University of Paderborn
Faculty of Business and Economics
Depatment of Information Systems
Decision Support& Operations Research Lab

Working Papers

WP1304

Cyclic and non-cyclic crew rostering problems in public bus transit

Lin Xie

Paderborn, Germany
April, 2013

Abstract The crew rostering problem arises in public transport bus companies, and addresses the task of assigning a given set of anonymous duties and some other activities, such as standbys and days off, to drivers or groups of drivers, without violating any complex labor union rules. Additionally, the preferences of drivers are considered during the assignment. The plan generated for each driver/group of drivers is called a roster. Optimal rosters are characterized by maximum satisfaction of drivers and minimal operational costs. In order to generate a personalized roster for each driver/group of drivers, the problem is formulated as a multi-commodity network flow problem in this paper. In each network layer, a roster is generated for each driver or driver group. The network model is very flexible and can accommodate a variety of constraints. Additionally, with a minor modification, the network can formulate the cyclic and non-cyclic crew rostering problems. To the best of our knowledge, this is the first publication which solves both problems with one model. The main goal of this paper is to develop a mixed-integer mathematical optimization network model for both problems with sequential and integrated approaches and to solve this model using commercial solvers. Both problems are usually solved with the sequential approach. Therefore, another contribution of this paper is comparing the sequential approach with the integrated one. Our experiments on real-world instances show that the integrated approach outperforms the sequential one in terms of solution quality.

Keywords Transportation · Crew rostering · Multi-commodity network flow · Cyclic crew rostering · Non-cyclic crew rostering

1 Introduction

The tactical and operational planning process of public bus transport companies is divided into four steps that are normally carried out sequentially. The relationship between the four planning problems of a public bus transport company is illustrated in Figure 1 (from [11]). In the tactical planning process, line routes and their frequencies are given as input for the *timetabling problem*. In addition, the travel times along the lines as well as any potential layover times at stations are assumed to be known. In timetabling, timetables are determined and a set of timetabled trips with start and end locations and times are given for the next step, operational planning. In *vehicle scheduling*, vehicles are assigned to timetabled trips resulting in vehicle blocks. After defining a sequence of tasks in each vehicle block, each task must be assigned to one duty for a working period, which usually is one day long. A duty is subject to satisfy several work regulations, such as the regulation of length and frequency of breaks within a duty. This process is defined as *crew scheduling*. The generated duties, as well as some other given activities, such as standbys, are required to be covered by drivers. Driver assignment occurs in the last step of the operative planning process in the *crew rostering problem*. The law and labor union rules as well as personal preferences of drivers are considered during the assignment. The resulting schedule for each driver/group of drivers is called a roster. The details about the input information of activities and drivers as well as the rules and regulations of the crew rostering problem can be found in Section 2.

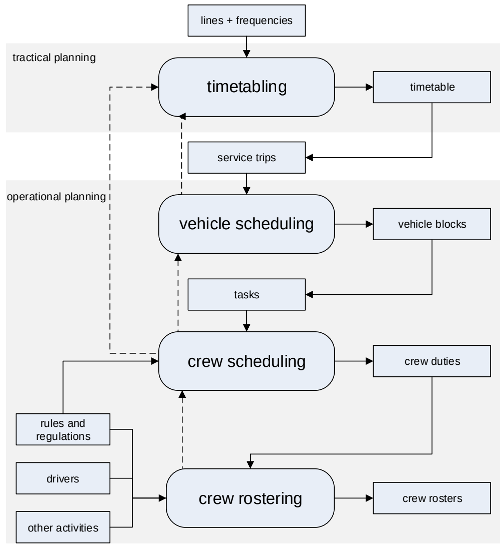


Fig. 1: The sequential planning process in public bus transit [11].

This paper focuses on the crew rostering problem in public transit, which has not achieved as much attention in the research literature as the other phases in operational planning. In this problem, not only the operational costs, as in other steps, are considered, but also the preferences of drivers. The rosters which are generated by considering desires of drivers bring higher acceptance than rosters that ignore individual wishes. Therefore, an efficient use of drivers becomes more and more important in public transport companies and, thus, is worth being thoroughly analyzed. There are different ways to generate a roster for a driver in public bus transit, i.e., cyclic and non-cyclic rostering. To the best of our knowledge, this is the first publication which solves both problems with one model. Additionally, both problems are mostly solved with different sequential approaches, however, a trend towards an integrated approach is observed in recent literature. In this paper we develop both the sequential and integrated approaches for comparison. In our approach we plan the rosters based on the given available number of drivers, i.e., we do not minimize the number of drivers as an objective.

2 Crew Rostering Problem

In this section, we first give details about the input information for solving the crew rostering problem. After a definition of cyclic and non-cyclic crew rostering problems, a short description of sequential and integrated approaches follows.

2.1 Input Information

As shown in Figure 1, the inputs for the crew rostering problem include activities (duties and others) that are required to be assigned, information about drivers, as well as work regulations.

Activity

Each generated *duty* in the crew scheduling problem has a beginning time, ending time, paid work duration (not necessarily equal to the ending time minus beginning time), the calendar day/weekday it belongs to, the *shift type* (e.g., an early shift, which means duties with this type begin between 0:01 am and 11:59 am.), and the vehicle and depot types. The duties need to be assigned to a group of possible drivers who work at the depot of those duties and can drive the bus with the required vehicle type. Not only the duties /shifts must be assigned to drivers, but also some other activities, such as standbys, leaves, and trainings. *Standby* activities are planned to cover the absences of drivers, while *leaves* are vacations. The *trainings* and vacations are preassigned and fixed for each driver and cannot be changed in the optimization. The distribution of them is given by bus companies and drivers. Summarized, the possible activities we consider in the crew rostering include the duties generated in crew scheduling, standbys, days off, and other activities such as training periods or annual leaves.

Driver information

The data of drivers not only include the depot and vehicle types, but also the number of the days off, the target working hours for the current planning period, the target number of standby activities as well as the required trainings and leaves. All of those depend on the work contracts and the current work-accounts of the drivers. The work-account of each driver includes the current overtime and the number of days off from the previous periods. It describes the credit of a driver, e.g., a driver with more overtime in the previous periods can get more jobs with shorter working hours in order to reduce overtimes gradually. Additionally, the drivers can express their preferences, including the daily desired activities and the possible combination of activities. The daily desire of a driver means that the driver wishes to get one activity on one day, while combination desire reflects the driver's desired combination of activities. For instance, a driver may desire similar shifts within a working week, or/and do not get an early shift after days off. Hanne et al. [22] explain the importance of considering the preferences of employees in scheduling their working time, in particular, the advantages of assigning flexible working times to employees. They mention that the employees can better manage their work-life balances, which brings increased morale and reduced absenteeism. That means less exchanges and less absence in operational days. Therefore, less recovery activities are expected, which implies lower operational costs. Moreover, they assume that happy staff means happy customers. Better services are expected. Furthermore, the drivers define their leaves, while the dates of training for each driver are given by bus companies. As mentioned before, these activities are fixed to one in the optimization.

Rule and Regulation

The rules and regulations can be labor rules, some of which are imposed by the bus companies, and the others are due to the agreement between the bus company and employee unions. It is also allowed that every bus company can define its own internal constraints that are stricter than those required by law. We consider three types of rules: horizontal rules, vertical rules, and quality rules. Compared to the rules in airline sector in [15], we have more complex horizontal rules (due to complex rules to generate a feasible roster), but less complex vertical rules (since drivers work mostly alone).

Horizontal rules depend only on one roster and consider compatibility, working/days off block.

Compatibility: The incompatible connections of activities are gathered in a list of forbidden sequences according to, for example, the working regulations. An early shift is forbidden to follow a late shift because of the short of rest times between them. The forbidden sequences are given by bus companies, and it is possible that an undesired combination of activities of drivers is included to be forbidden as well. The length of forbidden sequences is not limited to two, for instance, a late shift one day before a standby and an early shift one day later may be forbidden if they appear simultaneously on three consecutive days. Additionally, two other situations are also considered (the definition of double-off and single-off). A double-off is defined as at least two consecutive days off, while a single-off is a single day off between two work-related activities. Both definitions would be needed in the quality rules. Many of the restrictions in this class can be implicitly considered during the network generation (see Section 4.1).

Working/Days-off block: the maximum consecutive number of working days and days off are restricted based on working regulations. For example, the length of the consecutive working days is limited to five, while the length of the consecutive days off is restricted to three.

Vertical rules combine the information among all rosters.

Upper bound of activities: Not only each duty/shift of each day is limited to be assigned, but the available number of days off and standbys is also limited. The limit depends on the number of drivers and duties available on that day. The upper bound can be easily computed before optimization.

The horizontal and vertical rules described above are formulated as hard constraints in our model (see Section 4.2), while the following quality rules are formulated as soft constraints. They can be suspended by applying specific penalty costs in objectives to restrict the amount of violations.

Quality rules are horizontal rules, however, without these rules the generated rosters are legal, but they affect the quality of the rosters.

Preferences: in this category, the preferences of the bus company as well as the drivers are considered. As mentioned before, the daily desired activities and the possible combination of activities are given by drivers. Some alternatives can be provided by the bus company if the primary desires of the drivers cannot be satisfied. The number of alternatives, as well as the penalty costs of them can vary from one bus company to another. For example, a linear function as shown in [8] is possible for calculating the penalty costs of alternatives. A higher cost based on a calculation function will be assigned to a less desirable activity or activities combination. Due to the limited number of drivers that may simultaneously take a day off, some desired days off (on weekends) can not be satisfied, but be moved within a couple of days (earlier or later), for example, in three days. These are called moved days off (on weekends), and the number of them is limited due their unpopularity. The number of single-off activities is due to preferences often restricted. The distance between two double-off activities is defined in the company rules to make sure, for example, that a driver gets at least two doubles off within 16 days.

Underrun of activities: The underrun of defined number of days off for each driver reflects the fairness of the days off distribution for all drivers. Additionally, the shortage of standbys and duties/shifts might cause additional personnel costs or additional costs for overtimes of all planned drivers.

Fairness: Besides the above mentioned fair distribution of days off, the maximum overtime of all drivers should be minimized, so a huge difference in working times can be avoided. Additionally, this results in less payments for overtime.

Robustness: Usually in operational days many disruptions occur, such as illness, bad weather or technical problems. Therefore, a set of standbys should be optimally planned to cover the absences, while the number of them should be minimized to avoid unused standbys. In this work, the number of standbys per day is defined by the bus company by experience. One publication about the optimal planning of the assignment of standbys based on the historical statistics of illness is [34].

Moreover, we need to consider the previous planning period, which is assumed to be fixed for the actual planning period. Some other fixed activities are also considered in the current planning periods, such as a training period and annual leaves. We assume that all fixed activities comply with the horizontal and vertical rules. However, between the fixed and unfixed activities the horizontal rules should be checked. The details will be shown together with the example in Figure 3 of the next section.

2.2 Cyclic and Non-cyclic Crew Rostering

There are different ways to generate a roster. A *cyclic roster* is generated for a group of drivers who have the same qualifications and similar preferences. Such a roster includes several rows of duties from Monday to Sunday. The number of the rows is equal to the number of drivers in this group. All drivers within a group use the same roster but begin with different rows. An example is shown in Figure 2, in which different activities, ES (early shift), MS (midday shift), LS (late shift) as well as F (day off), are assigned to two drivers.

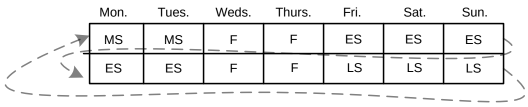


Fig. 2: An example of a cyclic roster.

In cyclic rostering, the number of days in the planning period is equal to the number of drivers multiplied by seven. Each week is assigned to each driver in such a way that each week pattern is worked parallel by a person. Cyclic rostering is a rather simple way to generate rosters, because instead of generating a roster for each driver a roster is generated for a group of drivers. Furthermore, the roster reflects the fairness since drivers within a group have the same duties including unpopular duties, and the days off and weekends-off are evenly distributed. However, a cyclic roster considers the duties from the weekdays, not the calendar days. Therefore, it is not flexible enough to respond to changes in traffic, such as more traffic on holidays. Moreover, a new employee can not be easily added to an existing group, since the planning horizon will be changed, and one more week must be planned. In summary, the cyclic roster needs recovery due to disruptions in the operational context.

The shortcomings of cyclic rostering can be avoided if an individual roster is generated for each driver within a given time period (e.g., one month). Such a *non-cyclic roster* may consider wishes with respect to special single day off or vacation periods. Non-cyclic rostering provides more freedom to take holidays and special events into account. Figure 3 shows an example of a two-week, non-cyclic rosters for two drivers, in which the first five days are in the previous period, and driver d1 is allowed to get a day off on the 11th day. As mentioned before, the horizontal rules between the fixed and unfixed activities must be checked. We assume that the maximum consecutive number of working days is restricted to 5. The five fixed activities in the last period of the driver d2 are work-related activities; therefore, a day off activity should be assigned to the first day of the planning period (Day 6 in the example). Additionally, we assume the maximum length of a days off period is three days. A working day is therefore required for the driver 1 on day 6. We assume that the distance of two double-off activities from the previous period to the current one is defined to be 7, therefore, in this example the last double-off in the previous period of the driver d1 is on the 5th day, and a double-off is required within the next 7 days. Such restrictions can be used to reduce the complexity of our network design in Section 4.1. Note that the previous period is implicitly considered in the cyclic roster, if the roster is not disrupted.

	Day1	Day2	Day3	Day4	Day5	Day6	Day7	Day8	Day9.	Day10	Day11	Day12	Day13	Day14
d1	MS	MS	F	F	F	ES	ES	ES	ES	F	F	LS	LS	LS
d2	ES	ES	MS	MS	MS	F	F	MS	MS	MS	MS	ES	F	F

Fig. 3: An example of a non-cyclic roster.

2.3 Sequential vs. Integrated Approach

The crew rostering problem is usually divided to several sequential sub-problems due to its high complexity (see [21]). A roster is a schedule of combinations of work-related activities (standby, shift/duty) and day off activities (single-off, double-off). The sub-problems are *days off scheduling* (the optimal distribution of days off by considering fair distribution of days off), *shift assignment* (the allocation of shifts to drivers), and *duty sequencing* (according to the result of shift assignment, duties are assigned to drivers). The problem of integrated days off scheduling and shift assignment is named *rota scheduling* in [9] and [34]. In this paper, the integrated shift assignment and duty sequencing problem is called *shift-duty assignment*.

In order to understand the drawbacks of the sequential crew rostering problems, for example the approach we use in this paper, rota scheduling first and duty sequencing second, two examples will be shown below. An *early shift* (ES) means a duty begins between 0:01 am and 11:59 am, while a duty with the *midday shift* (MS) begins between 12:00 pm and 8:59 pm. Each *late shift* (LS) duty begins between 9:00 pm and 12:00 midnight. The minimum rest period is defined as 12 hours.

Example 1. We assume that the sequence (MS, ES) is selected in the rota scheduling problem. The rush hours in the morning are, for example, between 6:00 and 8:00 am, and in the evening between 4:00 and 7:00 pm. However, a duty with MS that begins at 3:00 pm and ends at 10:00 pm cannot be followed by ES duties the next day, which begins before 10:00 am. That means, if we allow the sequence (MS, ES) in the rota scheduling problem, however, many MS duties begin before rush hours in the evening can not be followed by any ES duties during the rush hours on the morning. Therefore, the selected sequences (MS, ES) in the rota scheduling can not be assigned by most duties in duty sequencing. This results in many empty days for the employees, and more duties remain unassigned. However, some of them could be assigned, if we consider the duties in the rota scheduling. Such an approach is called *integrated crew rostering*.

Example 2. We assume that the sequence (LS, MS) is forbidden in the rota scheduling problem, due to violating overnight rest time. However, some duties of LS, such as ending after 4:00 am, can take the duties of MS on the day beginning after 4:00 pm. Therefore, they are not needed in the list of forbidden sequences of shifts. The check of violation can occur directly with the duties in the integrated crew rostering.

In order to avoid the drawback of the sequential approach in practice, it is standard to define more and more shift types, such as defining 14 shift types instead of 4. However, it brings more complexity, and some duties are still left unassigned. The details will be shown in the results section(see Section 5).

3 Literature Review

We classify the papers about cyclic and non-cyclic crew rostering in public transit in the following sections, according to which (sub-)problems they consider.

3.1 Cyclic crew rostering (CCR)

Prakash et al. [28] use goal programming to solve the multi-objectives days off scheduling problem. The goals are to limit the total number of crews having non-consecutive and consecutive days off, and to minimize the number of crews having non-consecutive days off. Two days off for each driver per week are guaranteed. In order to get an integer solution, a modified simplex algorithm is used to solve the problem. Pedrosa and Constantino [27] present a set covering model for the days off scheduling problem to minimize the total number of workers. Column generation is used in which the master problem is a set covering model, while the pricing problem is solved by using the resource constrained shortest path problem of a designated network. The LP solution is rounded or branch-and-bound is used to get the integer solution.

The objective of [31] (a simplified shift-duty assignment problem) is to produce a good match of weekday with week-end duties, and the weekday duties are considered be to identical. The three basic building units (1-week, 4-week and 5-week designs) are used. The problem is formulated as a 3-dimensional assignment problem by considering minimum breaks between consecutive duties. A 2-dimensional assignment problem is then solved, and alternatively a heuristic is used to reduce the computational effort with a small loss in optimality. Townsend [32] aims at producing an even spread of works at weekends through the roster, so a method is developed to produce the roster skeleton, which includes standard units containing fixed patterns of working (it is similar to duty sequencing based on the skeleton). Jachnik [14] solves the duty-sequencing problem considering planned leaves to minimize roster costs by using a constructive heuristic. After that, the solution is improved through swaps. In [10], the duty sequencing problem is modeled as an asymmetric traveling salesman model and solved using a 3-opt based heuristic. The goal is to minimize the variance of the rest periods between successive duties.

Lezaun et al. [17] propose a practical model at Metro Bilbao in Spain, which is solved as a sequence of four types of integer programming problems.

It is planned for the whole year, i.e., different periods of the year are planned and vacations and standby days are allocated by calendar weeks. First of all, standby days are grouped into weeks and standby weeks should, when possible, immediately precede or follow vacations. After that, multi-shift weekly patterns are determined for all combinations of shifts, and the most attractive patterns are generated and assigned to drivers. At last, the annual assignment of weekly work patterns to drivers, i.e., the compatibility of the last week of the last period with the first week of the actual period should be checked. The obtained solution is quasi optimal because all drivers do the same jobs.

Emden-Weinert et al. [9] solve the CCR problem as the rota scheduling and duty sequencing problems sequentially at the Bremer Strassenbahn AG in Bremen, Germany. The former problem is modeled as a mixed 0-1 linear program and solved with a standard solver, while the later ones use simulated annealing with pre- and post-processing to modify infeasible solutions. Multiple objectives are considered in the rota scheduling problem, such as even distribution of weekends-off, preferred shift distribution, etc. A weight for each objective defines its importance. The objectives considered in duty sequencing problem are, for example, even distribution of work over the duty blocks of roster, meeting the predefined average working time, etc. Sodhi and Norris [30] decompose the CCR problem into two steps, first shift patterns generation for the entire roster (rota scheduling) and, second assigning duties to the resulting patterns(duty sequencing). Rota scheduling aims at maximizing crew satisfaction by the weighted sum of regular weekends, of pairs of consecutive days off not including weekends, and of long weekends. A network MILP model is used to find an optimal cyclic path to cover the roster. The objective of duty sequencing is to minimize the violation of soft constraints. It is applied at London Underground. Xie et al. [34] propose a stochastic model for the rota scheduling problem in public bus transit in order to reduce the discrepancy between a planned roster and the actual one. Standbys are normally evenly planned for all drivers (e.g., in [17]) without considering more detailed information such as historical or weekly-depending sickness absence rates. In case that the absence rate exceeds the available standbys, additional drivers are called in manually. This causes discontent for the drivers as well as continuous organizational effort for the bus company. Xie et al. [33] propose a duty-block network for solving the shift-duty assignment problem. Optimal rosters are characterized by the maximal satisfaction of drivers, the minimal difference of overtime among all drivers, and the minimal number of unassigned duties. The model aims at minimizing the punishment costs for the undesired jobs for each driver while minimizing the maximum overtime of all drivers. The unassigned duties are also described as undesired duties with high punishment cost. In order to deal with multi-objectives, the former objective is included in the model of the duty-block network and improved by simulated annealing by considering the latter objective. The generated solution is shown to outperform the sequential approach.

Additionally, the CCR is widely used in the railway transport, due to its similarity to bus transport, i.e., many regular trips from Monday to Friday.

For example, the recent publications in [18] and [23] solve the sequencing sub-problems of the CCR.

3.2 Non-cyclic crew rostering (NCCR)

Days-off scheduling for the crews of a Finnish bus transportation company is solved in [16] by using a variation of the cooperative local search method (see [2]). The objective is to find a solution that minimizes the sum of weighted soft constraint violations without violating any hard constraints. A set of artificial test instances are used in this paper. The number of employees is up to 56 and the number of total weeks is limited to 36. Based on the days off scheduling in [16], the sequential days off and shift scheduling are solved in [25] by using meta heuristic methods (simulated annealing, tabu search and ejection-chains) for a Finnish bus transportation company to minimize the weighted sum of the soft constraint violations.

Carraresi and Gallo [3] model the shift assignment as a multi-level bottleneck assignment problem by considering minimization of maximum workload, i.e., evenly distribute the workload among the crews. A heuristic algorithm has been used to get asymptotically optimal solutions. Randomly generated and real life problems for five working days (in other words, one working week) were used for testing. Bianco et al. [1] proposed a new iterative heuristic algorithm for solving the shift-duty assignment problem. It aims at finding the minimum-cost combination of rosters while minimizing the maximum roster workload. ”Roster of type” is defined as a subset of the days of the time period, such as the ordered set 4,5,6,7,1, which represents a roster covering the days between Thursday and Monday. In each iteration the weekly rosters of type (5 days) are generated by solving the multilevel bottleneck assignment problem on the subset of unassigned duties, which aims at minimizing the maximum workload. The algorithm terminates if no duties are available or it is not possible to generate rosters of any type. The solution is close to optimality, but larger computation times are needed compared with the algorithm proposed by [3].

Another model was proposed by [5] for solving duty sequencing problem. The type of rosters (or roster of type in [1]) is used to define the sequence of working days such that there is no days off in between. The problem minimizes the maximum roster duration while the total roster cost is minimized. It is solved by column generation, where the master problem is a set covering problem, while the sub-problem is modeled as a shortest path problem with additional constraints. The approach is tested on randomly generated data sets.

Evolutionary heuristics for solving bi-objective NCCR have been proposed in [24]. This work minimizes the maximum overtime of drivers while minimizing the number of drivers during the rostering period. The test instances were designed to cope with the special case of a bus transport company in Portugal (the length of the rostering period is 4 weeks; the number of drivers is 45). Another paper on multi-objective model for solving the NCCR (in [29]) used a specifically designed memetic algorithm, which combines the best features of both heuristics shown in [24]. Computational results show that the new method outperforms the results of those heuristics.

A mixed integer programming (MIP) Model for solving NCCR is provided in [20], which aims at minimizing the number of drivers while minimizing the maximum overtime of all drivers. The bi-objective problem is solved by a goal programming approach to obtain near-optimal rosters. The instances are limited up to about 1300 duties and about 70 drivers.

One of the few publications about the railway NCCR is developed by [7] for Netherlands Railways, in which the similar decomposition approach of solving the CCR problem in [23] is proposed. Moreover, a prototype software for the multiobjective railway NCCR is developed in [22].

3.3 Goal of the paper

As shown above, most of the publications concentrate on the decomposed crew rostering problem, either the CCR or NCCR. Most publications about the integrated approach are based on heuristics. Additionally, the CCR and NCCR problems are considered as two separate problems. Although the CCR has some disadvantages compared to the NCCR, it is still widely used due to its simplicity, especially in small bus companies. Therefore, the goal of this paper is to introduce a new modeling approach for integrated crew rostering problems in public bus transit. The new model is based on a multi-commodity network flow technique, and enables us to solve both CCR and NCCR problems of practical size and complexity. Therefore, we are able to solve the problems by using standard optimization software (such as Gurobi Optimizer (see [12]) and IBM ILOG CPLEX Optimizer (see [13])). Moreover, the results of sequential rota scheduling and duty sequencing are compared to the ones of the integrated crew rostering. Additionall, this is the first publication in public bus transit to consider preferences of drivers in NCCR. The cyclic crew rostering is restricted by its structure, therefore, fewer preferences can be considered. The decomposed cyclic crew rostering problem in public transit, such as [17], [9], [30], and [33] consider the preferred combination of shifts.

4 Solution Approach

We first describe the network in Section 4.1. We have the same network design for the sub-problem rota scheduling as well as the integrated problem, for both CCR and NCCR problems. Based on the network, an overview of the mathematical formulations for solving the decomposed and integrated NCCR problems will be given in Section 4.2. The small modification of the network and models is then presented in Section 4.3 to solve the CCR Problems.

4.1 The underlying network

The NCCR problem in this work is formulated as a multicommodity flow network similar to the model developed for the airline crew rostering problem by [4]. Each network layer represents the valid activities of each day for a driver, and possible connections between them. Recall that an activity can be a shift, a standby, a day off, or a fixed activity, such as a training period, or annual leave. The activity arcs are only connected if they are compatible (not in forbidden sequences), i.e., the sequence of both activities does not violate any work regulation, such as the minimum rest periods between two consecutive shifts. A path in the network corresponds to a schedule for an employee for the planning horizon (for example: one month). On each network layer, the cost of each activity is defined to reflect the daily desire of the driver, while the cost of each connection arc is used to reflect the combination desire of activities of the driver. An example for the combination desire of a driver is that similar shifts are connected. Less desired activity arcs and combination arcs receive a higher cost. An illustration of the network layer of driver d1 for the example in Figure 3 is shown in Figure 4, in which the different types of activity arcs are illustrated as nodes for illustration purposes. The nodes with letter E, M, L represent sorted shift types: early (ES), midday (MS), late shifts (LS) respectively while the ones with number 1 and 2 represent the single-offs and double-offs respectively. The marked nodes are the activities that driver d1 prefers. Additionally, a start and end node for the path of the driver are defined on each layer.

The main differences between the network of the crew rostering problem for buses and it for airlines in [4] is summaried as follows, so that the model of [4] is extended to model our problem. A set of additional soft constraints is integrated, which is necessary for the bus crew rostering (see quality rules in 2). Besides that, one difference to the network model in [4] is that the costs of the activities and the combination of activities depend on the personal preferences of the employees. Furthermore, some constraints such as forbidden sequences can get extended to more than two consecutive days in bus traffic. Additionally, the previous planning period is considered as well.

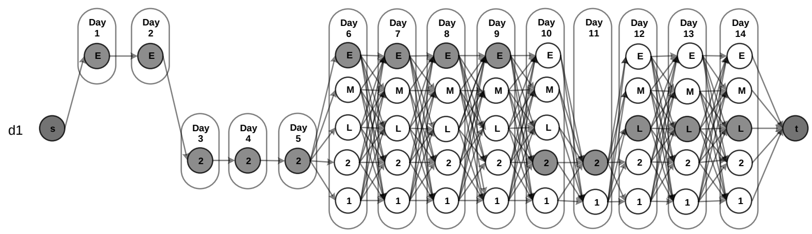


Fig. 4: Illustration of the network layer for the NCCR problem (driver d1).

Only the feasible activities are generated as activity arcs for each driver. If a fixed activity exists on one day, then there is only one activity on that day, and one connection arc between two fixed activities is generated. One exception is a fixed single day off on the actual planning horizon, for example, the fixed day off on 11th day in Figure 4. The single and double-off activities on that day are generated for the open case, whether a day off on day 10 or 12 will be selected. The following techniques are used to reduce the complexity of the network.

- We assume the maximum block of days off is three days. Therefore, on the 6th day only the work-related activities are possible and single and double-off activities are deleted (see Figure 5). Moreover, only the compatible and desirable connections for each two consecutive days are generated as connection arcs. Forbidden sequences of length two and those that violate a single-off or double-off are eliminated. The forbidden sequences of length two is in this example include (LS, ES) and (ES, LS).
- The activity arcs without incoming or outgoing arcs should be eliminated. The single-off on day 10, 12 of driver d1 can be eliminated. Figure 6 shows the reduced network for drivers d1 and d2.

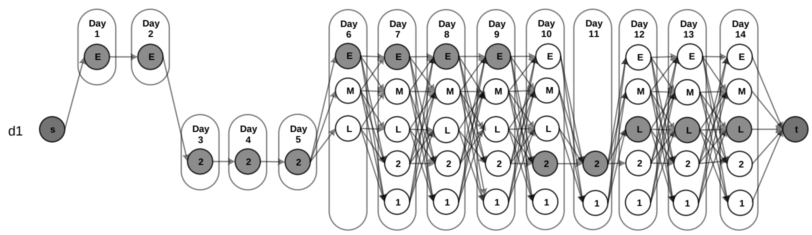


Fig. 5: Illustration of the reduced network layer of Figure 4.

The same network design, and the reducing techniques for NCCR, can be used for solving the CCR problem, but with the following modifications.

- We consider each driver group as one driver.
- For each driver there might be a different planning period, which is calculated as the number of drivers in the driver group multiplied by the number of weekdays within a week (i.e., seven).
- The capacity of each work-related/day off activity is no longer restricted by days, but by weekdays (see vertical rules in Section 2).
- Due to the properties of CCR, the compatibility of the assigned activity on the last day with the one on the first day should be checked. Additionally, the length of work days, days off blocks as well as the length of maximal distance between two double-off arcs should be checked. In order to retain the acyclic network, a sequence of artificial days is appended to the planning

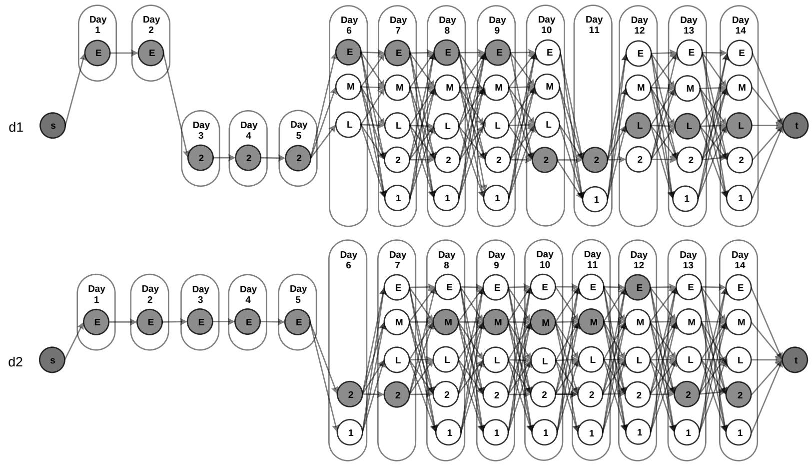


Fig. 6: Illustration of the reduced network layers of the example in Figure 3.

period whose length is equal to the largest block length we consider (such as the maximal distance between two double-off arcs). The available activity arcs of these artificial days are equal to those at the beginning of the planning period.

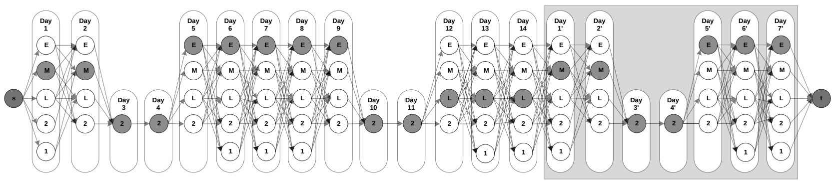


Fig. 7: Illustration of the network layer of the CCR problem in Figure 2.

We have two design variations of the network for the integrated problem. The first variation is to generate each duty instead of shift as an activity arc. The problem of this is that the network explodes, and most of the constraints we consider ignore the duty information, except the objective of minimizing maximum working time. The second variation is to maintain the network used for solving the sub-problems. However, some forbidden sequences of length two due to violating the required overnight rest time still remain in the network. For example, the forbidden sequence (LS, ES) remains in Figure 6 and 7 for the integrated approach. Moreover, some constraints about the feasibility of assigning duties are added to the network model. The details can be found in Section 4.2.3 and 4.3.3. The comparison of both variations of generating compatible arcs in the network of the integrated approach will be shown in 5.

4.2 The mathematical model of NCCR

The models for the sequential rota scheduling and duty sequencing, as well as the integrated models of NCCR are shown. That includes the basic and specialized index-sets, all necessary parameters and variables as well as the constraints and the objective function.

4.2.1 Rota scheduling for NCCR

Index-sets

$\mathcal{M}$	Set of all employees.
$\mathcal{T}_m$	Partially ordered set of all discrete time-elements t (here: days) of the planning horizon, which means $\mathcal{T}_m = \{1, \ldots, T_m\}\ \forall m \in \mathcal{M}$.
$\mathcal{D}$	Set of all activities d.
$\mathcal{D}_t \subseteq \mathcal{D}$	Set of all available activities on day $t \in \mathcal{T}_m$.
$\mathcal{D}_w \subseteq \mathcal{D}$	Set of all work-related activities.
$\mathcal{D}_b \subseteq \mathcal{D}_w$	Set of all stand by activities.
$\mathcal{D}_s \subseteq \mathcal{D}_w$	Set of all shift activities.
$\mathcal{D}_f \subseteq \mathcal{D}$	Set of all day off activities.
$\mathcal{D}_{df} \subseteq \mathcal{D}_f$	Set of all double-off activities.
$\mathcal{D}_{sf} \subseteq \mathcal{D}_f$	Set of all single-off activities.
$\mathcal{D}_{mf} \subseteq \mathcal{D}_f$	Set of all moved day off activities.
$\mathcal{D}_{mfw} \subseteq \mathcal{D}_f$	Set of all moved day off activities at a weekend.
$\mathcal{D}_{du} \subseteq \mathcal{D}$	Set of all dummy activities.
$\mathcal{E} = \mathcal{A} \cup \mathcal{C}$	Set of all edges e. This set include the set of all activity arcs $\mathcal{A}$ implied by the given activities with the set of all compatibility arcs $\mathcal{C}$.
$\mathcal{A} \subset \mathcal{E}$	Set of all activity arcs.
$\mathcal{A}^s \subset \mathcal{E}$	Set of all activity arcs e including the extended source e_m^s and sink-edges e_m^e per employee $m \in \mathcal{M}$. That is, $\mathcal{A}^s = \bigcup_{m \in \mathcal{M}} \{e_m^s, e_m^e\} \cup \mathcal{A}$
$\mathcal{A}_{m,t,d} \subseteq \mathcal{A}$	Subset of $\mathcal{A}$ that only contains the activities available to a given employee $m \in \mathcal{M}$ on day $t \in \mathcal{T}_m$ referring to an activity $d \in \mathcal{D}$.
$\mathcal{C} \subset \mathcal{E}$	Set of all compatibility arcs e, which connect compatible activity-arcs.
$\mathcal{F}_e \subset \mathcal{C}$	Set of all compatibility-arcs outgoing from activity-arc e, which link to all compatible activity arcs of the following day for a specific employee. The so-called “Forward-Star” (see also [4]), with $e \in \mathcal{A}^s$.

$\mathcal{B}_e \subset \mathcal{C}$	Set of all compatibility arcs incoming to activity arc e, which link to all compatible activity arcs of the preceding day for a specific employee. The so-called "Backward-Star" (see also [4]) with $e \in \mathcal{A}^s$.
$\mathcal{O} \subseteq \mathcal{C}$	Contains all compatibility arcs e that are forbidden due to incompatibility.
$\mathcal{S}$	Contains all sets s of sequencing activity arcs, which are forbidden to be simultaneously active in a feasible solution. This means, for example, $(e_i, \dots, e_j) \in \mathcal{S}$ with $e_i, \dots, e_j \in \mathcal{A}$. These sets of edges are implied by the given forbidden sequences. Due to the reduction in the network is each $\|s\| > 2$.

Parameters

T_m	Count of days of the planning horizon for employee m.
D	Maximal distance between two double-off arcs.
H	Minimal rest hours between two work-related activities.
L_m^R	Maximal length of consecutive day off activities for employee m.
L_m^W	Maximal length of consecutive work-related activities for employee m.
C^d	Penalty cost for not satisfying the maximal distance, defined by D, between two double-off activities.
C^f	Penalty cost for falling short of the minimal amount of day off activities per employee.
C^g	Penalty cost for exceeding the maximal amount of single-off activities per employee.
C^p	Penalty cost for falling below regarding the minimal count of standbys of an employee.
C^u	Penalty cost for an unallocated shift/duty.
C^{m1}	Penalty cost for activating an activity which indicates a moved day off activity.
C^{m2}	Penalty cost for activating an activity which indicates a moved day off activity at a weekend.
C^o	Penalty cost for the maximum overtime of all drivers.
C_e^a	Cost which appear when using the activity e.
C^{du}	Penalty cost for an unassigned day per employee.
I_d	Count of units of an activity $d \in \mathcal{D}_w \cup \mathcal{D}_f$ which are planned to get allocated.
W_d	Work time consumed when using an activity belonging to the work-related activity $d \in \mathcal{D}_w$.
T_d	The day of the work-related activity $d \in \mathcal{D}_w$.
F_m	Minimal count of days off for employee m.
E_m	Maximal count of single-off activities for employee m.
V_m	Minimal count of standbys for employee m.

R_m	Desired amount of working hours for employee m. When exceeding this value for the correlating employee the resulting surplus will be considered overtime.

Variables

x_e^a	Binary variable that depicts the flow over the activity arc $e \in \mathcal{A}^s$.
x_e^c	Binary variable, which depicts the flow over the compatibility arc $e \in \mathcal{C}$.
e_m^s	Binary variable for defining the source edge of employee $m \in \mathcal{M}$.
e_m^e	Binary variable for defining the sink edge of employee $m \in \mathcal{M}$.
σ_m	Sum of the days off for the employee m described through an integer-variable with the bounds $l = 0$ and $u = T^m$.
v_m	Binary variable that indicates if the sum of days off is satisfactory for the corresponding employee m.
τ_m	Sum of single-off activities for the employee m depicted by an integer variable with the bounds $l = 0$ and $u = T^m$.
w_m	Binary variable that indicates if the sum of single-off activities is overflown for the employee m.
ψ_m	Sum of the standbys of the employee m. This integer variable is bounded between $l = 0$ and $u = T^m$.
p_m	Binary variable, which indicates if the sum of used standbys is falling short for employee m.
λ_d	Sum of not allocated units of activity d depicted by an integer variable within the bounds $l = 0$ and $u = I_d$.
d_m	Binary variable, which indicates if a violation of the maximal distance between two double-off activities is present for the employee m.
ϕ	Maximal overtime over all employees as a positive continuous variable.
z_m	Integer variable that indicate the sum of open days for the employee m, since it is possible that there are some days without assigned activities for the employee m.

Hard constraints

As described above, the rota scheduling is formulated as a multi-commodity network flow problem, which contains the following *network constraints*. The set of constraints (1) makes sure that the value of an activity arc matches the sum of all compatibility arcs of the backward-star as well as the sum of all compatibility arcs of the forward-star (*flow conservation*), while the set of constraints (2) ensures that exactly one activity is selected for each driver per day (*demand satisfaction*). The units of each activity are restricted in the *capacity constraint* (7). In each network layer, a resource constrained shortest path has to be determined. The *resources* include the maximal block length of work-related days (see constraint set (3)) as well as days off for each driver (see constraint set (4)). Additionally, the double-off as well as forbidden sequences are defined in the constraints (5) and (6), respectively. For example, if a sequence $s = (e1, e2, e3)$ is forbidden, then is $x^a_{e1} + x^a_{e2} + x^a_{e3} \leq 2$.

$$x^a_e = \sum_{e' \in \mathcal{F}_e} x^c_{e'} \qquad \forall e \in \mathcal{A}^s \tag{1}$$

$$x^a_e = \sum_{e' \in \mathcal{B}_e} x^c_{e'} \qquad \forall e \in \mathcal{A}^s$$

$$e^s_m = 1 \qquad \forall m \in \mathcal{M} \tag{2}$$

$$e^e_m = 1 \qquad \forall m \in \mathcal{M}$$

$$\sum_{t'=t}^{t+L^W_m} \sum_{e \in \mathcal{A}_{m,t',d}} x^a_e \geq 1 \qquad \forall m \in \mathcal{M}, t \in \{1, ..., T_m - L^W_m\}, d \in \mathcal{D}_f \tag{3}$$

$$\sum_{t'=t}^{t+L^R_m} \sum_{e \in \mathcal{A}_{m,t',d}} x^a_e \geq 1 \qquad \forall m \in \mathcal{M}, t \in \{1, ..., T_m - L^R_m\}, d \in \mathcal{D}_w \tag{4}$$

$$\sum_{e \in \mathcal{A}_{m,t,d}} x^a_e \geq \sum_{e \in \mathcal{A}_{m,t+1,d}} x^a_e \qquad \forall m \in \mathcal{M}, t \in \{1, ..., T_m - 1\}, d \in \mathcal{D}_{df} \tag{5}$$

$$\sum_{e \in s} x^a_e \leq |s| - 1 \qquad \forall s \in \mathcal{S} \tag{6}$$

$$\sum_{m \in \mathcal{M}} \sum_{e \in \mathcal{A}_{m,t,d}} x^a_e \leq I_d \qquad \forall d \in \mathcal{D}_w \cup \mathcal{D}_f, t = T_d \tag{7}$$

Soft constraints

Besides the hard constraints listed above, some constraints are soft and can be weighted by applying specific penalty costs to them. The number of unassigned shifts is counted as λ_d in the constraint set (8). The fulfillment of the minimal sum of days off per driver is achieved by the constraint set (9). The first group of constraints sums the count of days off per driver, while the second group determines whether the minimal sum of days off is met. The constraint set (10) limits the number of unpopular single-off activities. Furthermore, a maximal distance between two double-off activities should be satisfied (see constraint set (11)). Additionally, if a given number of standbys falls short for a driver, the corresponding penalty costs will be added to the objective function (see constraint set (12)). In order to avoid the case that there is no activity possible for a driver on a day, a dummy activity is selected. In (13), the sum of all open days for driver m is stored in the integer variable z_m. Moreover, in order to prevent the large differences of overtimes among drivers, the constraint set (14) defines the maximal overtime of all drivers. Note that the working time of each shift type is defined as the average working time of the duties with the particular shift type.

$$I_d - \sum_{m\in\mathcal{M}} \sum_{e\in\mathcal{A}_{m,t,d}} x_e^a = \lambda_d \qquad \forall d \in \mathcal{D}_s, t = T_d \tag{8}$$

$$\sum_{e\in\mathcal{A}_{m,t,d}} x_e^a = \sigma_m \quad \forall m \in \mathcal{M}, t \in \mathcal{T}_m, d \in \mathcal{D}_f \tag{9}$$

$$F_m - v_m \cdot T^m \leq \sigma_m \quad \forall m \in \mathcal{M}$$

$$\sum_{e\in\mathcal{A}_{m,t,d}} x_e^a = \tau_m \quad \forall m \in \mathcal{M}, t \in \mathcal{T}_m, d \in \mathcal{D}_{sf} \tag{10}$$

$$E_m + w_m \cdot T^m \geq \tau_m \quad \forall m \in \mathcal{M}$$

$$\sum_{t'=t}^{t+D} \sum_{e\in\mathcal{A}_{m,t',d}} x_e^a \geq 1 - d_m \quad \forall m \in \mathcal{M}, d \in \mathcal{D}_{df}, t \in \{1, \ldots, T_m - D\} \tag{11}$$

$$\sum_{e\in\mathcal{A}_{m,t,d}} x_e^a = \psi_m \quad \forall m \in \mathcal{M}, t \in \mathcal{T}_m, d \in \mathcal{D}_b \tag{12}$$

$$\psi_m \geq V_m - p_m \cdot V_m \quad \forall m \in \mathcal{M}$$

$$\sum_{t\in\mathcal{T}_m} \sum_{d\in\mathcal{D}_{du}} \sum_{e\in\mathcal{A}_{m,t,d}} x_e^a = z_m \quad \forall m \in \mathcal{M} \tag{13}$$

$$\sum_{t\in\mathcal{T}_m} \sum_{d\in\mathcal{D}_w} \sum_{e\in\mathcal{A}_{m,t,d}} W_d \cdot x_e^a - R_m \leq \phi \quad \forall m \in \mathcal{M} \tag{14}$$

Objective function

There are three following types of objectives, which typically have to be considered in crew rostering problems.

1. Objectives related to real costs
   The number of unassigned activities, such as shift (22) or standby activities (21), and the number of open days (23) should be minimized in the objective function, since additional costs have to be paid, e.g., for hiring new (part-time) drivers or asking drivers to work in overtime. Additionally, the overtimes should be paid. In some cases it should be avoided, for example, that some drivers work overtime while others work substantially less than the regular working time (also a fairness aspect). Therefore, the maximum overtime among all drivers should be minimized in the objective (24).
2. Objectives related to the fairness among drivers
   One aspect of even distribution of workload (overtime) is considered in the last category (24). The other one, i.e., the given number of days off for each driver, should be satisfied as far as possible (18). Moreover, the other aspects of fairness, such as the distribution of unpopular activities, are embedded in the preferences of drivers (15)(i.e., the desirability of an activity on one particular day).
3. Objectives related to preferences
   One cost category reflects the desirability of an activity on one particular

day. This category includes the costs of selected activities (15) and the costs for moved days off (16) (at weekends (17)). Another cost reflects the desirability of the frequency of a desired/undesired activity, which contains the costs of underrun days off (18) and the costs for overflown single-off activities (19). Additionally, the violation of the distance between two double-off activities should be minimized (20).

$$\min z^R_{NCCR}$$
$$= \sum_{e \in \mathcal{A}^s} C^a_e \cdot x^a_e \qquad \text{(Costs of selected activities)} \quad (15)$$
$$+ \sum_{m \in \mathcal{M}} \sum_{d \in \mathcal{D}_{mf}} \sum_{e \in \mathcal{A}_{m,T_d,d}} C^{m1} \cdot x^a_e \qquad \text{(Moved days off)} \quad (16)$$
$$+ \sum_{m \in \mathcal{M}} \sum_{d \in \mathcal{D}_{mfw}} \sum_{e \in \mathcal{A}_{m,T_d,d}} C^{m2} \cdot x^a_e \qquad \text{(Moved days off at a weekend)} \quad (17)$$
$$+ \sum_{m \in \mathcal{M}} C^f \cdot v_m \qquad \text{(Costs of underrun days off)} \quad (18)$$
$$+ \sum_{m \in \mathcal{M}} C^g \cdot w_m \qquad \text{(Costs for overflown single-offs)} \quad (19)$$
$$+ \sum_{m \in \mathcal{M}} C^d \cdot \delta_m \qquad \text{(Violated dist. btwn double-offs)} \quad (20)$$
$$+ \sum_{m \in \mathcal{M}} C^p \cdot p_m \qquad \text{(Underrun standbys)} \quad (21)$$
$$+ \sum_{d \in \mathcal{D}_s} C^u \cdot \lambda_d \qquad \text{(Costs of unassigned shifts)} \quad (22)$$
$$+ \sum_{m \in \mathcal{M}} C^{du} \cdot z_m \qquad \text{(Costs of open days)} \quad (23)$$
$$+ C^o \cdot \phi \qquad \text{(Overtime)} \quad (24)$$

4.2.2 Duty sequencing for the NCCR

Let the set of duties be $\mathcal{J}$ and $\mathcal{J}_e$ is the set of duties belonging to the shift activity arc e, where $e \in \mathcal{A}^{a,w}_m$, which is defined as the set of selected shift activities after solving the rota scheduling for the driver $m \in \mathcal{M}$. A set $\mathcal{A}^w_m$ is defined as all assigned work-related activities, which contains the subset $\mathcal{A}^{a,w}_m$. Let T_e be the day of the activity arc $e \in \mathcal{A}^{a,w}_m$, and $\mathcal{A}_j$ be a set of assigned shift activity arcs, which contain the duty j. We define $\mathcal{S}^o$ as the set of sequencing duties, which are forbidden to be simultaneously active in a feasible solution due to violating the overnight rest period. Let $y_{e,j}$ be binary variable, which is equal to one, if the duty j is assigned to the selected shift activity arc $e \in \mathcal{A}^{a,w}_m$. Additionally, for each assigned shift activity a binary variable for the open day is defined as z_e, and each of them should be punished in the objective function with the cost factor C^{du}, which is the same for the open days in the rota scheduling. Moreover, a parameter W_j is defined as the working time of the duty $j \in \mathcal{J}$, while a parameter W is defined as the standard working time of each standby activity $d \in \mathcal{D}_b$.

Hard restrictions

Each duty might be available to several shift activity arcs, therefore, the constraints (25) make sure that each duty is selected at most once. The constraints (26) forbid the impossible combination of duties.

$$\sum_{e \in \mathcal{A}_j} y_{e,j} \leq 1 \qquad \forall j \in \mathcal{J} \tag{25}$$

$$y_{e,j} + y_{e',j'} \leq 1 \qquad \forall e, e' \in \mathcal{A}_m^{a,w}, m \in \mathcal{M}, T_e = T_{e'} - 1, (j, j') \in \mathcal{S}^o \tag{26}$$

Soft restrictions

The constraints (27) ensure the relation between duty and the selected shift activity arc, i.e., one duty or dummy is needed for each selected shift activity arc. The dummies are needed due to the sequencing approach. The constraints $(14a)$ re-define the calculation of maximal overtime of all drivers.

$$x_e^a = \sum_{j \in \mathcal{J}_e} y_{e,j} + z_e \quad \forall e \in \mathcal{A}_m^{a,w}, m \in \mathcal{M} \tag{27}$$

$$\sum_{e \in \mathcal{A}_m^w \setminus \mathcal{A}_m^{a,w}} W \cdot x_e^a + \sum_{e \in \mathcal{A}_m^{a,w}} \sum_{j \in \mathcal{J}_e} W_j \cdot y_{e,j} - R_m \leq \phi' \qquad \forall m \in \mathcal{M} \tag{14a}$$

Objectives

The duty sequencing problem aims to assign duties to the selected shifts fairly without violating any constraints. The fairness refers to minimizing the maximum overtime. Additionally, some more open days, or more precisely, some unassigned duties might be caused.

$$\min z_{NCCR}^D = C^o \cdot \phi' \qquad \text{(Overtime)} \tag{24a}$$

$$+ \sum_{m \in \mathcal{M}} \sum_{e \in \mathcal{A}_m^{a,w}} C^{du} \cdot z_e \quad \text{(Open days/unassigned duties)} \tag{28}$$

4.2.3 Integrated problem for the NCCR

Recall that the set of duties is defined as $\mathcal{J}$. Let $\mathcal{J}_e'$ be a set of duties belonging to the activity arc e, where $e \in \mathcal{A}_m^a, m \in \mathcal{M}$. The set $\mathcal{A}_j'$ is a set of possible shift activity arcs, to which the duty j can be assigned. Let $y_{e,j}'$ be a binary variable that is equal to 1 if the duty j is assigned to the activity arc $e \in \mathcal{A}_m^a$ of the driver m, and 0 otherwise. Additionally, let $\mathcal{S}_m^o$ be a superset of all sets s of sequencing duties (length 2) for the driver m, which are forbidden to be simultaneously active in a feasible solution due to violating the overnight rest period.

Hard restrictions

The set of hard constraints (1) to (7) remain the same as in Section 4.2.1. Additionally, each duty might belong to several shift activity arcs, therefore, the constraints (25a) make sure that each duty is selected at most once. Moreover, the constraints (26b) forbid the impossible combination of duties. Constraints (27a) ensure each assigned shift activity arc get a duty.

$$(1), (2), (3), (4), (5), (6), (7)$$

$$\sum_{e \in \mathcal{A}'_j} y'_{e,j} \leq 1 \qquad \forall j \in \mathcal{J} \tag{25a}$$

$$y'_{e,j} + y'_{e',j'} \leq 1 \qquad \forall e \in \mathcal{A}'_j, e' \in \mathcal{A}'_{j'}, (j, j') \in \mathcal{S}^o_m, m \in \mathcal{M} \tag{26b}$$

$$x^a_e = \sum_{j \in \mathcal{J}'_e} y'_{e,j} \qquad \forall e \in \mathcal{A}^a_m, m \in \mathcal{M} \tag{27a}$$

Soft restrictions

The set of soft constraints (8) to (13) in Section 4.2.1 remain the same. Additionally, the constraints (14b) replace (14) and re-define the calculation of maximal overtime of all drivers.

$$(8), (9), (10), (11), (12), (13)$$

$$\sum_{d \in \mathcal{D}_w \setminus \mathcal{D}_s} \sum_{e \in \mathcal{A}_{m,T_d,d}} W_d \cdot x^a_e + \sum_{e \in \mathcal{A}^a_m} \sum_{j \in \mathcal{J}'_e} W_j \cdot y'_{e,j} - R_m \leq \phi'' \quad \forall m \in \mathcal{M} \tag{14b}$$

Objectives

The objectives (15) to (23) remain. The new calculated maximum overtime ϕ'' is minimized in the objective function.

4.3 Modifications to the CCR problem

The network model of CCR is similar to the one of NCCR. The differences to the network of the NCCR problem is shown in Section 4.1. In this section, the modifications of the mathematical models in Section 4.2 for the CCR problem are shown, both for the sequential and integrated approaches.

4.3.1 Rota scheduling for the CCR

Hard restrictions

As mentioned before, the work-related activity capacities are no longer restricted by days, but by weekdays (7a). Therefore, $T_d{}'$ is defined as the weekday of the work-related activity $d \in \mathcal{D}_w$. Note that $\mathcal{T}_m$ is defined as the days of driver m in network, including the artifical days for the CCR problem. A set $\mathcal{T}_{m,t}$ is used to define the days (but not including artifical days) in the network of the weekday t for the driver $m \in \mathcal{M}$. For example, the weekday t is equal to 1, so a set $\mathcal{T}_{m,t}$ might be $\{1, 8, 15, ...\}$. The length of the artifical days for each driver m is stored in b_m^{art}. Constraints (29) make sure the selected activity on each artifical day $T_m - b_m^{art} + i$ is equal to the selected one on the day i, where $i \in \{1, 2, ..., b_m^{art}\}$. The rest remains the same as in Section 4.2.1.

$$(1), (2), (3), (4), (9), (6)$$

$$\sum_{m\in\mathcal{M}} \sum_{t'\in\mathcal{T}_{m,t}} \sum_{e\in\mathcal{A}_{m,t',d}} x_e^a \leq I_d \qquad \forall d \in \mathcal{D}_w, t = T_d{}' \tag{7a}$$

$$x_e^a = x_{e'}^a \qquad \forall m \in \mathcal{M}, i \in \{1, 2, ..., b_m^{art}\} \quad d \in \mathcal{D}_i, e \in \mathcal{A}_{m,T_m - b_m^{art}+i,d}, e' \in \mathcal{A}_{m,i,d} \tag{29}$$

Soft restrictions

$\mathcal{T}_m^{art} \subset \mathcal{T}_m$ is defined as the set of artifical days in the network of driver m. Additionally, the artificial days should be ignored by the constraints of calculations of unassigned shifts ($8a$), underrun days off ($9a$), overflown single-offs ($10a$), underun of standbys ($12a$), the unassigned days ($13a$), as well as the maximum of overtime ($14c$). The constraint set (11) remains the same.

$$(11)$$

$$I_d - \sum_{m\in\mathcal{M}} \sum_{t'\in\mathcal{T}_{m,t}} \sum_{e\in\mathcal{A}_{m,t',d}} x_e^a = \lambda_d{}' \qquad \forall d \in \mathcal{D}_s, t = T_d{}' \tag{8a}$$

$$\sum_{e\in\mathcal{A}_{m,t,d}} x_e^a = \sigma_m{}' \qquad \forall m \in \mathcal{M}, d \in \mathcal{D}_f, t \in \mathcal{T}_m \setminus \mathcal{T}_m^{art} \tag{9a}$$

$$F_m - v_m{}' \cdot T^m \leq \sigma_m{}' \qquad \forall m \in \mathcal{M}$$

$$\sum_{e\in\mathcal{A}_{m,t,d}} x_e^a = \tau_m{}' \qquad \forall m \in \mathcal{M}, d \in \mathcal{D}_{sf}, t \in \mathcal{T}_m \setminus \mathcal{T}_m^{art} \tag{10a}$$

$$E_m + w_m{}' \cdot T^m \geq \tau_m{}' \qquad \forall m \in \mathcal{M}$$

$$\sum_{e\in\mathcal{A}_{m,t,d}} x_e^a = \psi_m{}' \qquad \forall m \in \mathcal{M}, d \in \mathcal{D}_b, t \in \mathcal{T}_m \setminus \mathcal{T}_m^{art} \tag{12a}$$

$$\psi_m{}' \geq V_m - p_m{}' \cdot V_m \qquad \forall m \in \mathcal{M}$$

$$\sum_{t\in\mathcal{T}_m\setminus\mathcal{T}_m^{art}} \sum_{d\in\mathcal{D}_{du}} \sum_{e\in\mathcal{A}_{m,t,d}} x_e^a = z_m{}' \qquad \forall m \in \mathcal{M} \tag{13a}$$

$$\sum_{t\in\mathcal{T}_m\setminus\mathcal{T}_m^{art}} \sum_{d\in\mathcal{D}_w} \sum_{e\in\mathcal{A}_{m,t,d}} W_d \cdot x_e^a - R_m \leq \phi''' \qquad \forall m \in \mathcal{M} \tag{14c}$$

Objectives

The objective is modified corresponding to the altered soft constraints.

4.3.2 Duty sequencing for the CCR

We assume that the set $\mathcal{A}_j$, $\mathcal{A}_m^{a,w}$, and $\mathcal{A}_m^w$ do not contain the arcs on artifical days. Therefore, the constraints (25), (27), (14a), as well as the objective remain the same as in Section 4.2.2. Let $\mathcal{A}_m^{a,art}$ be the assigned shift activity arcs in the artifical days for driver m.

Hard restrictions

The combination of duties should be also checked in the artifical days, therefore, the constraints (26) should be changed to (26a). Moreover, a set of constraints in (30) make sure that the same duties are assigned in the artifical days and the original days.

$$(25)$$

$$y_{e,j} + y_{e',j'} \leq 1 \qquad \forall e, e' \in \mathcal{A}_m^{a,w} \cup \mathcal{A}_m^{a,art}, T_e = T_{e'} - 1, \quad (j, j') \in \mathcal{S}^o, m \in \mathcal{M} \tag{26a}$$

$$y_{e,j} = y_{e',j} \qquad \forall m \in \mathcal{M}, T_e \in \{1, 2, ..., b_m^{art}\}, e \in \mathcal{A}_m^{a,w}, \quad e' \in \mathcal{A}_m^{a,art}, T_{e'} = T_m - b_m^{art} + T_e, j \in \mathcal{J}_e \tag{30}$$

Soft restrictions

Since the set $\mathcal{A}_m^{a,w}$ and $\mathcal{A}_m^w$ do not contain the arcs on artifical days, therefore, the constraints (27) about the relation between duty and the selected shift activity arc and the constraints (14a) of defining maximum overtime of all drivers remain.

Objectives The objective remains as same as the one in duty sequencing for the NCCR.

4.3.3 Integrated problem for the CCR

The model for CCR in integrated planning is considered as an extended model of the rota scheduling in Section 4.3.1. The hard and soft constraints about the feasibility of duties are extended.

Hard restrictions

The constraints (1) to (29) in Section 4.3.1 remain the same. We assume that $\mathcal{A}'_j$ do not contain the shift activities on the artifical days. Therefore, the constraints (25a), (26b), (27a) from Section 4.2.3 remain. (30a) replaces (30) to make sure the assigned duties on the artificial days are equal to the original days.

$$(1), ..., (6), (7a), (29)(25a), (26b), (27a)$$

$$y'_{e,j} = y'_{e',j} \quad \forall m \in \mathcal{M}, T_e \in \{1, 2, ..., b_m^{art}\}, e, e' \in \mathcal{A}_m^a, T_{e'} = T_m - b_m^{art} + T_e, j \in \mathcal{J}'_e \tag{30a}$$

Soft restrictions

The soft constraints in Section 4.3.1 from ($8a$) to ($13a$) remain for the formulation of this problem. Additionally, the calculation of maximum overtime should avoid the duties on artifical days in constraint set ($14d$). The relation between a duty and its possible shift activity arcs is the same as in Section 4.2.3.

$$(8a), (9a), (10a), (11), (12a), (13a)$$

$$\sum_{t \in \mathcal{T}_m \setminus \mathcal{T}_m^{art}} \sum_{d \in \mathcal{D}_w \setminus \mathcal{D}_s} \sum_{e \in \mathcal{A}_{m,t,d}} W_d \cdot x_e^a + \sum_{e \in \mathcal{A}_m^a} \sum_{j \in \mathcal{J}_e'} W_j \cdot y_{e,j}' - R_m \leq \phi'''' \quad \forall m \in \mathcal{M} \tag{14d}$$

Objectives

All objectives in 4.3.1 are included, except a small modification of maximum overtime ϕ''''.

5 Computational experiments

In this section we begin with introducing a set of real-world instances from different German bus companies, followed by a description of the achieved solutions with different solvers, namely the Gurobi Optimizer (see [12]) and the IBM ILOG CPLEX Optimizer 12.0 (see [13]). Additionally, the comparison of the sequential and integrated approach is shown. All experiments were conducted on an Intel Core i7 2.93 GHz processor with 32 GB RAM running Windows 7 Professional (64 bit).

5.1 Properties of the real-world data instances

We test our solution approaches on 16 real-world instances for which the number of assigning duties varies between 1013 and 19486 duties. The number of drivers in the instances varies between 48 and 629. The names of the instances include the number of drivers (NCCR) / groups of drivers (CCR), the planning days as well as the number of included shift types. In Table 1, the important properties of the instances are shown, including the number of activities, i.e, duties, standbys, days off, the average alternatives, the number of activities arcs, the percent of fixed activity arcs in parentheses as well as the number of compatibility arcs. They are some of the factors causing difficulty of solving those instances. As we consider more activities (duties, standbys, days off) and drivers, more activity and compatibility arcs are generated in our network. However, if we consider more fixed activity arcs, then this problem is easier to solve (such as the instances 96-70-8 vs. 89-70-8). The instances are sorted according to the gap to optimum after 24 hours for solving the rota scheduling problem. The instances above the line can be solved (almost) optimally with CPLEX or Gurobi within a given acceptable time (here: 24 hours) in the sequential approach. We call these instances the set of optimally solvable instances (OSI). Contrast, the set of instances under the line is not optimally solved within 24 hours (unsolvable instances, USI).

Besides those factors shown in Table 1, another important factor is how many soft constraints are considered in those instances. Most of the instances in Table 1 consider all soft constraints in Section 4.2, but the instances 392-45-37, 393-45-37 and 397-40-37 only consider the objectives (15), (22), (23) as well as (24) and their corresponding soft constraints (see Table 2). Therefore, they are also located in the set of OSI. The instance 9-238-11 with 256 drivers is used to solve the CCR problem. It includes nine groups of drivers with different sizes, in which the maximum number of drivers in a group is 32, i.e., there are at most 224 planning days for a group. Seven artificial days are needed to expand the planning period, because there is a 14 days distance between two double-off activities. Therefore, we consider nine drivers in a planning period of at most 238 days. The instance size for the CCR problem is in practice usually small compared to the instance size for the NCCR problem. This instance 9-238-11 is already a large instance available from the Germany bus companies. The rests are provided to solve the NCCR problem. These instances are available at the web page `http://dsor.upb.de/crewrostering`.

Instance	Duties	Standbys	Days-off	ϕAlt.	Act.arcs(fixed)	Com.arcs
48-75-6	1,313	454	1,472	2.2	10,516 (12.7%)	34,584
52-73-6	1,288	664	1,487	2.2	11,367 (12.2%)	37,907
52-75-6	1,321	488	1,509	2.2	11,072 (15.5%)	36,113
9-238-11	1,013	303	483	4.7	6,672 (12.4%)	30,457
393-45-37	5,420	2,619	8,030	1.5	45,035 (23.8%)	145,491
392-45-37	5,815	2,342	8,364	1.3	43,367 (26.2%)	134,974
397-40-37	4,917	2,717	7,438	1.4	38,277 (26.3%)	115,236
96-70-8	3,200	876	1,936	5	38,413 (6.3%)	287,490
87-70-8	3,273	796	1,708	4.5	38,944 (4.1%)	301,374
214-45-34	3,966	1,472	3,240	6.6	55,366 (8.7%)	472,958
211-45-34	4,003	1,447	3,029	7	55,376 (8.9%)	490,521
89-70-8	3,260	671	1,758	4.8	40,392 (4.8%)	326,505
221-45-30	4,214	1,646	3,222	6.7	64,576 (7.6%)	624,520
629-46-26	11,073	4,292	9,246	5	150,616 (11.8%)	1,212,486
606-70-26	18,739	4,977	13,612	5.3	255,657 (7.5%)	2,158,525
607-70-26	19,486	4,549	13,364	5.1	258,075 (7.3%)	2,205,303

Table 1: Characteristics of the test set instances

The model we present is a multi-objective optimization model. The weight of each objective given by bus companies is shown in Table 2. The quality as well as the running time of the solution depends on the weights of the objectives. For example, we change the given C^u to be $2e7$ in instance 96-70-80, and the other weights remain. The optimal solution as well as the running time of the altered weights differs from the optimal solution determined with weights from the bus company (see Table 3). The number of unassigned shifts is reduced to 187 from 200, and the running time is reduced by about 86%.

Instance	C_e^a	C^u	C^{du}	C^f	C^{m2}	C^{m1}	C^d	C^p	C^g	C^o
48-75-6[1]	1	1e6	1e7	5e6	1e5	9e5	1.5e6	1e4	5e5	1e4
9-238-11	1	1e6	1e7	5e6	1e5	2.1e6	2.5e6	1e4	5e5	1e3
393-45-37[2]	1	1e6	1e7	0	0	0	0	0	0	1e1
96-70-8[3]	1	1e6	1e7	5e6	1e5	2.1e6	2.5e6	1e4	5e5	1e3
214-45-34	1	1e8	5e7	1.5e7	5e5	5e6	3e7	1e5	5e5	5e2
211-45-34[4]	1	5e7	5e7	2e7	1e5	1e7	1e7	1e5	5e5	1e3
629-46-26[5]	1	1e6	1e7	5e6	1e5	9e5	1.5e6	1e5	5e5	1e3

Table 2: The weights of all objectives in all instances. [1]: the same weights for the instances 52-73-6 and 52-75-6; [2]: the same weights for 392-45-37 and 397-40-37; [3]: the same for 87-70-8 and 89-70-8; [4]: the same for 221-45-34; [5]: the same for 606-70-26 and 607-70-26.

Info	96-70-8(given weights)	96-70-8(changed weights)
# Unassigned shifts	200	187
# Standbys underflow	1	1
# Used moved days off	1	13
# Used moved days off on weekends	8	13
# underrun days off	0	0
# single-off overflow	0	0
# double-off distance violations	0	2
Maximal overtime (in hours)	0.41	1.6
The selected activity costs	2963383	2741173
running time (in sec.)	27284	4069

Table 3: The comparison of solutions after solving rota scheduling for the instance 96-70-8 by using different weights of the objectives.

5.2 Results of solving the sequential planning

An overview of the size of sub-problems, i.e., rota scheduling (RS) and duty sequencing (DS), is given in Table 4. The largest rota scheduling problems with 606, 607 and 629 drivers result in an optimization problem which ranges in the class of the extra-large set of the MIPLIB 2010 (see [35]) regarding variable, constraint and non-zero count.

As shown in literature (such as [30]), the duty sequencing problem is usually easier to solve compared to the rota scheduling problem, due to the assigned shifts and other activities. Moreover, less restrictions are considered in this problem (see the model description in Section 4.2.2 and 4.3.2). The duty sequencing problem of all instances can be solved optimally within one hour, either using solver CPLEX or Gurobi. Therefore, we list only the solutions of the rota scheduling problem in Figure 8. The details about the solutions of the sequential approach are shown together with the solutions of the integrated approach in Table 7. The solution quality of this sequential approach depends highly on the solution of the rota scheduling problem. So we compare both approaches for the OSI.

In Figure 8, the solver Gurobi is suitable for the set of OSI (see Figure 8a and 8b), especially, the instance 96-70-8 and 87-70-8, since the computational time is much shorter than for the solutions obtained by the solver CPLEX.

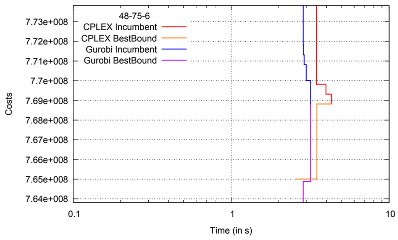
However, the solver CPLEX is the best choice for the USI (running times are limited to 24 hours), except the instance 221-45-30. In order to solve the USI with Gurobi more quickly, we perform parameter tuning. Only a small set of instances (48-75-6, 392-45-37, and 96-70-8) are chosen, due to the long running time of tuning. The default method in Gurobi for the MIP problem in this paper is 1, i.e., the dual method is chosen automatically for the MIP root node. We tried the barrier method, but it did not work appropriately on USI. Note that the barrier method (method 2 in Gurobi) is a recommendation of the tuning.

Instance	Stage	Int	Bin	Cols	Rows	NZs
48-75-6	RS	459	37,800	38,260	29,940	161,024
	DS	0	37,468	37,469	4,392	50,548
52-73-6	RS	461	41,714	42,176	32,084	174,557
	DS	0	41,613	41,614	3,969	53,764
52-75-6	RS	469	39,562	40,032	31,261	166,968
	DS	0	39,426	39,427	4,460	52,289
9-238-11	RS	93	54,210	54,304	21,479	174,660
	DS	0	100,821	100,822	61,295	254,775
393-45-37	RS	1,998	137,567	139,566	122,084	592,645
	DS	0	206,233	206,234	70,410	410,168
392-45-37	RS	1,958	128,225	130,184	116,666	558,222
	DS	0	222,085	222,086	78,042	447,784
397-40-37	RS	1,826	113,395	115,222	103,317	490,472
	DS	0	204,248	204,249	92,132	462,509
96-70-8	RS	658	264,238	264,897	154,439	1,078,099
	DS	0	235,227	235,228	46,204	374,940
87-70-8	RS	627	276,383	277,011	162,425	1,131,938
	DS	0	252,058	252,059	39,280	386,600
214-45-34	RS	1,338	435,951	437,290	185,623	1,547,708
	DS	0	312,771	312,772	42,630	438,848
211-45-34	RS	1,287	457,029	458,317	183,363	1,576,977
	DS	0	315,070	315,071	63,111	479,735
89-70-8	RS	631	303,541	304,173	175,855	1,231,574
	DS	0	261,922	261,923	46,124	409,105
221-45-30	RS	1,404	587,103	588,508	225,444	1,998,863
	DS	0	378,505	378,506	40,121	505,613
629-46-26	RS	2,507	1,095,668	1,098,176	675,251	4,446,551
	DS	0	2,197,945	2,197,946	366,693	3,456,166
606-70-26	RS	2,890	1,952,636	1,955,527	1,196,006	8,060,000
	DS	0	3,822,694	3,822,695	640,384	5,967,647
607-70-26	RS	2,916	2,000,713	2,003,630	1,239,754	8,304,197
	DS	0	3,903,682	3,903,683	285,569	5,069,953

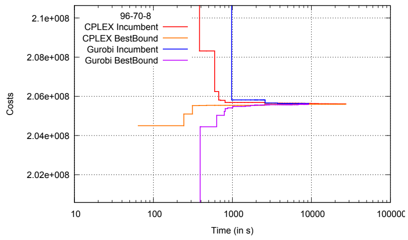
Table 4: Characteristics of the test-set optimization models of the rota scheduling (RS) and duty sequencing (DS) problems

5.3 Results of solving the integrated planning

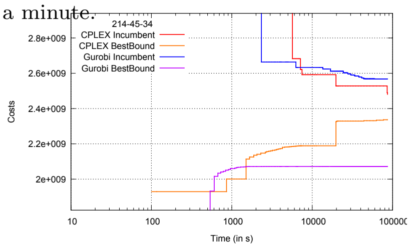
The characteristics of the models of the integrated planning are shown in Table 5. It is obvious that the integrated problem is more complex compared to sub-problems, especially the Rows and NZs.



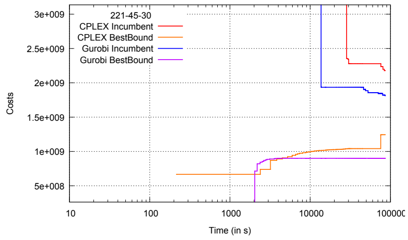
(a) Performance statistics for the instance 48-75-6, similar for the instances 52-73-6, 52-75-6, 9-238-11, 393-45-37, 392-45-37, and 397-40-37. They are solvable within a minute.



(b) Performance statistics for the instance 96-70-8, similar for the instance 87-70-8.



(c) Performance statistics for the instance 214-45-34, similar for the instances 211-45-34, 89-70-8, 629-46-26, 606-70-26, and 607-70-26, but with different gaps after 24 hours.



(d) Performance statistics for the instance 221-45-30.

Fig. 8: Performance statistics of using solvers CPLEX and Gurobi after 24 hours of computation for the rota scheduling.

Instance	Int	Bin	Cols	Rows	NZs
48-75-6	459	67,382	67,842	122,224	418,477
52-73-6	461	75,252	75,714	136,282	466,202
52-75-6	469	70,612	71,082	128,887	438,855
9-238-11	102	151,350	151,444	2,336,654	5,080,561
393-45-37	1,998	328,298	330,297	795,443	2,427,350
392-45-37	1,958	331,597	333,556	1,092,803	3,042,083
397-40-37	1,826	297,923	299,750	1,032,050	2,835,595
96-70-8	658	500,361	501,020	2,750,243	6,898,857
87-70-8	627	532,129	532,757	3,037,455	7,565,810
214-45-34	1,338	807,350	808,689	4,530,971	11,202,248
211-45-34	1,287	828,050	829,338	5,149,397	12,469,394
89-70-8	631	571,633	572,265	3,364,690	8,324,919
221-45-30	1,404	983,899	985,304	6,075,659	14,742,199

Table 5: Characteristics of the test-set optimization models of the integrated planning. The characteristics of the largest instances are unknown due to out of memory.

As shown in Table 6, we test two variations of the integrated approach. The running time of each instance in the var2 is set to be same as in the var1. The largest three instances are unsolvable for both variations due to out of memory. It seems that the instances in the USI need more running time for both variations to get a better result compared to the sequential approach.

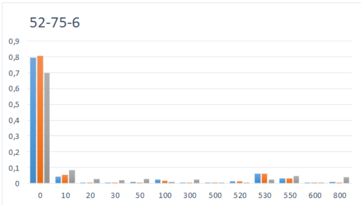
	var.1			var.2			
Instance	Obj.	Gap(%)	Time(sec.)	Obj.	Gap(%)	Time(sec.)	FS(%)
48-75-6	7.40×10^8	0	32	7.40×10^8	0	27	72.5
52-73-6	8.21×10^8	0	57	8.21×10^8	0	50	72.6
52-75-6	5.85×10^8	0	43	5.85×10^8	0	59	73.6
9-238-11	8.15×10^8	0	14,474	8.15×10^8	0	5,483	100
393-45-37	3.84×10^8	0	3,498	2.40×10^8	4.8	3,500	100
392-45-37	8.98×10^8	0	24,591	6.48×10^8	9	24,600	100
397-40-37	3.72×10^8	0	22,843	1.85×10^8	4.9	24,600	100
96-70-8	2.10×10^8	0	48,286	2.06×10^8	0.07	48,100	22.2
87-70-8	1.32×10^7	26.4	86,400	8.10×10^6	12.5	86,400	21.0
214-45-34	1.08×10^{10}	81	86,400	2.81×10^{10}	93	86,400	0
211-45-34	1.07×10^{10}	83.6	86,400	1.33×10^{11}	99	86,400	41.7
89-70-8	1.72×10^8	77.7	86,400	2.58×10^8	86.5	86,400	20.8
221-45-30	9.70×10^9	91.4	86,400	5.20×10^9	82.3	86,400	39.2

Table 6: Performance statistics of using solver CPLEX after 24 hours of computation of integrated planning. No integer solutions of the largest instances could be found because of out of memory.

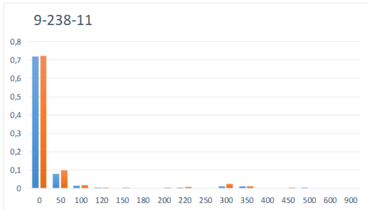
The first variation (var1) uses the same network and the same forbidden sequences as in the rota scheduling problem, additionally, the constraints about the duties are considered (see Section 4). The purpose to test var1 is to check whether it is necessary to consider the duties during the rota scheduling. In Table 7, we compare the solutions of this variation with the sequential solutions. As mentioned above, we compare both approaches for the OSI. The reason for that is to see the advantages and disadvantages of both approaches accurately. We notice that many duties are unassigned (or open days) after solving DS (duty sequencing) in the sequential approach. The var1 generally brings more assigned duties compared to that of the sequential approach (equal to: unassigned duties in RS plus those in DS). The rest of the characteristics of var1 are almost as good as that of the sequential approach, except the larger running times.

The second variation (var2) is what we presented above in Section 4, i.e., the main difference to var1 is that we do not eliminate those forbidden sequences of length two during the network generation, which violate the overnight rest time (the percentage of these forbidden sequences is shown in the column of FS in Figure 4). Similar running times as for var1 are set. The comparison to the sequential approach is shown in Table 8. Var2 is shown to outperform the sequential one and var1 in terms of the qualities of all criteria, especially for the reduction of the number of unassigned duties and open days, for example, the percentage of unassigned duties of the instance 393-45-37 is reduced from 8.6% to 4.4%. It is also expected that the running time is much longer than for the sequential approach.

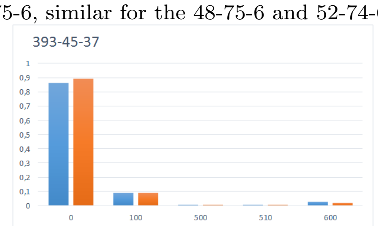
It is common that not all duties can be assigned in the optimization for real-world instances (also in the optimal solution of this integrated approach var2). One reason for that is the shortage of drivers. Some more (part time) drivers are required/hired to cover the unassigned duties. In this paper, the



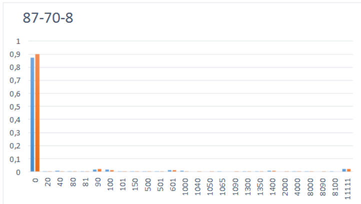
(a) The distribution of penalty costs against preferences for the instance 52-75-6, similar for the 48-75-6 and 52-74-6.



(b) The distribution of penalty costs against preferences for the instance 9-238-11.



(c) The distribution of penalty costs against preferences for the instance 393-45-37, similar for the 392-45-37 and 397-40-37.



(d) The distribution of penalty costs against preferences for the instance 87-70-8, similar for the 96-70-8.

Fig. 9: The distribution of penalty costs against the driver preferences (from left to right: seq., var2, var2 without considering preferences (optional)).

number of unassigned duties is aimed to be minimized (high punishment cost of each unassigned duties). The decision on the number of additional drivers is not part of this paper. Another reason of the unassigned duties is the large number of forbidden sequences, which might cause less compatible duties. Furthermore, a bad duty plan might be generated in the crew scheduling problem without considering the driver information. That causes as well the unassigned duties. The comparison shows that the sequential approach causes more unassigned duties as well as open days, and the integrated approach brings generally better solutions, but with longer running times.

Moreover, the costs of assigning less desirable duties reflect the effectiveness of the solution. The costs are categorized and represented in Figure 9. The cost 0 means that the most preferred activities are selected while larger cost means that less attractive activities are assigned. As expected, most preferred activities have been assigned in all solutions, both in sequential and integrated approaches. The integrated approach provides a bit more assigned activities as desired, since more duties, including the most desired ones, are not assigned in the sequential approach. For all instances, less unfulfilled preference are present for each driver. Such unfulfilled preferences could be an expected day off, etc, however, the driver could swap the undesired activity with another driver, and a small number of swaps do not reflect the quality of the solution and do not cost much operational expense. Alternatively, these drivers could be compensated by means of extra vacation, etc. If we ignore the penalty of preferences in the objective function, then more drivers get, for example in the instance 48-75-6, 52-73-6 as well as 52-75-6, less desired jobs (more than 10% activities are assigned not as desired). Additionally, the other characteristics of the solutions remain the same as in Table 8.

6 Conclusion & Outlook

In this paper, we provide a multi-commodity network flow formulation of the cyclic and non-cyclic crew rostering problems, an appropriate implementation and solver bindings for CPLEX and Gurobi. In the literature, the crew rostering problem is mostly solved with a sequential approach to deal with the size of real world instances, such as rota scheduling first and duty sequencing second. For the sequential and integrated approaches, some of the given real world instances were solved successfully in acceptable solution times regarding the size of the correlated optimization model. The integrated approach is proved to get better solutions compared to the sequential one.

Due to the given real-world instances in German bus companies, we consider only the law and labor union rules of those companies. However, our model can be modified/extended to adopt the other rules, such as the percentage of weekends-off.

However, considering some instances can not be solved optimally within 24 hours with solvers (especially with the integrated approach), a faster approach to generate feasible solutions is necessary for practical purposes. One possibility might be a column generation implementation. An explanation of the basic technique can be found in [26]. Another approach could be a meta-heuristic based method such as simulated annealing and tabu-search such as in [19]. Furthermore, an ant colony optimization based approach appears to be a good choice since it directly works on the graph by nature. Therefore, it will always result in a feasible solution regarding the flow conservation constraints. A detailed explanation of the technique can be found in [6].

Moreover, there are several objectives in our network model. The weight of each objective is provided by bus companies by experience. As shown in Table 3, a different weight-combination might bring a different solution as well as a different running time. Thus, it would be very attractive to research the multi-objective crew rostering problem (such as the simplified NCCR problem in [24]).

instance	unassigned duties(%)			# unassigned days			# standbys underflow		# moved days off		# moved days off weekends		# underrun days off		# single-off overflow		# double-off dis. vio.		maximal overtime (in hr.)		running time (in sec.)	
	RS	DS	var1	RS	DS	var1	seq.	var1	seq.	var1	seq.	var1	seq.	var1	seq.	var1	seq.	var1	seq.	var1	seq.	var1
48-75-6	0.3	1.1	0.3	0	14	0	0	1	0	0	0	0	48	48	0	0	0	0	80.7	81.2	11	32
52-73-6	0	0.5	0	0	7	0	1	2	0	0	0	0	52	52	1	1	0	0	57.9	58.4	10	57
52-75-6	0	0.5	0	0	16	0	1	2	0	0	0	0	52	52	0	0	0	0	60.8	53.9	8	43
9-238-11	1.5	4.8	1.5	8	49	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	14,474
393-45-37	6.0	2.6	7	0	141	0	0	0	0	0	0	0	0	0	0	0	0	0	81.3	81.3	81	3,448
392-45-37	14	2.9	15.4	0	165	0	0	0	0	0	0	0	0	0	0	0	0	0	82.9	81.9	78	24,505
397-40-37	6.4	2.6	7.6	0	132	0	0	0	0	0	0	0	0	0	0	0	0	0	73.9	73.9	61	22,799
96-70-8	6.3	5.4	6.4	0	174	0	1	0	1	1	8	9	0	0	0	0	0	0	0	0	27,313	47,551
87-70-8	0.2	3.8	0.2	0	127	0	12	13	0	1	3	3	0	0	0	1	0	1	1.1	0	51,907	86,445

Table 7: Detailed breakdown of the characteristics of the solutions of the integrated (var1) vs. the sequential (seq.) approaches (CPLEX). The number of unassigned duties in the sequential approach is equal to the number in RS plus those in DS.

instance	unassigned duties (%)		# unassigned days		# standbys underflow		# moved days off		# moved days off weekends		# underrun days off		# single-off overflow		# double-off dis. vio.		maximal overtime (in hr.)		running time (in sec.)	
	seq.	var2	seq.	var2	seq.	var2	seq.	var2	seq.	var2	seq.	var2	seq.	var2	seq.	var2	seq.	var2	seq.	var2
48-75-6	1.3	0.3	14	0	0	1	0	0	0	0	48	48	0	0	0	0	80.7	80.2	11	27
52-73-6	0.5	0	7	0	1	2	0	0	0	0	52	52	1	1	0	0	57.9	60.3	10	50
52-75-6	1.3	0	16	0	1	2	0	0	0	0	52	52	0	0	0	0	60.8	53.8	8	59
9-238-11	6.3	1.5	57	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	5,483
393-45-37	8.6	4.4	141	0	0	0	0	0	0	0	0	0	0	0	0	0	81.3	81.3	81	3,500
392-45-37	16.9	11.1	165	0	0	0	0	0	0	0	0	0	0	0	0	0	82.9	82.9	78	24,600
397-40-37	9.0	3.8	132	0	0	0	0	0	0	0	0	0	0	0	0	0	73.9	73.9	61	24,600
96-70-8	11.7	6.3	174	0	1	0	1	1	8	8	0	0	0	0	0	0	0	0	27,313	48,100
87-70-8	4.0	0.2	127	0	12	12	0	0	3	3	0	0	0	0	0	0	1.1	0	51,907	86,400

Table 8: Detailed breakdown of the characteristics of the solutions of the integrated (var2) vs. the sequential (seq) approaches (CPLEX)

References

1. L. Bianco, M. Bielli, A. Mingozzi, S. Ricciardelli, and M. Spadoni, A heuristic procedure for the crew rostering problem, European Journal of Operational Research, volume 58, pages 272-283 (1992)
2. E. K. Burke and J. D. Landa Silva, The design of memetic algorithms for scheduling and timetabling problems, In N. Krasnogor, W.E. Hart, and J.E. Smith, editors, Recent Advances in Memetic Algorithms, Studies in Fuzziness and Soft Computing, volume 166, pages 289-312. Springer (2005).
3. P. Carraresi and G. Gallo, A multilevel bottleneck assignment approach to the bus driver's rostering problem, European Journal of Operational Research, volume 16(2), pages 163-173 (1984)
4. P. Cappanera and G. Gallo, A multicommodity flow approach to the crew rostering problem. Operations Research, volume 52(4), pages 583-596 (2004)
5. F. Catanas and J. Paixão, A new approach for the crew rostering problem, In J. Daduna, I. Branco, and J. Paixão, editors, Computer-Aided Transit Scheduling, volume 430 of Lecture Notes in Economics and Mathematical Systems, pages 267-277. Springer (1995)
6. M. Dorigo and C. Blum, Ant colony optimization theory: A survey, Theoretical Computer Science, volume 344(2), pages 243-278 (2005)
7. G. De Pont, Personalized crew rostering at Netherlands railways, Master's thesis, University of Tilburg (2006)
8. P.R. Day, D.M. Ryan, Flight attendant rostering for short-haul airline operations. Operations Research, volume 45(5), pages 649-661 (1997)
9. T. Emden-Weinert, H.G. Kotas, and U Speer, Dissy-a driver scheduling system for public transport, Technical report, VSS GmbH and Bremer Strassenbahn AG, Bremen, Germany (2000)
10. J. Ferreira and R. Guimaraes, A travelling salesman model for the sequencing of duties in bus crew rotas, Journal of the Operational Research Society, volume 46(4), pages 415-426 (1995)
11. R. Freling, Models and Techniques for Integrating Vehicle and Crew Scheduling, PhD thesis, Erasmus University of Rotterdam (1997)
12. Gurobi Optimization, Gurobi Optimizer 5.5. Gurobi: http://www.gurobi.com/ (2013)
13. IBM Corporation. IBM ILOG CPLEX Optimizer 12.5. IBM: http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/ (2013)
14. J. Jachnik, Attendance and rostering systems, In A. Wren, editor, Computer Scheduling of Public Transport, pages 337-344. Elsevier (1981)
15. N. Kohl and S. Karisch, Airline crew rostering: Problem types, modeling, and optimization, Annals of Operations Research, volume 127, pages 223-257 (2004)
16. J. Kyngäs and K. Nurmi, Days-off scheduling for a bus transportation company, Int. J. Innovative Computing and Applications, volume 2(1), pages 42-59 (2011)
17. M. Lezaun, G. Pérez, and E. S. de la Maza, Crew rostering problem in a public tranport company, Journal of the Operational Research Society, volume 57, pages 1173-1179 (2005)
18. M. Lezaun, G. Pérez, and E. S. de la Maza, Rostering in a rail passenger carrier, Journal of Scheduling, volume 10(4-5), pages 245-254 (2007)
19. P. Lucic and T. Teodorovic, Metaheuristics approach to the aircrew rostering problem, Annals of Operations Research, volume 155, pages 311-338 (2007)
20. M. Mesquita, M. Moz, A. Paias, J. Paixão, M. Pato, and A. Respício. A new model for the integrated vehicle-crew-rostering problem and a computational study on rosters. Journal of Scheduling, volume 14(3), pages 319-334 (2011)
21. M. Moz and M.V. Pato. A genetic algorithm approach to a nurse rerostering problem. Computers and Operations Research, volume 34(3), pages 667-691 (2007)
22. T. Hanne, R. Dornberger, and L. Frey, Multiobjective and preference-based decision support for rail crew rostering. In: Proc 11th conference on Congress on Evolutionary Computation. IEEE, pages 990-996 (2009)
23. A. Hartog, D. Huisman, E.J.W. Abbink, and L.G. Kroon, Decision support for crew rostering at NS, Public Transport, volume 1(2), pages 121-133 (2009)
24. M. Moz, A. Respicio, and M. V. Pato, Bi-objective Evolutionary Heuristics for Bus Drivers Rostering, Centro de Investigacao Operacional, University of Lisbon (2007)

25. K. Nurmi and J. Kyngäs, Driver Rostering for a Finnish Bus Transportation Company, Satakunta University of Applied Sciences, Finland (2011)
26. G.L. Nemhauser and L.A. Wolsey, Integer and combinatorial optimization, volume 18, Wiley New York (1998)
27. D. Pedrosa and M. Constantino. Days-off scheduling in public transport companies. In S. Voss and J. Daduna, editors, Computer-Aided Scheduling of Public Transport, volume 505 of Lecture Notes in Economics and Mathematical Systems, pages 215-232. Springer Berlin Heidelberg (2001)
28. J. Prakash, S.B. Sinha, and S.S. Sahay, Bus transportation crews planning by goal programming, Socio-Economic Planning Sciences, volume 18(3), pages 207-210 (1984)
29. A. Respicio, M. Moz, and M. V. Pato, A Memetic Algorithm for a Bi-objective Bus Driver Rostering Problem, Centro de Investigacao Operacional, University of Lisbon (2007)
30. M.S. Sodhi and S. Norris, A fexible, fast, and optimal modeling approach applied to crew rostering at London underground, Annals of Operations Research, volume 127, pages 259-281 (2004)
31. W. Townsend, An application of the assignment model to bus crew rostering, Journal of Mathematics in Management, volume 1, pages 45-52 (1986)
32. W. Townsend, An approach to bus-crew roster design in London regional transport, Journal of the Operations Research Society, volume 6, pages 543-550 (1988)
33. L. Xie, N. Kliewer, and L. Suhl, Integrated driver rostering problem in public bus transit, In Procedia-Social and Behavioral Sciences, volume 54, pages 656-665 (2012)
34. L. Xie, M. Naumann, and L. Suhl, A stochastic model for rota scheduling in public bus transport, In Proceedings of 2nd Stochastic Modeling Techniques and Data Analysis International Conference, pages 785-792 (2012)
35. Zuse Institute Berlin, MIPLIB 2010, ZIB: http://miplib.zib.de/miplib2010.php (2013) last retrieved 07.01.2013.