway crew scheduling, dynamic programming as a common exact solution method yields its limits within the single-digit range of trips in a feasible duty. Chen and Shen (2013) introduce the notion of ignoring the RCSPP by choosing duties with negative reduced costs from a reasonably large and pre-compiled pool of promising productive duties. Moreover, a heuristic solution approach simplifies the integration of newly arising practical requirements. Therefore, we use an improved genetic algorithm (GA) based on the description of Hoffmann et al. (2017).

Some enhancements to the proposed algorithm are implemented. To achieve some kind of variation, the initial population consists of $\frac{4}{5} \cdot popSize$ best and $\frac{1}{5} \cdot popSize$ randomly selected individuals from the duty pool. The value of *popSize* is equals $|M|$, which makes a reference to the considered network. Liu et al. (2010) note the termination of the GA (in each CG-iteration) when a fixed number of feasible individuals is created. Because we are only interested in feasible duties with negative reduced costs, our GA stops immediately if more than 100 new duties ($\bar{c}_j < 0$) have been found. This slows down the growth of the duty pool in the first iterations, in particular, and ensures the use of proper dual values. We also vary the number of iterations made in the recombination phase for each k, depending on the number of new duties that are generated in previous iterations for k of the column generation approach.

Requiring symmetrical duties in combination with the exclusive use of an OPC leads to the fact that only duties with the same crew base can be used for each recombination step. This may result in the unlikely case that trips are permanently assigned to a single crew base, which happens if a trip is covered only by duties that start at the same crew base. To avoid this situation, a two point crossover (TPC) is suitable for breaking up such assignments. The OPC itself is already a complex procedure under consideration of all temporal and spatial requirements.

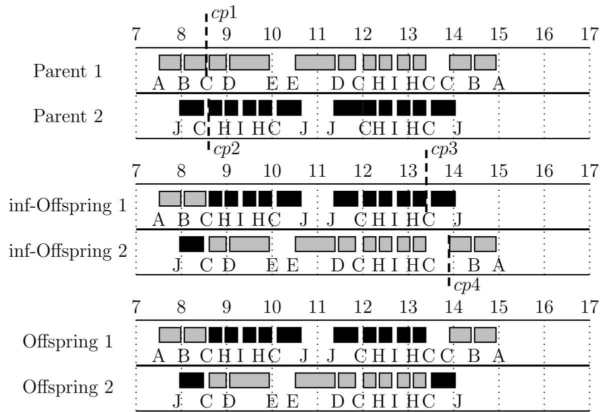


Figure 6: Two Point Crossover

Therefore, we implemented the TPC by calling the OPC twice. As shown in Figure 6, duties can be recombined with different crew bases in compliance with conditions $time(cp1) < time(cp3)$ and $time(cp2) < time(cp4)$ for the associated times of the cutting points $cp1$, $cp2$, $cp3$, and $cp4$. By using the TPC, an exhaustive exploration of the solution space is ensured. Preliminary tests show that calling OPC with a probability of 50 % and TPC with 30 % is a suitable setting. For the remaining 20 %, a mutation is done only on a randomly selected individual. Further, if the OPC was not successful, the TPC is then called. A mutation for a new individual created by a crossover happens with a probability of 10 %. The mutation operator itself replaces a randomly selected trip of a duty with another suitable one. We also tested roulette selection and tournament selection as variants for choosing individuals for recombination, but no improvements could be seen for both when compared to random selection.

5. Computational Analysis

5.1. Experimental Design

Real-world decision support is only guaranteed if a number of different crew-scheduling planners benefit from such a system. Therefore, the entire solution approach has been embedded in a client-server architecture. The generated schedules are directly transformable into action or can be used for a realistic evaluation of different scenarios. The algorithm itself was implemented in `C#`, and all tests were run on an Intel(R) Xenon(R) CPU E5-4627 with a 3.3 GHz clock speed and 768 GB RAM. RMP and rRMP were solved using Gurobi 7.5. Commonly, rRMP during CG is solved using a dual simplex algorithm. However, Gurobi also provides the barrier method (interior point method, e.g., Bixby et al. (1992)). Rousseau et al. (2007) show a clear improvement in computing time when using an interior point within column generation for a vehicle routing problem with time windows. The same is true for our instances of the OMCSPAR. Hence, the barrier method was employed in these tests. The maximum of parallel threads used by Gurobi was limited to four, whereas the GA was run on a single core. For each run, we limited the computation time to reasonable values for a tactical decision support system. Column generation was terminated after six hours, and solving RMP was limited to three hours. Because the GA is a probabilistic approach, each test was run 10 times. Table 3 summarizes all the parameter values used in the presented solution approach.

Table 3: Parameter Values

Costs	Initial Solution	Master Problem	Pricing Problem
$c^{\text{fix}} = 2{,}000$	$minD = 120\,\text{min}$	$maxAgeOfDuties = 7$	$P(\text{OPC}) = 0.5$
$c^{\text{var}} = 50$	$maxD = 360\,\text{min}$	$reducedCostThreshold = 1000$	$P(\text{TPC}) = 0.3$
$c^{\text{pen}} = 500{,}000$	$maxT = 120\,\text{min}$		$P(\text{Mutation only}) = 0.2$
$s = 500$	$maxS = 6$		$P(\text{Mutation additional}) = 0.1$
	$10 \leq Depth \leq 25$		

To demonstrate practical applicability, we consider 18 real-world instances with a planning horizon of two weeks. Table 4 provides the relevant data and the specific requirements for each network, with set B containing all relief points. Because

Table 4: Considered Networks

			# trips per attendance rates										Constraints			
n	$\lvert B\rvert$	$\lvert E\rvert$	0 %	10 %	25 %	30 %	50 %	67 %	90 %	100 %	150 %	od	(11)	(12)	(13)	(14)
I	18	10	972			7,560			1,304							
I*	18	10	972			7,560			1,304				●			
II	13	4	184		6,312					1,038						
II*	13	4	184		6,312					1,038			●			
III	15	4	300		6,396					1,566						
III*	15	4	300		6,396					1,566			●			
IV	11	5	156		3,794					4,326						
V	21	6	12,300	340						4,338		●				
VI	17	7	174		848		8,614			2,966				●		
VII	18	6	1,260					15,034				●				
VIII	12	5	716				5,292			1,528				●	●	●
IX	14	11	182				8,556			1,982	2,172		●		●	
X	43	10	1,044		13,114					7,704						
X*	43	10	1,044		13,114					7,704			●			
XI	77	13	1,628		34,208		1,076			3,350						
XII	3	2	68							768						
XIII	4	2	450							994						
XIV	8	3	256							1376						

Notation: n: network; $|B|$: # of relief point; $|E|$: # of crew bases; od: overlapping duties; (11): uniform distribution trips; (12): uniform distribution trains; (13): crew base capacity; (14): part-time employees

schedules are created at the tactical level, distinguishing requirements that characterize the instance itself, such as sets B, E, and M as well as the attendance rates, is essential. Based on our experience in hands-on cooperation with DB Regio AG, the requirements given on the right side of the table are commonly scenario dependent and can be assumed as changeable at the tactical level. Therefore the table shows both 14 networks and 18 instances. To restrict the amount of testing within reasonable limits, we have chosen this representative set of instances. Note that instances I, II, III, and X are each listed twice. Because Hoffmann et al. (2017) indicate that constraints (11) make solving considerably more difficult(an additional constraint for each trip), we consider all four networks with and without this requirement. A star (*) indicates that the instance requires uniform distribution for each trip. The table also includes three classic instances with only 100 % trips, which equals CSPs for train drivers (XII-XIV).

All instances are made publicly available at: `https://bit.ly/3dzlWvF`. We also provide a script, which contains the relevant requirements for the duty generation and can serve to validate generated schedules.

For an evaluation of the improvements of the column generation approach proposed in Section 4, using the approach of Hoffmann et al. (2017) as a benchmark is appropriate in Section 5.2.1. This is followed by the presentation of the results for all 14 networks in Section 5.2.2. Because the pricing problem is solved heuristically, we have no information regarding optimality gaps. However, by removing all limits used in Section 4.2 from the BG, we can generate all feasible duties for networks XII, XIII, and XIV in a reasonable amount of time. Subsequently, solving the unrestricted master problem (URMP) results in the optimal solution. Hence, we can compare the column generation approach to the optimal solution for these small instances. For larger instances, we use a productivity value ϕ (see, e.g., Gopalakrishnan and Johnson (2005); Jütte et al. (2011)), which is also commonly used in practice. This value is based on the ratio of protected working time and paid time, each of which is accumulated over all duties of the final schedule and given by

$$\phi = 1 - \frac{\text{cumulated paid time} - \text{cumulated protected working time}}{\text{cumulated paid time}}. \tag{19}$$

Nevertheless, it is merely an auxiliary value to obtain an idea of the solution quality because productivity is highly dependent on the network's characteristics. Therefore we use a reduced version of the arc flow formulation for the complete planning problem introduced by Hoffmann and Buscher (2019) expanded to a multi-periodic approach for generating valid lower bounds. This reduced version considers all requirements presented above, except the average paid time requirements (see RMP, constraints (9) and (10)), two rules for positioning breaks during a duty and the integrity constraints. These minor simplifications help to speed up the calculation of the bound significantly. A detailed description is available in the Appendix A. In the following we will refer to this as break relaxation (BR).

5.2. Evaluation and Comparison of Algorithms

5.2.1. Comparison with Hoffmann et al. (2017)

Table 5 summarizes the improvements made by the actual approach (A). All values are the averages of 10 runs. For each instance, two groups of columns exist: on the one hand, relevant values concerning the column generation steps are displayed (CG); on the other hand, key values for solving the RMP are given (RMP). The basic approach of Hoffmann et al. (2017) (H) is used as a basis for evaluating the gained improvements.

First, it should be noted that only the actual approach is able to solve all instances. For instances where a comparison is possible, this approach also provides better results. Only network II* is solved by A and H with almost the same quality.

Table 5: Comparison with Hoffmann et al. (2017)

network	approach	CG t	CG it	CG $\lvert N_{it}\rvert$	CG $\lvert\hat{N}_{it}\rvert$	RMP t	RMP OBJ	STD	δ	LB$^{\text{BR}}$	GAP$^{\text{BR}}$
I	H	4.1	846	256	256	2.9	4.861	0.25	0.0	4.358	10.15
	A	6.0	2,239	510	53	1.8	4.738	0.05	-2.3		8.04
I*	H	-	-	-	-	-	-	-	-	4.700	-
	A	6.0	1,846	826	75	3.0	5.121	0.09	$-\infty$		8.23
II	H	4.7	978	726	726	1.5	2.739	0.16	0.0	2.563	6.42
	A	6.0	2,887	500	160	0.4	2.719	0.17	-0.8		5.58
II*	H	6.0	12	727	727	3.0	3.258	2.35	0.0	2.688	17.49
	A	0.6	281	283	46	3.0	3.069	0.55	-5.8		12.14
III	H	-	-	-	-	-	-	-	-	3.160	-
	A	6.0	3,732	529	97	0.6	3.374	0.27	$-\infty$		6.35
III*	H	-	-	-	-	-	-	-	-	3.465	-
	A	6.0	1,812	723	144	3.0	3.758	0.32	$-\infty$		7.81

Notation: H: Hoffmann et al. (2017); A: Actual Approach; t: CPU time in h; it: # of iterations; $|N_{it}|$: total # of generated duties in thousand; $|\hat{N}_{it}|$: # of used duties in RMP in thousand; OBJ: objective function value in millions; STD: standard deviation in %; δ: rel. improvement of A compared to H; LB$^{\text{BR}}$: lower bound in millions generated by BR; GAP$^{\text{BR}}$: optimality gap in % based on LB$^{\text{BR}}$

The number of iterations is increased for A by removing columns (Section 4.1). This means that after seven iterations (*reducedCostThreshold*), the problem size decreases considerably ($|N_{it}| \gg |\hat{N}_{it}|$). If more iterations can be performed, the objective value decreases. Only network I is an exception because a much larger solution pool was created in the same time. Furthermore, computational effort was shifted from rRMP to GA, indicating that more time is used to explore the solution space (creating columns). This also suggests that the solution space is searched in a more structured manner because similar or better objective values are achieved. Furthermore, it is evident that the solution quality could be significantly improved, particularly for instances with uniform distribution (constraints (11)). An average gap of 8.09 % was achieved across all 6 instances. Note that the uniform distribution is a very weak constraint for BR. This explains the higher gaps when it is required. Considering the instance size and the fact that the lower bound is based on a relaxation, the solution quality can be assessed as very good.

Finally, it should be mentioned that the instances for this test were chosen in such a way that a comparison with the literature is possible. On the one hand, only requirements that were also taken into account are included (see Section 3.1). On the other hand, the size and complexity is sufficiently small that the algorithm of Hoffmann et al. (2017) has a chance to solve it (i.e. is at least able to generate a solution). A detailed analysis of the impact of the different proposed improvements of our column generation approach can be found in the Appendix.

5.2.2. Real-world Instances

Table 6 shows the results for Networks I–XI using the actual approach. On average, we achieve a productivity ϕ of 83.7% for all networks. In practice, $\phi > 80\%$ are assessed considerably positively by crew-scheduling planners. However, productivity does not represent an explicit measure for solution quality in each case. In particular, the minimum paid working time and the average paid time are input parameters that can distort the resulting values of ϕ. If these parameters are too high, long and unproductive duties might be generated to fit these values.

Table 6: Results for Considered Real-World Networks I–XI

	t^{CG}	t^{RMP}	OBJ	STD	LB^{BR}	GAP^{BR}	ϕ	1^{st}
I	6.0	1.8	4.738	0.05	4.358	8.04	89.1	
I*	6.0	3.0	5.121	0.09	4.700	8.23	87.5	•
II	6.0	4.2	2.718	0.17	2.563	5.68	88.3	
II*	0.6	3.0	3.069	0.55	2.688	12.41	83.2	
III	6.0	0.6	3.374	0.27	3.160	6.35	82.7	•
III*	6.0	3.0	3.758	0.32	3.465	7.81	79.4	•
IV	6.0	3.0	5.973	0.14	5.583	6.53	89.2	•
V	6.0	0.3	9.703	0.23	8.780	9.65	65.5	•
VI	6.0	3.0	10.099	0.72	8.830	12.61	88.8	•
VII	6.0	3.0	14.816	0.81	12.687	14.37	85.9	•
VIII	6.0	3.0	5.458	0.11	5.061	7.27	82.3	•
IX	6.0	3.0	14.087	5.83	12.168	13.62	88.7	•
X	6.0	3.0	19.693	0.70	-	-	79.1	•
X*	6.0	3.0	21.338	0.70	-	-	77.4	•
XI	6.0	3.0	12.205	0.61	-	-	89.1	•

Notation: t^{CG}: CPU time column generation in h; t^{RMP}: CPU time RMP in h; OBJ: objective function value in millions; STD: standard deviation in %; $\phi(\%)$: productivity of solution from eq. (19) in %; LB^{BR}: lower bound in millions generated by BR; GAP^{BR}: optimality gap in % based on LB^{BR}; 1^{st}: a (heuristic) solution is obtained for the first time

If overlapping duties are required, these can also lead to distortion of the productivity. Because only few trains run at night, avoiding longer interruptions by an efficient change of trains is not always possible. For example, both factors apply to network V. Nevertheless, high values of ϕ are indicators for good solutions. Additionally, for each run of all instances, the over-fulfillment of attendance rates was lower than 1 %, which also proves the high efficiency of the gained solutions. Within a time limit of 5 days and the use of up to 24 threads we were able to generate 12 of 15 valid lower bounds by solving BR for the instances shown in Table 6.[1] Again the gap is higher than 10% for three instances with uniform distribution (VI, VII and IX). However, very high productivity values ϕ are achieved for these instances, so that it can be guessed that the large gaps are caused by the rather weak lower bound.

Table 7 shows the results for the smaller networks XII–XIV. The actual approach was able to find the same solutions as those gained by solving the URMP for each instance, indicating that the optimal solution can be determined. If necessary, the column generation approach was limited to the time used of the exact approach. The gap based on BR (GAP^{BR}) is in a similar range compared to Instances I-XI. Therefore it can be expected that even for the very large instances results are obtained which are closer to the optimal solution than the gap suggests. Note, that the presented approach is also able to solve small instances to optimality with attendance rates less than 100% as well (see Section 6.1).

In conclusion, it can be said that the presented approach finds optimal solutions for small instances and performs very well for large instances, taking several practical requirements into account. All tests presented in this section required an accumulated net computing time of longer than two months.

[1] Note the very large problem sizes. BR(X): 48,127,462 variables; 6,610,369 constraints. BR(X*): 51,333,382 variables; 7,051,711 constraints. BR(XI): 49,835,222 variables; 6,489,146 constraints.

Table 7: Results for Considered Real-World Networks XII-XIV

	t	OBJ^{RMP}	LB^{URMP}	GAP^{URMP}	LB^{BR}	GAP^{BR}
XII	0.4	1.620	1.620	0.00	1.490	7.99
XIII	0.2	3.101	3.101	0.00	2.997	3.38
XIV	1.1	2.324	2.324	0.00	2.193	5.60

Notation: t: CPU time of the presented approach in h; OBJ^{RMP}: objective value after solving RMP in millions; LB^{URMP}: lower bound after solving URMP in millions; GAP^{URMP}: optimality gap based on LB^{URMP}; LB^{BR}: lower bound after solving BR in millions; GAP^{BR}: optimality gap based on LB^{BR}

6. Managerial insights for decision makers in the tender process

6.1. Cost effects of varying attendance rates

In this and in the following section, we discuss the effects that arise from the consideration of attendance rates. The objective is to provide better insights into the concept of attendance rates. Note that this section is interesting for different stakeholders, including not only railway companies but also principals (i.e., federal states or subsidiary transport associations). The latter defines the general conditions (including attendance rates and uniform distribution) for the tendering process. Thus, both sides can better estimate cost changes owing to modified conditions. First, we analyze the influence of attendance rates on the total costs of the final schedule.

For the analyses, we manipulate the attendance rates of instances I–III and XII–XIV. This manipulation is necessary because statements regarding the influence of attendance rates can only be made if the same network is solved with different rates. If several attendance rates occur, we unify them. This simplifies the interpretation of the results immensely.

For the first issue, we start with the small instances XII–XIV because we can solve these optimally. Figure 7 indicates the relation between costs and attendance rates for these networks. The horizontal axis indicates the attendance rates. The vertical axis shows the proportional costs in relation to the solution with attendance rates of 100% ($g = 1$).

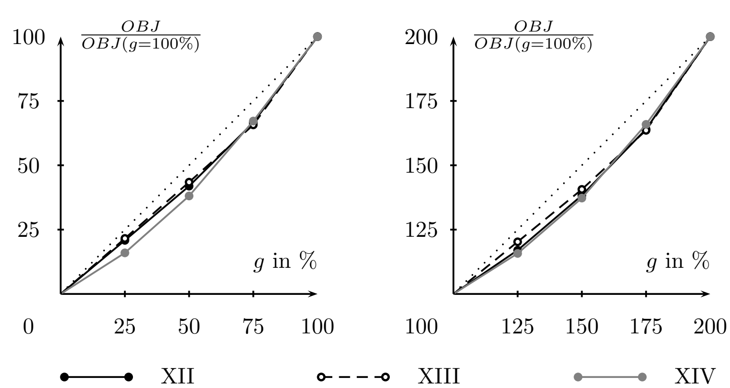


Figure 7: Progression of Objective Values with increasing Attendance Rates (small instances)

In the left graph, attendance rates less than 100 % clearly lead to disproportionate cost saving. For example, a halving of the attendance rate (100 % to 50 %) enables cost savings of more than 50 % because unproductive trips or trip combinations can be avoided. In other words, those duties that meet the required attendance rates at the lowest costs can be selected from the set of all possible duties.

In the right graph of Figure 7, this effect can also be clearly observed for attendance rates higher than 100 %. Thus, for example, a rate of 175 % leads to less than 175 % of the cost compared to the 100 % solution. In addition, the question arises as to whether the solutions can be added, i.e., for example, whether the schedule of the 125 % solution corresponds to the combination of the 100 % solution and 25 % solution. Intuitively, such a combination would be expected, but the results allow for other conclusions to be drawn. Because this is difficult to recognize in Figure 7, Figure 8 shows the results for the same test on networks I–III. The shape of the curves

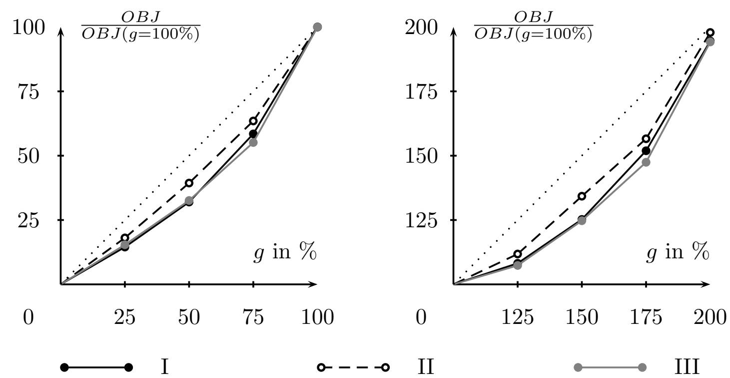


Figure 8: Progression of Objective Values with increasing Attendance Rates (large instances)

is analogous to Figure 7, but the bulge is more pronounced. Note that these are heuristic solutions. The differences between the two sides become much clearer for these instances. Based on the objective values, the 125 % solution is clearly not the addition of the 100 % and the 25 % solution. The final schedules show that this is due to deadheads. Figure 9 gives an illustrative example.

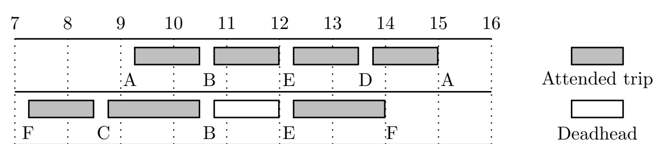


Figure 9: Deadhead Example for $g > 100\%$

Assuming the two duties shown are part of the 100 % solution, then the second one contains a deadhead. Note that it costs the same regardless of whether or not the third trip is a deadhead (paid time does not change). For a solution with a rate higher than 100 % we can change this trip from deadhead to attended trip within the same costs. Therefore, the value of the corresponding y-variable for this trip changes from one to two. However, the value of the left side of constraint (2) increases automatically ($\sum_{k\in K}\sum_{i\in M_k} d_{ig}y_{ik}$). Thus, for the 125 % example only less than 25 % of the kilometers must be additionally attended. Consequently, less than 25 % of additional costs are incurred.

Additional cost-saving potential results from the fact that with rates higher than 100 % duty combinations can also be chosen, which have mutually excluded themselves with 100 % because of the resulting deadheads.

6.2. Cost effects of less predictable schedules

In the second step, we conduct a more precise investigation on the influence of the two definitions of uniform distribution. In general, it should be noted that both variants only affect the solution at attendance rates less than 100 %. For rates greater than or equal to 100 % every trip is still attended.

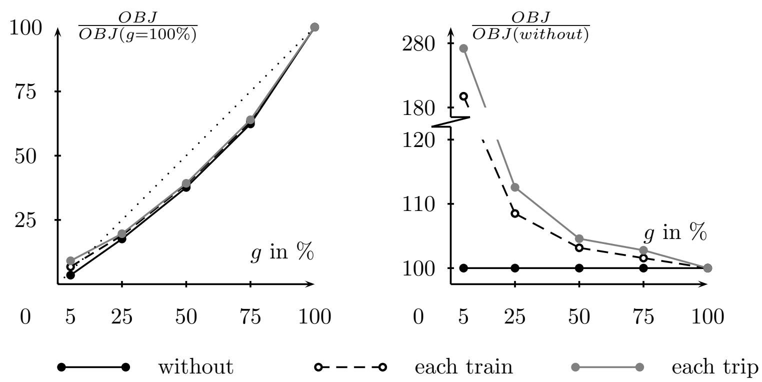


Figure 10: Progression of Objective Values depending on Uniform Distribution

For the analysis, we solved networks I–III and XII–XIV with and without uniform distributions for different attendance rates. Figure 10 shows the results. The illustration on the left side are similar to those in Figure 7. However, the values are not shown individually for each instance; instead, the average value was calculated. In contrast to the previous figures, no structural differences that depend on the instance size could be found here. Compared to the 100 % solution, the impact on costs seems small and only relevant for low rates. However, this presentation does not present the interrelations with sufficient clarity.

The right side shows the results in relation to the solution without uniform distribution but at the same rate. In the range between 25 % and 100 %, cost increases because of both types of uniform distribution are relatively moderate. Nevertheless, absolute values correspond to considerable additional costs and must not be ignored in practice. The lower the rate, the more extreme is the relative cost increase. The variant in which each trip must be attended at least once always creates more costs than the train-based rule. For example, one train contains an average of 2.7 trips for the networks I–III. The trip-based rule forces each trip to be attended. For the train-based rule, only one of each train is sufficient. The differences between both variants increase with decreasing rates.

At rates of 5 %, the additional costs correspond to almost the same (train) or double (trip) the original costs. At rates of 0 %, the optimal solution without uniform distribution is an empty schedule (no constraint requires an attended trip). Therefore, the cost increase caused by uniform distribution is infinite.

Finally, in addition to the cost increases, uniform distribution represents a considerable challenge for planners in practice. In an appropriate form, this can only be dealt with through automated planning support, as is possible with the approach presented.

7. Conclusions and Further Research

In this paper, we presented a highly sophisticated column generation approach for solving multi-period CSPs, which is integrated into a running software and used by DB Regio AG in practice. Further, we focused on the integration of several necessary real-world requirements. To the best of our knowledge, these conditions have been presented for the first time.

Moreover, the algorithm itself was accelerated by several adjustments. A holistic consideration of the complete algorithm enabled us to achieve a better solution quality within reasonable computation times at the tactical planning level. In addition, we were able to solve some instances for the first time.

In the context of column generation, we also considered aspects in detail that have rarely been discussed in the literature to date, such as creating an initial solution, choosing a suitable setting for solving the rRMP, and distinct optimality gaps. The proposed algorithm was exemplarily proven to be able to solve 24 real-world problems in regional rail transport and is used successfully in practice. Additionally, small instances were proven to be optimally solvable.

Finally, we provided valuable managerial insights into the mode of action of attendance rates. Disproportionate cost savings were shown to be achievable with a smaller attendance rate.

Nevertheless, several interesting directions remain for future research. First, assessing the quality of the solution in terms of optimality for large instances would be worthwhile. Because introducing and determining lower bounds (for large instances) end in a complex optimization problem itself, implementing an exact approach for solving the pricing problem is necessary. By disregarding the used time limits, an optimal solution may be obtained through this exact approach if the GA does not create new duties anymore. Clearly, this approach would be considerably time consuming and only for scientific interest.

For practical applications, the identification of a better termination criterion for the column generation could be helpful. Specifically, convergence-based criteria seem suitable. Furthermore, a detailed discussion on solving the RMP must be carried out. In particular, heuristic solution approaches have to be investigated. Finally, the proposed algorithm should be tested for larger networks or for a combination of several networks. If necessary, integration into a decomposition approach is possible.

Acknowledgment

This research was supported by DB Regio AG. The authors would like to thank Volker Jacobsen and Detlef Jacobs for the successful cooperation as well as the anonymous reviewers for their valuable comments and suggestions for improving the quality of the paper.

Appendix A. Reduced multi periodic arc flow formulation

Generating a lower bound for crew scheduling problems is a very hard optimization problem itself. Since the GA still generates new duties even after several days of computing time in column generation, it seems to be impossible to reach a regular end and get a lower bound this way. Therefore we solve a relaxation of the complete problem modeled as multi-periodic arc flow formulation. The formulation is adapted from Hoffmann and Buscher (2019) and extended to the multi-periodic approach by generating a graph for each day of the planning horizon. Figure A.11 shows the graphs for two consecutive days enabling overlapping duties. The trip arcs between the gray marked nodes A and C represent the same trip which can be covered in both graphs (i.e. by both days; black: previous day; gray: next day). Beside this we refer to Hoffmann and Buscher (2019) for a detailed explanation of the graph.

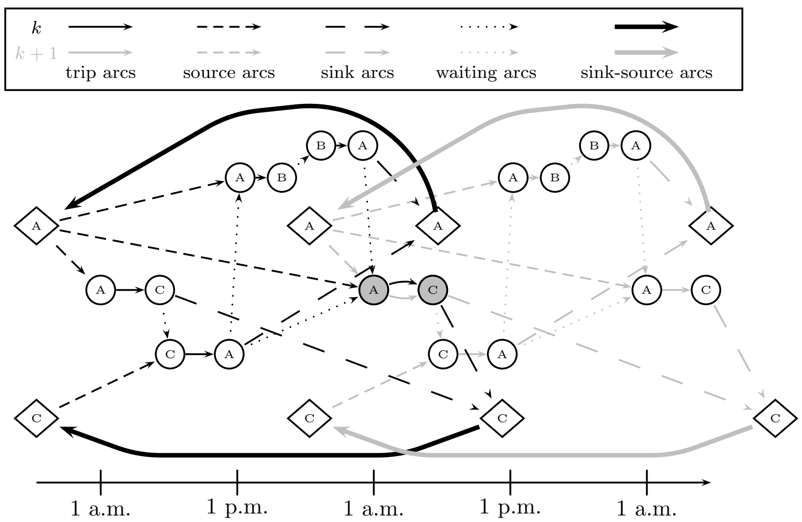


Figure A.11: Example graph with trip, source, sink, waiting and sink-source arcs

Further, Table A.8 shows the used notation. Again this is very similar to Hoffmann and Buscher (2019). For generating lower bounds we can omit the node-related resources.

Based on this notation we introduce a relaxation for the complete planning problem given by (A.1)–(A.21). As mentioned in Section 5.1 this corresponds to a reduced formulation of Hoffmann and Buscher (2019) expanded to a multi-periodic approach. Because of the strong similarity the following description is very briefly.

$$[\text{BR}]: \quad \min \sum_{k \in K} \left(c^{\text{var}} \cdot \sum_{c \in C} pt_{ck} + c^{\text{fix}} \cdot \sum_{c \in C} \sum_{q \in Q_k} \sum_{j \in V_k} x_{qjck} \right) \tag{A.1}$$

$$\text{s.t.} \quad pt_{ck} \geq \sum_{(i,j)\in A_k} t_{ij} x_{ijck} - (30u_{ck} + 15v_{ck}) \qquad \forall k \in K, c \in C \tag{A.2}$$

$$pt_{ck} \geq t^{\min} \cdot \sum_{q\in Q_k} \sum_{j\in V_k} x_{qjck} \qquad \forall k \in K, c \in C \tag{A.3}$$

$$\sum_{k\in K} \sum_{(i,j)\in F_k} d_{ijgk} y_{ijk} \geq g \sum_{k\in K} \sum_{(i,j)\in F_k} d_{ijgk} \qquad \forall g \in D \tag{A.4}$$

$$\sum_{c\in C} x_{ijck} \geq y_{ijk} \qquad \forall k \in K, (i,j) \in F_k \tag{A.5}$$

$$y_{ijk} \geq x_{ijck} \qquad \forall k \in K, (i,j) \in F_k, c \in C \tag{A.6}$$

$$\sum_{h\in V_k:(h,i)\in A_k} x_{hick} - \sum_{j\in V_k:(i,j)\in A_k} x_{ijck} = 0 \qquad \forall k \in K, i \in V_k, c \in C \tag{A.7}$$

$$\sum_{(i,j)\in R_k} x_{ijck} \leq 1 \qquad \forall k \in K, c \in C \tag{A.8}$$

$$\sum_{(i,j)\in A_k} t_{ijk} x_{ijck} \leq t^{\max} \qquad \forall k \in K, c \in C \tag{A.9}$$

$$\sum_{(i,j)\in A_k} s_{ijk} x_{ijck} \leq s^{\max} \qquad \forall k \in K, c \in C \tag{A.10}$$

$$\sum_{(i,j)\in A_k} s_{ijk} x_{ijck} \geq 361 \cdot u_{ck} \qquad \forall k \in K, c \in C \tag{A.11}$$

$$\sum_{(i,j)\in A_k} s_{ijk} x_{ijck} - (s^{\max} - 360)u_{ck} \leq 360 \qquad \forall k \in K, c \in C \tag{A.12}$$

$$\sum_{(i,j)\in A_k} s_{ijk} x_{ijck} \geq 541 \cdot v_{ck} \qquad \forall k \in K, c \in C \tag{A.13}$$

$$\sum_{(i,j)\in A_k \in K,} s_{ijk} x_{ijck} - (s^{\max} - 540)v_{ck} \leq 540 \qquad \forall k \in K, c \in C \tag{A.14}$$

$$\sum_{(i,j)\in A_k} b_{ijk}x_{ijck} \geq 30 \cdot u_{ck} \qquad \forall k \in K, c \in C \tag{A.15}$$

$$\sum_{(i,j)\in A_k} b_{ijk}x_{ijck} \geq 45 \cdot v_{ck} \qquad \forall k \in K, c \in C \tag{A.16}$$

$$x_{ijck} \in \{0,1\} \qquad \forall k \in K, (i,j) \in A_k, c \in C \tag{A.17}$$

$$y_{ijk} \in \{0,1\} \qquad \forall k \in K, (i,j) \in F_k \tag{A.18}$$

$$pt_{ck} \in \mathbb{R}^+ \qquad \forall k \in K, c \in C \tag{A.19}$$

$$u_{ck} \in \{0,1\} \qquad \forall k \in K, c \in C \tag{A.20}$$

$$v_{ck} \in \{0,1\} \qquad \forall k \in K, c \in C. \tag{A.21}$$

Objective (A.1) minimizes the total costs of all duties over all days of the planning horizon. The paid time pt_{ck} for each duty is calculated by constraint (A.2). To avoid very short duties constraints (A.3) set a lower bound for pt_{ck}. Constraints (A.4) ensure the coverage of the attendance rates. Constraints (A.5) and (A.6) are linking constraints for variables x_{ijck} and y_{ijk}. The flow conservation for each duty is given by constraints (A.7). Constraints (A.8) ensure that each conductor returns to the crew base only once. The duty time and the protected working time are restricted to the given limits by constraints (A.9) and (A.10). Constraints (A.11) set variable u_{ck} to zero if no break is required. Constraints (A.12) cause the opposite if the protected working time is bigger than 6 hours. Constraints (A.13) and (A.14) are used for a 45 minute break analogously. If a break is required, constraints (A.15) and (A.16) ensure that enough time is available. Finally constraints (A.17)–(A.21) state the domains. Note for generating a lower bound the binary constraints are relaxed and we solve the linear program only.

The average paid time (see constraints (9) and (10)) and two positioning rules for breaks during a duty are not considered in this formulation. The first prohibits breaks within the first and last two hours of a duty. The second requires a break after no more than 6 hours of protected working without a break. Both are modeled by Hoffmann and Buscher (2019) in detail. The average paid time constraints link all x variables in two constraints. The positioning rules require the tracking of accumulated resources variables across all nodes of the graph for each duty. Since both makes solving of the arc flow formulation considerably more difficult we relax these. The resulting lower bounds are valid because not considering them leads to a decrease of the bound (i.e., the minimum required costs are underestimated). Note, that the consideration of the positioning rules are not mentioned explicitly in Section 3, because it is only a additional feasibility check in the GA without novelty. Nevertheless both are considered during the hybrid solution approach.

The optional constraints for both types of uniform distribution are given by constraints (A.22) and (A.23).

$$\sum_{(i,j,k)\in T_m} y_{ijk} \geq 1 \qquad \forall m \in M \tag{A.22}$$

$$\sum_{(i,j,k)\in T_z} y_{ijk} \geq 1 \qquad \forall z \in Z \tag{A.23}$$

All other optional constraints described in Section 3.2 (e.g. crew base capacity) are not considered for generating lower bounds. Obviously a consideration in future approaches would further improve it. Some networks require a minimum break time of 30 minutes without interruptions. Again this is not mentioned explicitly in Section 3, because it is only a additional feasibility check in the GA without novelty. Nevertheless it is considered during the hybrid solution approach. For generating a lower bound this can be modeled by constraints (A.24).

$$\sum_{(i,j,k)\in A_k : b_{ijk} \geq 30} x_{ijck} \geq u_{ck} \qquad \forall \in M \tag{A.24}$$

Finally we adapt three valid inequalities for the multi periodic formulation: symmetrie breaking constraints, prohibiting the use of parallel arcs and preassigning 100% trips to conductors (Constraints (47), (49) and (50) in Hoffmann and Buscher (2019)).

Table A.8: Sets and parameters

Sets		Parameters	
K	periods (days)	d_{ijgk}	distance of trip arc (i,j) with rate g on day k
M	trips		
T_m	trip arcs of all days of trip m	t_{ijk}	duty time of arc (i,j) on day k
T_z	trip arcs of all days of train z	s_{ijk}	protected working time of arc (i,j) on day k
V_k	nodes on day k		
Q_k	sources on day k	b_{ijk}	possible break time of arc (i,j) on day k
S_k	sinks on day k		
A_k	arcs on day k	$t^{\min}$	minimum paid time
F_k	trip arcs on day k	$t^{\max}$	maximum duty time
R_k	sink-source arcs on day k	$s^{\max}$	maximum protected working time
C	set of conductors	c^{fix}	fixed costs per duty
D	set of attendance rates	c^{var}	variable costs per minute

Decision variables

$$x_{ijck} = \begin{cases} 1, & \text{if conductor } c \text{ uses arc } (i,j) \text{ on day } k, \\ 0, & \text{otherwise} \end{cases}$$

$$y_{ijk} = \begin{cases} 1, & \text{if trip arc } (i,j) \text{ is in solution on day } k, \\ 0, & \text{otherwise} \end{cases}$$

$$u_{ck} = \begin{cases} 1, & \text{if protected working time of conductor } c \text{ is} > 360 \text{ on day } k, \\ 0, & \text{otherwise} \end{cases}$$

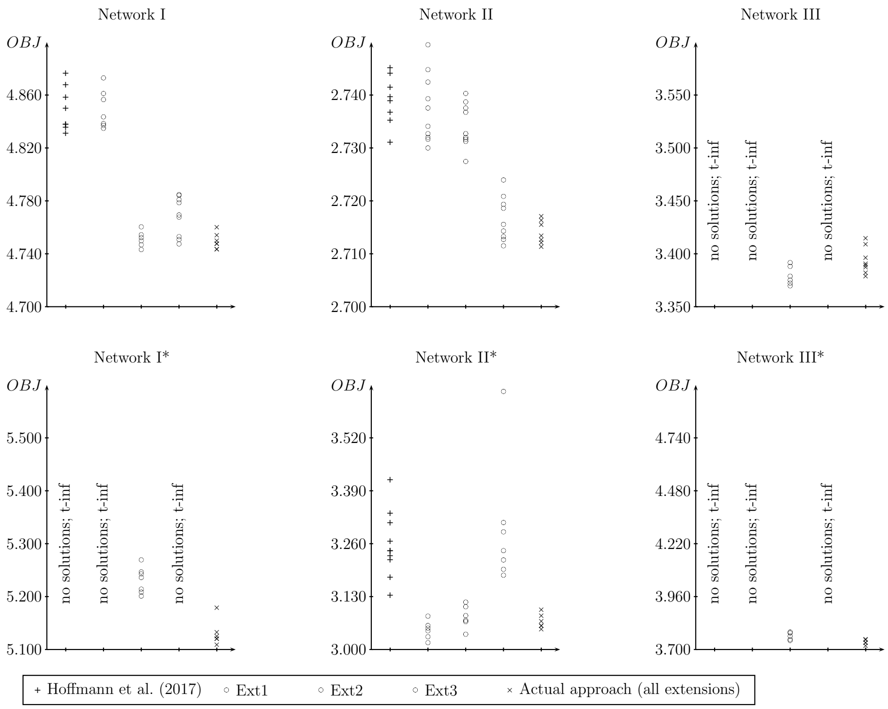
$$v_{ck} = \begin{cases} 1, & \text{if protected working time of conductor } c \text{ is} > 540 \text{ on day } k, \\ 0, & \text{otherwise} \end{cases}$$

pt_{ck} paid time for conductor c on day k

Appendix B. Evaluation of Improvements of the Solution Approach

Figure B.12 illustrates the results for testing the extensions of our column generation approach separately from each other. Setting Ext1 represents the basic approach of Hoffmann et al. (2017) extended by the general adjustments of the column generation framework only (see Section 4.1). Settings Ext2 and Ext3 represent this approach with the adjustments for creating an initial solution (see Section 4.2) and solving the pricing problem (see Section 4.3). We also consider the proposed approach (a combination of all extensions). The figure shows extension-wise resulting objective values of 10 runs for each instance.

The algorithm of Hoffmann et al. (2017) as well as the extensions Ext1 and Ext3 are not able to generate feasible initial solutions for all networks. For networks I* and III*, not all trips could be scheduled in blocks to meet constraints (11). For network III, this applies for constraints (2) with $g = 100\%$. These results are marked with t-inf. This clarifies that the initial solution procedure (Ext2) is essential to be able to solve real-world networks. In addition, this improves the solution quality for those instances where a comparison is possible (I, II, II*). The extensions of Section 4.1 (Ext1) seem most effective for solving instances requiring uniform distribution, see e.g. network II*. Although at this point the statement is only supported by one



Notation: OBJ: objective function value in millions; t-inf: trip infeasibility, see Section 4.2.1
Note: An outlier was not displayed for Ext1 on instance III* ($OBJ = 9.8$).

Figure B.12: Comparison of Improvements

instance, we were able to observe this effect for many real-world instances during the cooperation with DB Regio AG. These adjustments speed up the process of solving the rRMP, whereby a significant higher number of iterations can be achieved. This results in lower objective values for instances with uniform distribution. In contrast, only convergence is accelerated for instances without uniform distribution. Although the GA improves the solution for several instances significantly (Ext3), it does not seem to be a stand-alone improvement. This is because the performance of the GA depends on the quality of the initial solution.

Table B.9 shows the average computing times for each algorithm. It should be noted that faster computing times can only be observed with the combined consideration of all three extensions. This applies to both column generation and solving the RMP. The individual extensions accelerate the solution only for individual instances. In general, the combination of all three extensions achieves the best results for all networks. At this point, it can be observed that the improvements work very well.

Table B.9: Average Computing Times - Algorithm Extensions

	H		Ext1		Ext2		Ext3		A	
	CPU CG	CPU RMP	CPU CG	CPU RMP	CPU CG	CPU RMP	CPU CG	CPU RMP	CPU CG	CPU RMP
I	2.6	1.9	2.2	1.8	2.3	0.6	3.7	2.5	2.3	0.8
I*	-	-	-	-	6.0	3.0	-	-	6.0	3.0
II	4.7	1.5	4.9	1.0	2.6	1.0	5.5	0.9	4.6	0.1
II*	6.5	3.0	3.4	3.0	2.2	3.0	6.3	3.0	0.6	2.8
III	-	-	-	-	6.0	0.2	-	-	3.9	0.0
III*	-	-	-	-	6.0	3.0	-	-	6.0	3.0

Notation: $\frac{\text{CPU}}{\text{CG}}$: CPU time column generation in hours; $\frac{\text{CPU}}{\text{RMP}}$: CPU time integer RMP in hours; H: Hoffmann et al. (2017); A: Actual Approach.

References

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

Albers, M., 2009. Freight Railway Crew Scheduling: Models, Methods, and Applications. Logos Verlag Berlin GmbH.

Alefragis, P., Goumopoulos, C., Housos, E., Sanders, P., Takkula, T., Wedelin, D., 1998. Parallel crew scheduling in paros, in: European Conference on Parallel Processing, Springer. pp. 1104–1113.

Arabeyre, J., Fearnley, J., Steiger, F., Teather, W., 1969. The airline crew scheduling problem: A survey. Transportation Science 3, 140–163.

Bach, L., Dollevoet, T., Huisman, D., 2016. Integrating timetabling and crew scheduling at a freight railway operator. Transportation Science 50, 878–891.

Barnhart, C., Cohn, A.M., Johnson, E.L., Klabjan, D., Nemhauser, G.L., Vance, P.H., 2003. Airline crew scheduling, in: Hall, R.W. (Ed.), Handbook of Transportation Science. Springer US. volume 56 of *International Series in Operations Research & Management Science*. chapter 14, pp. 517–560.

Bengtsson, L., Galia, R., Gustafsson, T., Hjorring, C., Kohl, N., 2007. Railway crew pairing optimization, in: Geraets, F., Kroon, L., Schoebel, A., Wagner, D., Zaroliagis, C.D. (Eds.), Algorithmic Methods for Railway Optimization, Springer Berlin Heidelberg. pp. 126–144.

Van den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E., De Boeck, L., 2013. Personnel scheduling: A literature review. European Journal of Operational Research 226, 367–385.

Bixby, R.E., Gregory, J.W., Lustig, I.J., Marsten, R.E., Shanno, D.F., 1992. Very large-scale linear programming: A case study in combining interior point and simplex methods. Operations Research 40, 885–897.

Caprara, A., Fischetti, M., Toth, P., Vigo, D., Guida, P.L., 1997. Algorithms for railway crew management. Mathematical programming 79, 125–141.

Caprara, A., Kroon, L., Monaci, M., Peeters, M., Toth, P., 2007. Passenger railway optimization, in: Barnhart, C., Laporte, G. (Eds.), Transportation. Elsevier. volume 14 of *Handbooks in Operations Research and Management Science*. chapter 3, pp. 129–187.

Caprara, A., Monaci, M., Toth, P., 2001. A global method for crew planning in railway applications, in: Computer-Aided Scheduling of Public Transport. Springer, pp. 17–36.

Carraresi, P., Gallo, G., 1984. Network models for vehicle and crew scheduling. European Journal of Operational Research 16, 139–151.

Chen, S., Shen, Y., 2013. An improved column generation algorithm for crew scheduling problems. Journal of Information & Computational Science 10, 175–183.

Dauzère-Pérès, S., De Almeida, D., Guyon, O., Benhizia, F., 2015. A lagrangian heuristic framework for a real-life integrated planning problem of railway transportation resources. Transportation Research Part B: Methodological 74, 138–150.

Desrosiers, J., Lübbecke, M.E., 2011. Branch-price-and-cut algorithms. Wiley Encyclopedia of Operations Research and Management Science .

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

Ernst, A.T., Jiang, H., Krishnamoorthy, M., Nott, H., Sier, D., 2001. An integrated optimization model for train crew management. Annals of Operations Research 108, 211–224.

Fuentes, M., Cadarso, L., Marín, Á., 2019. A hybrid model for crew scheduling in rail rapid transit networks. Transportation Research Part B: Methodological 125, 248–265.

Gopalakrishnan, B., Johnson, E.L., 2005. Airline crew scheduling: State-of-the-art. Annals of Operations Research 140, 305–337.

Goumopoulos, C., Housos, E., 2004. Efficient trip generation with a rule modeling system for crew scheduling problems. Journal of Systems and Software 69, 43–56.

Heil, J., Hoffmann, K., Buscher, U., 2020. Railway crew scheduling: models, methods and applications. European Journal of Operational Research 283, 405–425.

Hoffmann, K., Buscher, U., 2019. Valid inequalities for the arc flow formulation of the railway crew scheduling problem with attendance rates. Computers & Industrial Engineering 127, 1143–1152.

Hoffmann, K., Buscher, U., Neufeld, J.S., Tamke, F., 2017. Solving practical railway crew scheduling problems with attendance rates. Business & Information Systems Engineering 59, 147–159.

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

Ibarra-Rojas, O., Delgado, F., Giesen, R., Muñoz, J., 2015. Planning, operation, and control of bus transport systems: A literature review. Transportation Research Part B: Methodological 77, 38–75.

Irnich, S., Desaulniers, G., 2005. Shortest path problems with resource constraints, in: Desaulniers, G., Desrosiers, J., Solomon, M.M. (Eds.), Column generation. Springer, pp. 33–65.

Janacek, J., Kohani, M., Koniorczyk, M., Marton, P., 2017. Optimization of periodic crew schedules with application of column generation method. Transportation Research Part C: Emerging Technologies 83, 165–178.

Joncour, C., Michel, S., Sadykov, R., Sverdlov, D., Vanderbeck, F., 2010. Column generation based primal heuristics. Electronic Notes in Discrete Mathematics 36, 695–702.

Jütte, S., Albers, M., Thonemann, U.W., Haase, K., 2011. Optimizing railway crew scheduling at db schenker. Interfaces 41, 109–122.

Jütte, S., Thonemann, U.W., 2012. Divide-and-price: A decomposition algorithm for solving large railway crew scheduling problems. European Journal of Operational Research 219, 214–223.

Jütte, S., Thonemann, U.W., 2015. A graph partitioning strategy for solving large-scale crew scheduling problems. OR Spectrum 37, 137–170.

Kasirzadeh, A., Saddoune, M., Soumis, F., 2017. Airline crew scheduling: models, algorithms, and data sets. EURO Journal on Transportation and Logistics 6, 111–137.

Koniorczyk, M., Talas, B., Gedeon, F., 2015. Preconditioning in the backtracking duty generation of passenger rail crew scheduling: A case study. Communications - Scientific Letters of the University of Zilina 17, 23–29.

Kwan, R.S., 2011. Case studies of successful train crew scheduling optimisation. Journal of Scheduling 14, 423–434.

Liu, M., Haghani, A., Toobaie, S., 2010. Genetic algorithm-based column generation approach to passenger rail crew scheduling. Transportation Research Record: Journal of the Transportation Research Board 2159, 36–43.

Mourgaya, M., Vanderbeck, F., 2007. Column generation based heuristic for tactical planning in multi-period vehicle routing. European Journal of Operational Research 183, 1028–1041.

Narayanaswami, S., Rangaraj, N., 2011. Scheduling and rescheduling of railway operations: A review and expository analysis. Technology Operation Management 2, 102–122.

Neufeld, J.S., 2019. Efficient railway crew scheduling in german regional passenger transport. Impact 2019, 7–10. doi:10.1080/2058802X.2019.1664072.

Nishi, T., Muroi, Y., Inuiguchi, M., 2011. Column generation with dual inequalities for railway crew scheduling problems. Public Transport 3, 25–42.

Rousseau, L.M., Gendreau, M., Feillet, D., 2007. Interior point stabilization for column generation. Operations Research Letters 35, 660 – 668.

Şahin, G., Yüceoğlu, B., 2011. Tactical crew planning in railways. Transportation Research Part E: Logistics and Transportation Review 47, 1221–1243.

Shen, Y., Chen, S., 2014. A column generation algorithm for crew scheduling with multiple additional constraints. Pacific Journal of Optimization 10, 113–136.

Shen, Y., Peng, K., Chen, K., Li, J., 2013. Evolutionary crew scheduling with adaptive chromosomes. Transportation Research Part B: Methodological 56, 174–185.

Steinzen, I., Suhl, L., Kliewer, N., 2009. Branching strategies to improve regularity of crew schedules in ex-urban public transit. OR Spectrum 31, 727–743.

Suyabatmaz, A., Şahin, G., 2015. Railway crew capacity planning problem with connectivity of schedules. Transportation Research Part E: Logistics and Transportation Review 84, 88–100.

Teodorović, D., Janić, M., 2017. Public transportation systems, in: Teodorović, D., Janić, M. (Eds.), Transportation Engineering. Butterworth-Heinemann, pp. 387–493.

Vaidyanathan, B., Jha, K.C., Ahuja, R.K., 2007. Multicommodity network flow approach to the railroad crew-scheduling problem. IBM Journal of Research and Development - Business optimization 51, 325–344.

Veelenturf, L.P., Potthoff, D., Huisman, D., Kroon, L.G., Maróti, G., Wagelmans, A.P., 2014. A quasi-robust optimization approach for crew rescheduling. Transportation Science 50, 204–215.

Yaghini, M., Karimi, M., Rahbar, M., 2015. A set covering approach for multi-depot train driver scheduling. Journal of Combinatorial Optimization 29, 636–654.