Two-Stage Weekly Shift Scheduling for Train Dispatchers

Tomas Lidén
Department of Science and Technology (ITN), Linköping University, Sweden

Christiane Schmidt
Department of Science and Technology (ITN), Linköping University, Sweden

Rabii Zahir
Department of Science and Technology (ITN), Linköping University, Sweden

Abstract

We consider the problem of creating weekly shift schedules for train dispatchers, which conform to a variety of operational constraints, in particular, several work and rest time restrictions. We create the schedules in a two-stage process. First, using a previously presented IP model, we create a set of feasible daily shifts, which takes care of minimum-rest and shift-length requirements, taskload bounds, and combinability of dispatching areas. We then formulate an IP model to combine these daily shifts into weekly schedules, enforcing that each daily shift is covered by some dispatcher every day of the week, while ensuring that the weekly schedules comply with various restrictions on working hours from a union agreement. With this approach, we aim to identify "good" sets of daily shifts for the longer schedules. We run experiments for real-world sized input and consider different distributions of the daily shifts w.r.t. shift length and ratio of night shifts. Daily shifts with shift-length variability, relatively few long shifts, and a low ratio of night shifts generally yield better weekly schedules. The runtime for the second stage with the best daily-shift pattern is below three hours, which – together with the runtime for stage 1 of ca. 2 hours per run – can be feasible for real-world use.

2012 ACM Subject Classification Applied computing → Transportation

Keywords and phrases shift scheduling, IP, train dispatcher shift scheduling

Digital Object Identifier 10.4230/OASIcs.ATMOS.2024.6

Funding This work is part of the Trafikverket-funded project Capacity Modeling and Shift Optimization for Train Dispatchers (CAPMO-Train) (TRV 2020/99267). Partially funded by the Swedish Research Council through grant agreement no. 2022-06725.

Acknowledgements The computations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) at Linköping university (the Tetralith server). We would like to thank Magnus C. Johansson, Carl-Johan Wallentin, and Sebastian Olred from Trafikverket, TC Malmö, for their time and expertise in mapping the constraints on train-dispatcher shifts. Finally, we thank the reviewers of this paper for their valuable feedback.

1 Introduction

Train dispatchers are responsible for assigning tracks, for routing trains safely and efficiently, and for guaranteeing the safety of staff working close to the tracks [18]. A good performance in their dispatching work is crucial for safe and efficient operations. This becomes even more important with increasing traffic volumes: railway passenger traffic had increased pre-pandemic and has again caught up with these volumes [9].

In Sweden, the Swedish Transport Administration (Trafikverket) manages ca. 90% of the ∼15.600-km track network [21]. For this task, the network is divided into eight dispatching centers, each of which is further partitioned into geographical areas. Such a geographical area must be controlled by a dispatcher, but when the traffic situation allows – in particular, with a low enough traffic volume – the working positions of several dispatchers, and with that the geographical areas, can be combined to fewer working positions. However, the dispatchers will need an endorsement for each area they control.

Apart from these operational requirements, dispatcher shifts must comply with a variety of restrictions from a union agreement [20]: limits on the length of a single shift, limits on the rest periods between consecutive shifts, limits on the weekly working hours (depending on whether and how much a dispatcher works during nights and weekends). Currently, shift scheduling for train dispatchers is a manual process and – up to our work – it had obtained little attention by researchers, despite its complexity. In previous work, we [16, 15, 14] presented integer programming (IP) models to compute daily train-dispatcher shifts, which are feasible w.r.t. all shift-specific legal and operational requirements and for which the dispatcher task load is limited.

While the daily shift plans are a good start, the planning horizon for dispatchers is longer. When moving from planning for a single day to planning for longer periods, for starters to a full week, a variety of relatively complicated restrictions from the union agreement [20] for Trafikverket come into play:

- The weekly rest in a seven-days period should include at least 36 hours of contiguous rest.
- In the average of the considered period (= the calendar year), ordinary working hours (refer to the standard weekly hours a dispatcher is expected to work excluding overtime), may not exceed:
  - On average 40 hours per week without holidays.
  - On average 38 hours per week for employees who have ordinary working hours before 06AM or after 8PM on average *one* time per week *or* ordinary working hours on a Saturday, Sunday, or holiday.
  - On average 36 hours per week for employees who have ordinary working hours before 6AM or after 8PM on average *two* times per week *and* ordinary working hours occur on Saturdays, Sundays, or holidays.
  - On average 36 hours per week in case of ordinary working hours which on average per employee occur 1 time/week *and* on average 2 hours/week and with on average 2 hours/week between 11PM and 5AM *and* ordinary working hours occur on Saturdays, Sundays, or holidays.
  - On average 34 hours and 15 minutes per week if ordinary working hours on average per employee occur with at least 1.4 times/week between 11PM and 5AM *and* at least once on average 2 hours between 11PM and 5AM *and* ordinary working hours occur on Saturdays, Sundays, and holidays.

We denote this list of restrictions by $\mathcal{R}$. Since the planned horizon considered is one week, resulting in a cyclic schedule, we use the total working hours to track the averages presented in $\mathcal{R}$.

Modeling all the constraints on weekly schedules mathematically incurs quite a high complexity, and we did not want to add these to the functionality of our models for daily shifts [16, 15, 14], as we do not expect to obtain solutions even with very high limits on the computation times. Hence, for this paper, we instead use the output of the optimization model for one-day shifts as puzzle bits that we aim to combine as good as possible for feasible weekly schedules. More precisely, we take a set of feasible one-day shifts as our puzzle bits and enforce that all of these are used every day – in contrast to using a larger set of possible shifts (puzzle bits), where our mathematical model would have to ensure that with the selected shifts each area is monitored by a dispatcher during all periods of each day (which would yield a column-generation approach). We use this approach, as we do not solely want to compute the optimal puzzle bits, but we aim to identify good puzzle bits (daily shifts) and how to create them. Thus, in our mathematical model for the weekly schedules, we do not need to take care of aspects covered in the daily shifts: dispatchers are assigned one or several combinable areas, for which they hold an endorsement; each dispatching area is assigned a dispatcher during every period of the day; each dispatcher shift complies with upper and lower bounds on the shift length; and the task load for each dispatcher does not exceed an upper bound in any time period. Of course with this approach we will likely not reach the global optimum for the weekly shifts. The runtime for stage one in our approach (using the previous models to compute daily shifts), even if we minimize the dispatcher-area-assignment switches (an operationally desirable property), is less than two hours.

In this paper, we minimize the number of dispatchers scheduled for work during the week. The motivation for this is twofold: the labor turnover for dispatchers is relatively high (possibly because of the partly undesirable working hours) and from the operational side it is interesting to know the minimal number of dispatchers that are needed to cover the existing traffic to be able to size the workforce; for us, the objective yields a baseline to which we can compare results for other possible objectives in later stages. This is in line with our work for daily shifts, where we first aimed to minimize the number of dispatchers and in later stages aimed to minimize the number of dispatcher-area assignment changes while keeping the number of dispatchers to the minimum.

Related Work. Shift-scheduling problems are typically different for each type of profession, and even for each specific case study, making them difficult to solve by standard models [19]. However, some tentative approaches for a generic shift scheduling were presented in [1, 19]. These approaches categorize the problems' constraints into hard ones, such as staffing and competence constraints, and soft ones (called horizontal) consisting of counter constraints (e.g., number of days off), series constraints (e.g., consecutive worked nights) and their combination (e.g., number of resting hours between consecutive shifts). The objective function contains a weighted sum of each of the violated horizontal constraints. In real life, where scheduling is done manually, some horizontal constraints may be violated [8]. In our paper, in contrast to [1, 19], we do not allow constraints to be violated, though we do have horizontal constraints, e.g., on consecutive hours off. Several solution approaches were proposed in the literature, which are based on exact methods, such as linear programming [4], constraint programming [13] and column generation [6, 7, 10, 12], or on heuristics, such as variable neighborhood search [3], tabu search [2, 17] and simulated annealing [5].

Guo et al., [11] used a two-stage approach for scheduling air traffic controllers over a two-week period. The type of shifts (morning, evening and night) and the corresponding demand were given. In the first step, they make shift-employee assignments, while breaks are added in the second phase. This paper is organized as follows: in Section 2, we describe the problem. In Section 3, we present the mathematical formulation. In Section 4, we describe the experiments and discuss corresponding results. Finally, in Section 5, we present our conclusions and recommendations for future work.

2 Problem Description

In this section, we briefly formalize the problem we aim to solve in this paper, the problem of shift scheduling for train dispatchers for several days:

Input: A set S of daily shifts that feasibly cover a single day; a set of train dispatchers, D; the length of the planning horizon in days, p; and a lower bound on the rest time between two consecutive shifts, $r^{\min}$.

Output: A dispatcher-minimal assignment of all daily shifts in S to dispatchers for each of the p days that fulfill all restrictions in $\mathcal{R}$, such that the rest time between consecutive shifts is at least $r^{\min}$.

3 Mathematical Formulation

In this section, we present an optimization model for the weekly schedules, with the parameters and variables presented in Tables 1 and 2, respectively. We give our model for daily shifts from [15, 14] in Appendix A for the ease of the reader.

The number of periods equals the number of days for which we are planning, that is, for a week, we have $p = 7$. The input is a set of shifts S that completely cover a day (where we for starters assume the same traffic volume during all 7 days) – S could contain an optimal set of shifts computed with one of our previous models [16, 15, 14], but also a different set of shifts (e.g., computed with the previous models but with other parameters on shift length, such as to obtain better "puzzle bits" for the weekly plan). We denote the elements of S as "daily shifts". We then construct the weekly schedule by enforcing that each shift is handled by some dispatcher every day.

We assume that each dispatcher holds endorsements for all areas, which provides a lower bound for other endorsement structures. If dispatchers hold endorsements only for some areas, we could define subsets $S_i \subseteq S$ of shifts that dispatcher i can cover.

We now give the constraints of our formulation, followed by the objective function and detailed explanations.

$$\sum_{i \in D} x_{i,j,k} = 1 \quad \forall j \in S, \forall k \in P \tag{1}$$

$$\sum_{j \in S} x_{i,j,k} \leq 2 \quad \forall i \in D, \forall k \in P \tag{2}$$

$$x_{i,j,k} \leq q_i \quad \forall i \in D, \forall j \in S, \forall k \in P \tag{3}$$

$$\sum_{j \in S} \sum_{k \in P} x_{i,j,k} \geq q_i \quad \forall i \in D \tag{4}$$

$$x_{i,j,k} + x_{i,j',k} \leq 1 \quad \forall i \in D, \forall k \in P, \forall j, j' \in S : \Delta_{j,j'} < r^{\min} \tag{5}$$

$$x_{i,j,k} + x_{i,j',(k+1) \bmod p} \leq 1 \quad \forall i \in D, \forall k \in P, \forall j, j' \in S : \Delta'_{j,j'} < r^{\min} \tag{6}$$

$$\sum_{k \in P} wr_{i,k} + wr'_{i,k} \geq q_i \quad \forall i \in D \tag{7}$$

$$\delta_{j,j'} \cdot (x_{i,j,k} + x_{i,j',(k+1) \bmod p}) \leq 2 - wr_{i,k} \quad \forall i \in D, \forall k \in P \; \forall j, j' \in S \tag{8}$$

$$\delta'_{j,j'} \cdot (x_{i,j,k} + x_{i,j',(k+2) \bmod p}) \leq 2 - wr'_{i,k} \quad \forall i \in D, \forall k \in P \; \forall j, j' \in S \tag{9}$$

$$x_{i,j,(k+1) \bmod p} + wr'_{i,k} \leq 1 \quad \forall i \in D, \forall j \in S, \forall k \in P \tag{10}$$

$$\sum_{j \in S} \sum_{k \in \{6,7\}} x_{i,j,k} + \sum_{\substack{j' \in S: \\ end_{j'} < start_{j'}}} x_{i,j',k'=5} \leq M \cdot wk_i \quad \forall i \in D \tag{11}$$

$$\sum_{j \in S} \sum_{k \in \{6,7\}} x_{i,j,k} + \sum_{\substack{j' \in S: \\ end_{j'} < start_{j'}}} x_{i,j',k'=5} \geq wk_i \quad \forall i \in D \tag{12}$$

$$\sum_{j \in S} \sum_{k \in P} o_j^{20/06} \cdot x_{i,j,k} + wk_i \leq M \cdot w_i^{38} \quad \forall i \in D \tag{13}$$

$$\sum_{j \in S} \sum_{k \in P} o_j^{20/06} \cdot x_{i,j,k} + wk_i \geq w_i^{38} \quad \forall i \in D \tag{14}$$

$$\sum_{j \in S} \sum_{k \in P} o_j^{20/06} \cdot x_{i,j,k} - 1 \leq M \cdot \theta_i^{20/06} \quad \forall i \in D \tag{15}$$

$$\sum_{j \in S} \sum_{k \in P} o_j^{20/06} \cdot x_{i,j,k} \geq 2 \cdot \theta_i^{20/06} \quad \forall i \in D \tag{16}$$

$$\theta_i^{20/06} + wk_i - 1 \leq w_i^{36} \quad \forall i \in D \tag{17}$$

$$\theta_i^{20/06} + wk_i \geq 2 \cdot w_i^{36} \quad \forall i \in D \tag{18}$$

$$\sum_{j \in S} \sum_{k \in P} \ell_j \cdot x_{i,j,k} + 2 \cdot (w_i^{38} + w_i^{36}) \leq w^{\max} \quad \forall i \in D \tag{19}$$

$$\sum_{j \in S} \sum_{k \in P} o'^{23/05}_j \cdot x_{i,j,k} - 1 \leq M \cdot \theta'^{23/05}_i \quad \forall i \in D \tag{20}$$

$$\sum_{j \in S} \sum_{k \in P} o'^{23/05}_j \cdot x_{i,j,k} \geq 2 \cdot \theta'^{23/05}_i \quad \forall i \in D \tag{21}$$

$$\theta'^{23/05}_i + wk_i - 1 \leq w'^{36}_i \quad \forall i \in D \tag{22}$$

$$\theta'^{23/05}_i + wk_i \geq 2 \cdot w'^{36}_i \quad \forall i \in D \tag{23}$$

$$\sum_{j \in S} \sum_{k \in P} \ell_j \cdot x_{i,j,k} + 4 \cdot w'^{36}_i \leq w^{\max} \quad \forall i \in D \tag{24}$$

$$\sum_{j \in S} \sum_{k \in P} o_j^{23/05} \cdot x_{i,j,k} - 1 \leq M \cdot \theta_i^{23/05} \quad \forall i \in D \tag{25}$$

$$\sum_{j \in S} \sum_{k \in P} o_j^{23/05} \cdot x_{i,j,k} \geq 2 \cdot \theta_i^{23/05} \quad \forall i \in D \tag{26}$$

$$\theta_i^{23/05} + \theta'^{23/05}_i + wk_i \leq 2 + w_i^{34} \quad \forall i \in D \tag{27}$$

$$\theta_i^{23/05} + \theta'^{23/05}_i + wk_i \geq 3 \cdot w_i^{34} \quad \forall i \in D \tag{28}$$

$$\sum_{j \in S} \sum_{k \in P} \ell_j \cdot x_{i,j,k} + 6 \cdot w_i^{34} \leq w^{\max} \quad \forall i \in D \tag{29}$$

The objective function minimizes the number of used dispatchers: $\min . \sum_{i \in D} q_i$ (30)

Constraint (1) ensures that each shift is assigned to exactly one dispatcher during each day. With Constraint (2), we limit the number of shifts a single dispatcher can be assigned during a day to two: with shifts of minimum length 6 and a rest period of at least 11 hours, that is the maximum possible number. Constraint (3) ensures that if a dispatcher is assigned to some shift during some day, they are counted as a working dispatcher, while Constraint (4) ensures that if a dispatcher is not used for any shift during any period, they are not counted as a working dispatcher. Constraint (5) states that if between shifts j and j', worked successively and started on the same day, we have less than the minimum resting time between two consecutive shifts, we can assign at most one of j and j' to the same dispatcher during a day. Similarly, Constraint (6) states that if between two shifts j and j', worked successively and j being an overnight shift, we have less than the minimum resting time between two consecutive shifts, we can assign at most one of j and j' to the same dispatcher during consecutive days. Constraint (7) ensures that any working dispatcher has at least one

Table 1 Model Parameters.

Parameter	Description
D	set of train dispatchers, indexed by i
S	set of daily shifts, indexed by j
P	set of week days in the time horizon, indexed by k
$start_j$	the beginning of the first working hour of shift j
end_j	the beginning of the last working hour of shift j
ℓ_j	the length of shift j
$\Delta_{j,j'}$	the gap (number of rest hours) between shifts j and j' if worked successively and start on the same day
$\Delta'_{j,j'}$	the gap (number of rest hours) between shifts j and j' if worked successively and the first shift ends the next day (overnight shift)
$\delta_{j,j'}$	binary, whether the gap between shifts j and j' is less than 36h if worked in two consecutive days
$\delta'_{j,j'}$	binary, whether the gap between shifts j and j' is less than 36h if started in k and $k+2$, respectively
$r^{\min}$	the minimum resting time between two consecutive shifts
$w^{\max}$	the maximum weekly working hours
$o_j^{20/06}$	binary, whether shift j overlaps period 20-06
$o_j^{23/05}$	binary, whether shift j overlaps period 23-05
${o'}_j^{23/05}$	integer, the number of hours in shift j that overlap period 23-05
$p = \|P\|$	number of time periods (days) in the time horizon
$d = \|D\|$	number of available dispatchers (which maximally can be scheduled in the model)
M	a big number

Table 2 Model Variables.

Variable	Description
$x_{i,j,k}$	binary, whether disp. i is assigned shift j during period k
q_i	binary, whether disp. i works at least one shift/week
$wr_{i,k}$	binary, whether disp. i has weekly rest (36h) between periods k and $k+1$
$wr'_{i,k}$	binary, whether disp. i has weekly rest (36h) between periods k and $k+2$
wk_i	binary, whether disp. i works, at least once, on a weekend
$\theta_i^{20/06}$	binary, whether disp. i works at least twice in period 20-06
$\theta_i^{23/05}$	binary, whether disp. i works at least twice in period 23-05
${\theta'}_i^{23/05}$	binary, whether disp. i works at least 2h/week in period 23-05
w_i^{38}	binary, whether disp. i works at least once in period 20-06 **or** on weekend
w_i^{36}	binary, whether disp. i works at least twice in period 20-06 **and** at least once on weekend
${w'}_i^{36}$	binary, whether disp. i works at least 2h/week in period 23-05 **and** at least once on weekend
w_i^{34}	binary, whether disp. i works at least twice in period 23-05 for at least 2h/week **and** at least once on weekend

weekly rest of at least 36 hours starting during some of the periods. Constraint (8) enforces that there is no weekly rest period starting during day k for dispatcher i if they are assigned a shift j during k followed by a shift j' during $k+1$ between which we have less than 36 hours. Constraint (9) ensures that there is no weekly rest period starting during day k for dispatcher i, if they are assigned a shift j during k followed by a shift j' during $k+2$ between which we have less than 36 hours. Constraint (10) enforces that dispatcher i can be assigned at most one of two things during day $k+1$: a weekly rest period of 36 hours starting during day k and ending during day $k+2$, or a shift during day $k+1$.

Constraint (11) ensures that if dispatcher i is working some weekend shift, that is, a shift starting Saturday or Sunday (periods 6 and 7) or a shift starting on a Friday but ending on the Saturday, the variable indicating weekend work is set to 1. Constraint (12) enforces that if dispatcher i is not working any weekend shift, the variable indicating weekend work is set to 0. If a dispatcher is working either during a weekend or/and a shift during the week that overlaps with the time interval 20-06, the variable indicating this, w_i^{38}, is set to 1 with Constraint (13). If a dispatcher is working none of these shifts, the variable is set to 0 with Constraint (14). Similarly, Constraints (15) and (16) correctly set the variable indicating whether dispatcher i works at least twice during the hours 20-06; Constraints (17) and (18) correctly set the variable indicating whether dispatcher i works at least twice during the hours 20-06 and a weekend shift. Constraint (19) sums up the length of all shifts during the week assigned to dispatcher i and limits those plus 2 hours – if the dispatcher may by the other constraints work at most 38 hours – or plus 4 hours – if the dispatcher may work at most 36 hours – by the maximum weekly working hours of 40.

Constraints (20) and (21) keep track whether dispatcher worked at least two hours during 23 and 05 with the variable $\theta_i^{'23/05}$. Constraint (22) and (23) enforce that the variable indicating whether dispatcher i is working at least two hours within the hours 23-05 and at least once during the weekend, $w_i^{'36}$, is assigned correctly. If that variable is set to 1, Constraint (24) limits the lengths of shifts assigned to i to 36 (otherwise, the limit is 40 hours). Constraints (25) and (26) keep track whether dispatcher i worked at least twice during 23-05 with the variable $\theta_i^{23/05}$. Constraint (27) and (28) enforce that the variable indicating whether dispatcher i is working at least twice and for at least two hours within the hours 23-05 and at least once during the weekend, w_i^{34}, is assigned correctly. If that variable is set to 1, Constraint (29) limits the lengths of shifts assigned to i to 34[1].

4 Experiment Setup and Results

In this section, we present our experiments and their results followed by a discussion. We consider real-world-sized data, comparable to Malmö dispatching center, which is responsible for train traffic in southern Sweden. The dispatching area is partitioned into 15 dispatching areas (the area adjacency graph is given in Figure 2 in Appendix B) and the number of train movements per hour (approximating the dispatcher taskload) is based on discussions with operational experts, where we allow a maximum taskload of 30 per hour and dispatcher (see [16] for a detailed description of the train movements and the area's adjacencies). We assume that a dispatcher can handle a maximum of three areas simultaneously. We use one of our IPs for daily shifts [16, 15, 14] to create feasible sets of daily shifts. In Figure 3 in Appendix C, we present an example of the output of one of these models.

[1] More precisely, 34h and 15 minutes rounded down to 34h given the time resolution of 1h.

The input to the weekly problem consists of a set of shifts, S, represented by the start time, $start_j$, and end time, end_j, of each shift j. Using this data we compute the parameters for the weekly-scheduling model, such as ℓ_j, $\Delta_{j,j'}$, $\Delta'_{j,j'}$, $\delta_{j,j'}$, $\delta'_{j,j'}$, $o_j^{20/06}$, $o_j^{23/05}$, and $o'^{23/05}_j$. For all instances, we set the minimum resting time between two consecutive shifts, $r^{\min}$, to 11 periods (which equals 11 hours here), and the computational time limit to 48 hours.

We run three experiment series, varying the shift lengths and the number of night shifts (i.e., shifts overlapping with the period 20-06) in the daily-shifts input. In the first series, we change the interval of feasible shift lengths and the number of daily shifts when generating data for the daily-shifts model. In the second series, we change the distribution of the shift lengths, i.e., for each instance we generate daily shifts grouped by their length within predefined intervals. The purpose is to better control the structure of the shift lengths (short, medium, long) and to study the effect on the weekly dispatcher-shift assignments. Lastly, in the third series, we keep all parameters from the previous two series, in particular, the total number of daily shifts, while pushing for as few night shifts as possible (as night shifts strongly constrain the allowed weekly working hours). The purpose here is to examine the impact of the night-shift ratio on the weekly schedule.

For modeling and solving the IPs, we use the programming language Python and the solver Gurobi. As hardware, we use a powerful server (Tetralith server, 2019), utilizing Intel HNS2600BPB computer nodes with 32 CPU cores, 384 GB, provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS).

4.1 Series 1: Changing Shift Lengths and Total Number of Shifts

This first experiment series consists of two subseries (1.1 and 1.2). In the first one, we use a set of shifts with lengths within the interval [6-11], and we gradually increase the number of daily shifts. To generate these, we run the daily-shifts model where we set the lower ($T^{\min}$) and upper bound ($T^{\max}$) for the feasible shift length to 6 and 11 periods (which equal hours here), respectively. We start with 21 daily shifts, which is the minimum for a one-day coverage for that specific instance, and gradually increase this number to 25. The rationale for using increasing number of daily shifts is to create more puzzle bits and see how this will affect the quality of the weekly schedule. We use the notation $I_{t,n}^{[\ell b,ub]}$ for the instance with feasible shift lengths within the interval $[\ell b, ub]$, and where t is the total number of daily shifts and n is the number of night shifts among those. In the second experiment subseries (1.2), we adjust the feasible interval of the shift lengths in the daily-shifts model to [7-9]. The idea here is to avoid shifts that are either too short or too long and examine the impact of this change. The minimum number of shifts to cover one day of operations, for this specific instance, is 28. Similarly to Subseries 1.1, we gradually increase the number of daily shifts from 28 to 31.

We define the *slack* as the working hours of a dispatcher that we are not using with the current schedule. For example, if a dispatcher i has been assigned a shift pattern with a total of 34 hours, while according to the legal constraints for this specific shift pattern dispatcher i could have worked up to 38 hours, then the slack is 4 hours. We are not steering for a small slack with our model, but we use the slack as a performance indicator to gauge a shift pattern – with the general expectation that a schedule with large slack uses many dispatchers.

We report the shift-length distribution and the results of the two subseries in rows 2-10 of Table 3. We present the sum of slacks, their average per dispatcher, the total number of needed dispatchers, the optimality gap, and the runtime (*rt*) for each instance.

For both subseries, increasing the number of daily shifts in the input did not always result in a decrease in the number of needed dispatchers for the week. However, the number of dispatchers decreased between the instances $I_{21,16}^{[6,11]}$ and $I_{22,14}^{[6,11]}$. The first instance uses the minimum number of daily shifts, this makes it more likely that many shifts are pushed to the maximum of 11 hours. However, "puzzle bits" of 11 hours do not match well with the upper bounds on working hours of 34, 36, 38, and 40 hours. Thus, instances with a high number of 11-hour shifts are likely to yield large slacks. Increasing the number of daily shifts resulted in a decrease in the number of dispatchers in both $I^{[6,11]}_{22,14}$ and $I^{[6,11]}_{24,17}$, while this was not the case for $I^{[6,11]}_{23,18}$, and even worse for $I^{[6,11]}_{25,21}$, where the number of dispatchers increased to 49. These high numbers of dispatchers in these two instances are probably due to a high number of 11-hour shifts (15 in both cases). Increases/decreases in average slack are accompanied with increases/decreases in the number of 11-hour shifts.

The changes in daily shifts in the second subseries (1.2) yield smaller changes in the number of dispatchers, where the highest value (47) was obtained in $I^{[7,9]}_{29,22}$, which is the instance with the highest number of 9h-shifts in this subseries. The same instance has also the highest slack (both sum and average) within its group. Generally, the slack does not always follow the same trend as the number of dispatchers, which is the value we are minimizing.

From this experiment series, we conclude that adding more daily shifts may improve the weekly schedule, mainly because of adding shorter shifts at expense of longer ones. Too many 11-hour shifts may create a large number of weekly schedules with ca. 33 hours, which in turn yield a high slack. Hence, these experiments highlight the importance of the length-distribution of daily shifts.

The runtime reached the maximum limit of 48h in six out of nine instances with an optimality gap below 4.26. However, most instances reached the current solution relatively quickly, and the latest current solution has been reached before 9 hours.

Table 3 Length distribution and results for the first experiment series (1.1 and 1.2) and second experiment series.

instance name	night ratio	slack $\sum$	slack avg	slack min;max	disp $\sum$	gap %	*rt* (h)	length distr. 6/7/8/9/10/11
$I^{[6,11]}_{21,16}$	0.76	132	2.87	0;7	46	0	0.36	0/0/2/1/1/17
$I^{[6,11]}_{22,14}$	0.64	24	0.58	0;5	41	3.37	max	0/5/1/3/1/12
$I^{[6,11]}_{23,18}$	0.78	61	1.33	0;5	46	3.98	max	0/5/0/1/2/15
$I^{[6,11]}_{24,17}$	0.71	14	0.33	0;4	42	0	9.47	4/3/2/0/6/9
$I^{[6,11]}_{25,21}$	0.84	55	1.12	0;5	49	3.12	max	0/4/2/3/1/15
$I^{[7,9]}_{28,20}$	0.71	30	0.69	0;3	44	2.27	max	0/10/6/12/0/0
$I^{[7,9]}_{29,22}$	0.76	39	0.83	0;5	47	4.26	max	0/6/10/13/0/0
$I^{[7,9]}_{30,23}$	0.77	14	0.30	0;3	46	2.17	max	0/13/6/11/0/0
$I^{[7,9]}_{31,24}$	0.77	7	0.15	0;4	46	0	3.45	0/13/11/7/0/0
$I^{\frac{1}{2}:([6,8],[8,11])}_{21,16}$	0.76	7	0.19	0;1	37	0	0.73	0/1/10/1/0/9
$I^{\frac{1}{2}:([6,8],[8,11])}_{22,17}$	0.77	16	0.4	0;1	40	0	1.49	0/1/10/0/0/11
$I^{\frac{1}{2}:([6,8],[8,11])}_{23,15}$	0.65	12	0.3	0;1	40	0	1.50	0/4/8/0/1/10
$I^{\frac{1}{3}:([6,7],[7,9],[9,11])}_{21,16}$	0.76	3	0.08	0;1	36	0	3.19	0/7/0/7/0/7
$I^{\frac{1}{3}:([6,7],[7,9],[9,11])}_{22,15}$	0.68	6	0.16	0;1	37	0	1.48	0/8/0/7/0/7
$I^{\frac{1}{3}:([6,7],[7,9],[9,11])}_{23,16}$	0.69	12	0.31	0;2	39	0	5.80	1/6/1/8/1/6

4.2 Series 2: Splitting the Shift-Length Interval in Equal Subintervals

To generate more variability in the daily shifts, we adjust the input for the daily-shifts model, splitting the complete interval of allowed shift lengths into subintervals, and requiring roughly equally many shifts that have a shift length in each subinterval. This gave us the input for the second experiment series, containing the Subseries 2.1 and 2.2, which we report together with the corresponding results in rows 11-16 of Table 3. The instances of the first subseries series are $I_{t,n}^{(\frac{1}{2}:[6,8],[8,11])}$, where half[2] of the shifts have a length in the interval [6-8], and the other half have a length within [8-11]. We generated these shifts by adding extra constraints in the daily-shifts model. The two intervals overlap, because non-overlapping intervals may not yield feasible daily shifts when we keep all other parameters unchanged. The instances in the second subseries are $I_{t,n}^{(\frac{1}{3}:[6,7],[7,9],[9,11])}$, where we split the shifts into three equinumerous sets with lengths within the intervals [6,7], [7,9] and [9,11], respectively (again we allow overlapping intervals)[3]. In both subseries, we have instances with 21, 22, and 23 daily shifts.

Both the slack and the number of dispatchers improve in each instance $I_{t,n}^{\frac{1}{2}:([6,8],[8,11])}$ compared to its correspondent in Subseries 1.1 with the same number of daily shifts, $I_{t,n}^{[6,11]}$. Comparing the instances only within this subseries, $I_{21,16}^{\frac{1}{2}:([6,8],[8,11])}$ performs best, which is probably at least partly due to relatively few long (11-hour) shifts. The same trend holds for instances $I_{t,n}^{\frac{1}{3}:([6,7],[7,9],[9,11])}$ compared to their correspondent instances $I_{t,n}^{[6,11]}$. Every instance in the second experiment series outperformed any one in the first series w.r.t. the number of used dispatchers. Moreover, the runtimes are generally much shorter in this second series.

4.3 Series 3: Changing the Percentage of Night Shifts

In all the previous experiments, relatively few shifts covered only daytime, while the majority overlapped with night hours. In this experiment series, we aim to investigate the impact of changing the ratio of the number of night shifts and the total number of shifts by decreasing the number of night shifts (while keeping the total number of daily shifts constant). For each of the previous instances, we generate a corresponding instance with the lowest night-shift ratio for the given total number of daily shifts. The idea here is to minimize the number of shifts involved in the constraints on the total weekly working hours, hence, we define the night shifts as any shift that overlaps with the period 20-06 (we cannot impact the weekend shifts, as all daily shifts have to be manned during the weekend). To generate these shifts in the daily-shifts model, we prohibit some shifts from covering the time interval 20-06, performing a binary search for the highest number of daily shifts that are prohibited to overlap with the night hours, while yielding a feasible solution. In Table 4, we report the shift distribution and the results of these instances, which we denote by I' instead of I.

The number of dispatchers in instance I' is lower than in instance I in 12 out of 15 instances; this number remains unchanged in two cases; and only for $I'^{[6,11]}_{22,12}$, the number is higher than in $I_{22,12}^{[6,11]}$. This sole increase in the number of dispatchers is probably caused by an increase in the number of 11-hour shifts.

The slack decreased for all but three instances I' in comparison to the corresponding instance I (these three instances are marked in red in Table 4). However, in two out of three cases, the increase in slack is accompanied by a decrease in the number of dispatchers (our objective). Hence, the trend in slack is often a good indicator for the trend in the number of needed dispatchers, but this does not always hold.

---

[2] More precisely, we have $\lceil \frac{t}{2} \rceil$ daily shifts in one shift-length interval and $\lfloor \frac{t}{2} \rfloor$ in the other interval.

[3] And again, we actually have two shift-length intervals with $\lfloor \frac{t}{3} \rfloor$ shifts and one shift-length interval with $\lceil \frac{t}{3} \rceil$ shifts.

Table 4 Results for the third experiment series (with adjusted parameters of all instances in the first two series), where the number of night shifts is minimized. For instances I' with a higher slack than the corresponding instance I, we highlight the slack in red; for instances with a higher number of dispatchers, we highlight the number in bold.

instance name	night ratio	slack $\sum$	slack avg	slack min;max	disp $\sum$	gap %	rt (h)	length distr. 6/7/8/9/10/11
$I'^{[6,11]}_{21,13}$	0.62	30	0.77	0;5	39	2.56	max	1/0/1/1/4/14
$I'^{[6,11]}_{22,12}$	0.54	46	1.07	0;5	**43**	2.33	max	2/1/0/2/2/15
$I'^{[6,11]}_{23,11}$	0.48	4	0.1	0;2	38	0	1.42	5/5/0/2/2/9
$I'^{[6,11]}_{24,10}$	0.42	10	0.24	0;4	41	0	2.13	4/4/1/3/1/11
$I'^{[6,11]}_{25,9}$	0.36	1	0.02	0;1	40	0	3.52	7/4/2/1/1/10
$I'^{[7,9]}_{28,13}$	0.46	43	1.02	0;8	42	0	30.28	0/16/5/7/0/0
$I'^{[7,9]}_{29,13}$	0.49	20	0.44	0;5	45	2.22	max	0/10/5/14/0/0
$I'^{[7,9]}_{30,13}$	0.43	12	0.26	0;3	46	2.17	max	0/9/10/11/0/0
$I'^{[7,9]}_{31,13}$	0.42	2	0.04	0;1	46	2.17	max	0/13/9/9/0/0
$I'^{\frac{1}{2}:([6,8],[8,11])}_{21,13}$	0.62	8	0.22	0;2	36	2.78	max	2/3/6/0/1/9
$I'^{\frac{1}{2}:([6,8],[8,11])}_{22,12}$	0.54	4	0.11	0;1	37	0	1.59	6/1/4/0/2/9
$I'^{\frac{1}{2}:([6,8],[8,11])}_{23,12}$	0.52	9	0.23	0;3	39	0	0.18	6/1/5/0/1/10
$I'^{\frac{1}{3}:([6,7],[7,9],[9,11])}_{21,13}$	0.62	1	0.03	0;1	35	0	2.94	1/6/1/6/0/7
$I'^{\frac{1}{3}:([6,7],[7,9],[9,11])}_{22,12}$	0.54	3	0.08	0;1	36	0	0.28	4/4/1/6/0/7
$I'^{\frac{1}{3}:([6,7],[7,9],[9,11])}_{23,12}$	0.52	5	0.13	0;1	38	0	2.61	2/6/1/7/1/6

Since we are not steering the shift-length distribution in this experiment series, we exclude the instances for which not only the number of night shifts changed, but for which the number of the longest shifts decreased. The rationale behind this is that changes in both would not allow us to determine whether the improvements are due to having fewer night shifts. Considering only instances where the number of longest shifts did not decrease (10 instances), only two out of these ten instances, namely $I'^{[6,11]}_{22,12}$ and$I'^{\frac{1}{2}:([6,8],[8,11])}_{21,13}$, had worse results either in terms of increased slack or both increased slack and number of dispatchers. The remaining eight out of ten instances yield better results, which is possibly because of fewer night shifts.

In Figure 1, we present the final weekly schedule obtained for instance $I'^{\frac{1}{3}:([6,7],[7,9],[9,11])}_{21,16}$. This is the instance with the lowest number of needed dispatchers (35).

Generally, minimizing the number of night shifts has a positive effect on the weekly schedule, which is in line with our expectations, since having more night shifts would activate more constraints that limit the weekly working hours, thus, increasing the number of needed dispatchers.

5 Conclusions and Future Work

Solving shift-scheduling problems with a long time horizon and a high resolution is a complex problem that requires a huge amount of computation time, and in many cases depending on the size and the problem type, without reaching a global optimum. In this paper, we suggest a two-stage approach for weekly shift scheduling of train dispatchers. First, we use our previous model from [16, 15, 14] to generate feasible daily shifts, which we then use as input to the weekly-shift-scheduling model presented in this paper. We run three experiment

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
disp.1	15-22	-	-	-	08-19	08-19	11-20
disp.2	14-21	11-20	-	-	-	08-19	08-19
disp.3	16-23	18-05	23-08	22-09	-	-	-
disp.4	05-12	18-05	18-05	23-08	-	-	-
disp.5	13-20	22-09	-	-	-	18-05	16-23
disp.6	14-20	08-19	06-13	10-19	13-20	-	-
disp.7	06-13	15-00	23-10	23-10	-	-	-
disp.8	10-19	06-13	14-20	11-20	11-20	-	-
disp.9	11-20	-	08-19	08-19	10-19	-	-
disp.10	11-20	14-20	08-19	06-13	06-13	-	-
disp.11	15-00	-	13-20	11-20	-	14-20	13-20
disp.12	05-14	10-19	-	14-20	-	06-13	06-13
disp.13	15-23	15-23	16-23	15-23	14-21	-	-
disp.14	23-08	05-12	18-05	18-05	-	-	-
disp.15	08-19	08-19	-	-	-	11-20	14-21
disp.16	08-19	-	-	15-00	-	10-19	10-19
disp.17	18-05	23-08	-	23-10	05-12	-	-

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
disp.18	23-10	23-10	-	16-23	15-00	-	-
disp.19	22-09	05-14	-	18-05	16-23	-	-
disp.20	23-10	23-10	05-12	-	11-20	-	-
disp.21	18-05	16-23	23-10	-	05-14	-	-
disp.22	-	-	05-14	-	23-10	05-12	05-14
disp.23	-	-	22-09	05-14	-	14-21	15-00
disp.24	-	-	10-19	-	15-23	15-23	23-10
disp.25	-	-	-	-	18-05	18-05	18-05
disp.26	-	-	11-20	08-19	-	15-00	11-20
disp.27	-	13-20	11-20	-	14-20	11-20	15-22
disp.28	-	-	14-21	-	08-19	15-22	18-05
disp.29	-	-	-	05-12	22-09	23-10	05-12
disp.30	-	-	-	14-21	15-22	23-10	08-19
disp.31	-	14-21	-	-	18-05	16-23	22-09
disp.32	-	11-20	15-00	13-20	-	13-20	14-20
disp.33	-	-	15-23	-	23-10	05-14	15-23
disp.34	-	15-22	-	15-22	-	22-09	23-10
disp.35	-	-	15-22	-	23-08	23-08	23-08

Figure 1 The weekly schedule obtained for instance $I'^{\frac{1}{3}:([6,7],[7,9],[9,11])}_{21,16}$. The values in the cells represent the start and end time of each shift. To exemplify the daily assignment of all daily shifts, we marked two of the 21 daily shifts, the shifts 05-12 and 16-23 in red and blue, respectively.

series, with a real-world sized data, where we: change the number of daily shift and their feasible lengths, split the feasible shift-length interval into equal subintervals, and change the percentage of night shifts among the daily shifts.

We conclude that increasing the number of daily shifts may improve the weekly schedules, especially if this would decrease the percentage of the too long shifts (11h). Moreover, enforcing more variability in the daily shifts' length, and reducing the night-shift ratio can also improve the quality of the weekly schedules.

For stage two in our approach, the runtimes in our experiments were between a few minutes up to being stopped after two days with an optimality gap below 4.26%. The instances with minimized night shift ratio and with three subintervals of the feasible lengths performed very well, and also had a relatively low runtime (between 0.28 and 2.94h). Thus, we recommend those daily-shift patterns. For stage one in our approach, computing one feasible set of daily shifts – using a model to minimize the dispatcher-area-assignment switches [15, 14] – takes less than two hours. However, if we want to achieve the fewest possible night shifts, we may have to perform multiple runs. Minimizing the number of dispatchers only is much faster (with a runtime of maximum 20 seconds). We could use this objective for our daily-shifts model to obtain the minimum number of night shifts (performing a binary search on that number with a few seconds runtime per run) and then once run the computationally more expensive model to minimize dispatcher-area-assignment switches; with this, we would prioritize the smallest number of night shifts over the lowest possible number of assignment switches. Given the planning horizon, the resulting runtime for the complete approach can be acceptable for real-world use, in particular, given the quality of the resulting schedules.

We opted for a two-stage approach, mainly to gain insight not only in the weekly schedules, but also in "good" daily-shift distributions for the operational planning. Another natural approach is column generation, where both constraints on the daily shifts and those on the weekly shifts are integrated. The split in master and pricing problem could, for example, be steered by operational vs. legal constraints on the shifts, respectively.

References

1. Burak Bilgin, Patrick De Causmaecker, Benoît Rossie, and Greet Vanden Berghe. Local search neighbourhoods for dealing with a novel nurse rostering model. *Annals of Operations Research*, 194:33–57, 2012.
2. Michael J Brusco and TR Johns. An integrated approach to shift-starting time selection and tour-schedule construction. *Journal of the Operational Research Society*, 62(7):1357–1364, 2011.
3. Edmund K Burke, Timothy Curtois, Gerhard Post, Rong Qu, and Bart Veltman. A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem. *European journal of operational research*, 188(2):330–341, 2008.
4. Mehmet Tolga Cezik and Pierre L'Ecuyer. Staffing multiskill call centers via linear programming and simulation. *Management Science*, 54(2):310–323, 2008.
5. Jean-François Cordeau, Gilbert Laporte, Federico Pasin, and Stefan Ropke. Scheduling technicians and tasks in a telecommunications company. *Journal of Scheduling*, 13(4):393–409, 2010.
6. Sophie Demassey, Gilles Pesant, and Louis-Martin Rousseau. Constraint programming based column generation for employee timetabling. In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems: Second International Conference, CPAIOR 2005, Prague, Czech Republic, May 31-June 1, 2005. Proceedings 2*, pages 140–154. Springer, 2005.
7. Sophie Demassey, Gilles Pesant, and Louis-Martin Rousseau. A cost-regular based hybrid column generation approach. *Constraints*, 11(4):315–333, 2006. `doi:10.1007/s10601-006-9003-7`.
8. Robert G Drake. The nurse rostering problem: from operational research to organizational reality? *Journal of advanced nursing*, 70(4):800–810, 2014.
9. eurostat. Passenger transport by type of transport (detailed reporting only), 2024. URL: `https://ec.europa.eu/eurostat/databrowser/view/RAIL_PA_TYPEPAS/default/line?lang=en`.
10. Michel Gamache, François Soumis, Gérald Marquis, and Jacques Desrosiers. A column generation approach for large-scale aircrew rostering problems. *Operations Research*, 47(2):247–263, 1999. `doi:10.1287/opre.47.2.247`.
11. Jia Guo and Jonathan F Bard. Air traffic controller scheduling. *Computers & Industrial Engineering*, 191:110123, 2024.
12. Dennis Huisman. A column generation approach for the rail crew re-scheduling problem. *European Journal of Operational Research*, 180(1):163–173, 2007. `doi:10.1016/j.ejor.2006.04.026`.
13. Gilbert Laporte and Gilles Pesant. A general multi-shift scheduling system. *Journal of the Operational Research Society*, 55(11):1208–1217, 2004.
14. Tomas Lidén, Christiane Schmidt, and Rabii Zahir. Improving attractiveness of working shifts for train dispatchers. Under revision for journal publication.
15. Tomas Lidén, Christiane Schmidt, and Rabii Zahir. Improving attractiveness of working shifts for train dispatchers. In *25th Euro Working Group on Transportation Meeting (EWGT 2023)*, 2023. Extended abstract presented only.
16. Tomas Lidén, Christiane Schmidt, and Rabii Zahir. Shift scheduling for train dispatchers. In *RailBelgrade 2023: the 10th International Conference on Railway Operations Modelling and Analysis (ICROMA)*, 2023.
17. Nysret Musliu, Andrea Schaerf, and Wolfgang Slany. Local search for shift design. *European journal of operational research*, 153(1):51–64, 2004.
18. Emilie M. Roth, Nicholas Malsch, and J. Multer. Understanding how train dispatchers manage and control trains : results of a cognitive task analysis. Technical Report DOT-VNTSC-FRA-98-3;DOT/FRA/ORD-01/02, John A. Volpe National Transportation Systems Center (U.S.), 2001.

19 Pieter Smet, Burak Bilgin, Patrick De Causmaecker, and Greet Vanden Berghe. Modelling and evaluation issues in nurse rostering. *Annals of Operations Research*, 218:303–326, 2014.
20 Trafikverket. Kollektivavtal: Trafikverkets affärsverksavtal mellan Trafikverket och Saco-S, OFR-S och Seko, 2019. URL: `https://www.seko.se/siteassets/forhandlings-branschorganisationer/seko-klubb-trafikverket/pdf-er/test/trafikverkets-affarsverksavtal_andringar_tillagg_till_och_med_2019_10_01_20191025.pdf`.
21 Trafikverket. Sveriges järnvägsnät, 2024. Accessed: 2024-06-26. URL: `https:https://www.trafikverket.se/resa-och-trafik/jarnvag/sveriges-jarnvagsnat/`.

A Mathematical Model for Daily Shifts

For the ease of the reader, we provide the full model used for computing the daily shifts that we presented in [15, 14] (an improvement of the model from [16]) in this section of the appendix. While we also presented different approaches for minimizing changes in the dispatcher-area assignment in [15, 14], here, we give the improved model for minimizing the number of used dispatchers while observing all operational and legal requirements for shifts during one day.

We give the notation we used in the model in Tables 5 and 6. Because we aim to minimize the total number of used dispatchers, our objective function is given by (31).

$$\min . \quad \sum_{i \in D} q_i \tag{31}$$

The rest of the model is given by Constraints (32)-(43).

Table 5 Model parameters in the one-day-shifts model.

Parameters	Description
D	set of train dispatchers, indexed by i
A	set of geographical areas, indexed by j
P	set of time periods, indexed by k
C	set of area combinations, indexed by ℓ
$TL_{j,k}$	task load in area j during period k
$TL^{\max}$	maximum allowed task load
$A^{\max}$	maximum number of assigned areas to a dispatcher per period
$e_{i,j} \in \{0,1\}$	=1 if dispatcher i holds an endorsement for area j
$T^{\min}$	minimum shift length (in time periods)
$T^{\max}$	maximum shift length (in time periods)
$r^{\min}$	minimum number of rest periods between two shifts
$p = \|P\|$	number of time periods in the time horizon

Table 6 Model variables in the one-day-shifts model.

Variables	Description
$x_{i,j,k} \in \{0,1\}$	=1 if dispatcher i is assigned area j during period k
$c_{i,\ell,k} \in \{0,1\}$	=1 if dispatcher i is assigned area combination ℓ during period k
$y_{i,k} \in \{0,1\}$	=1 if dispatcher i is at work during period k
$v_{i,k} \in \{0,1\}$	=1 if dispatcher i starts a shift at the beginning of period k
$q_i \in \{0,1\}$	=1 if dispatcher i is used during some period

$$\sum_{\mu=k+1-T^{\min}}^{k} v_{i,\mu(\text{mod } p)} \leq y_{i,k} \qquad \forall i \in D, \forall k \in P \tag{32}$$

$$\sum_{\mu=k+1-T^{\max}}^{k} v_{i,\mu(\text{mod } p)} \geq y_{i,k} \qquad \forall i \in D, \forall k \in P \tag{33}$$

$$v_{i,k} \geq y_{i,k} - y_{i,(k-1)(\text{mod } p)} \qquad \forall i \in D, \forall k \in P \tag{34}$$

$$v_{i,k} \leq y_{i,k} \qquad \forall i \in D, \forall k \in P \tag{35}$$

$$v_{i,k} \leq q_i \qquad \forall i \in D, \forall k \in P \tag{36}$$

$$\sum_{k \in P} v_{i,k} \geq q_i \qquad \forall i \in D \tag{37}$$

$$\sum_{\mu=k+1}^{k+r^{\min}} v_{i,\mu(\text{mod } p)} \leq q_i - y_{i,k} \qquad \forall i \in D, \forall k \in P \tag{38}$$

$$\sum_{j \in A} \sum_{\ell \in C} c_{i,\ell,k} \cdot a_{\ell,j} \cdot TL_{j,k} \leq TL^{\max} \qquad \forall i \in D, \forall k \in P \tag{39}$$

$$c_{i,\ell,k} \leq e_{i,j} \qquad \forall i \in D, \forall \ell \in C, \forall j \in \ell, \forall k \in P \tag{40}$$

$$\sum_{\ell \in C \setminus \{0\}} c_{i,\ell,k} = y_{i,k} \qquad \forall i \in D, \forall k \in P \tag{41}$$

$$c_{i,0,k} = 1 - y_{i,k} \qquad \forall i \in D, \forall k \in P \tag{42}$$

$$\sum_{\ell \in C \setminus \{0\}} \sum_{i \in D} a_{\ell,j} \cdot c_{i,\ell,k} = 1 \qquad \forall k \in P, \forall j \in A \tag{43}$$

B Adjacency Graph of the Considered Dispatching Areas

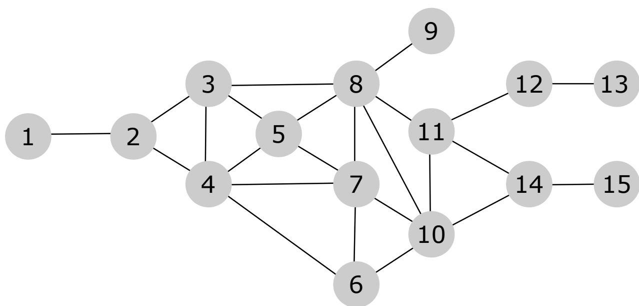

Figure 2 Adjacency graph of the considered dispatching areas.

C Example Output of the Daily-Shifts Model

	pr.0	pr.1	pr.2	pr.3	pr.4	pr.5	pr.6	pr.7	pr.8	pr.9	pr.10	pr.11	pr.12	pr.13	pr.14	pr.15	pr.16	pr.17	pr.18	pr.19	pr.20	pr.21	pr.22	pr.23
D1						10 11	10 11	10 11	10	12 13	12 13	12	12	12	12	12								
D2	6	6														6	6	6	6	6	6 7	6 7	6 7	6
D3									11	10 11	6 10 11	6 10 11	6 10 11	6 10 11	6 10 11	10 11	10 11	10 11	10					
D4											3	3	3	3	3	3	3	3	3	3	3			
D5											14 15	14 15	14 15	14 15	14	14	14	14	14	14	14			
D6	3	3	3	3	3 4	3 4	3 4	3 4	3 4	3														3
D7	4 5	4 5	4 5	4															4	4	4 5	4 5	4 5	4 5
D8									1 2	1 2	1 2	1 2	1 2	1 2	1 2	1 2	2	2	2					
D9	10 11	10 11	10 11	10 11	10 11														11	10 11	10 11	10 11 14	10 11 14	10 11
D10										4	4 7	4 7	4 7	4 7	4 7	4 7	4 7	4 7	7	7				
D11												13	13	13	13	13	12 13	12 13	12 13	12 13	12 13	12 13		
D12															8	8	8	8	8	8	8 9	8 9		
D13																	1	1	1	1 2	1 2	1 2	1 2	1
D14	8 9	8 9	8 9	5 8 9	5 8 9	5 8 9	5 8 9	5 8 9	5 8 9														8 9	8 9
D15															15	15	15	15	15	15	15	15	15	
D16	12 13	12 13	12 13	12 13	12 13	12 13	12 13	12 13	12 13														12 13	12 13
D17										9	9	9	9	9	9	9	9	9	9	9				
D18										5 8	5 8	5 8	5 8	5 8	5	5	5	5	5	5				
D19	1 2	1 2	1 2	1 2	1 2	1 2	1 2	1 2														3	3	2
D20	14 15	14 15	14 15	14 15	14 15	14 15	14 15	14 15	14 15	14 15														14 15
D21	7	7	6 7	6 7	6 7	6 7	6 7	6 7	6 7	6 7														7

Figure 3 An optimum schedule obtained by the daily-shifts model given an instance with 24 periods and 15 areas. Rows and columns stand for dispatchers and time period of a day, respectively. The values in the cells represent the assigned areas, while the colors are used for clarity and distinction between rows.