

A duty-block network approach for an integrated driver rostering problem in public bus transport

L. Xie & L. Suhl

DS&OR Lab, University of Paderborn, Germany

Abstract

The driver rostering problem (DRP), arising in public bus transport companies, aims to generate for each driver a roster while management considerations, labor laws, and desires of drivers have to be satisfied. Optimal rosters are characterized by maximal satisfaction of drivers, minimal difference of overtime among all drivers, and minimal number of unassigned duties. The DRP is mostly solved sequentially due to its high complexity, namely firstly the rota scheduling problem, and secondly the duty sequencing problem. However, this method may generate sub-optimal rosters. In order to avoid a sub-optimal solution, the paper discusses an integrated DRP, which is solved with a duty-block network approach for real-world instances and compared with a solution of a heuristic-based integrated DRP.

Keywords: integrated driver rostering problem, network optimization model, public transport, rotation-based rostering.

1 Introduction

The operational planning process of public bus transport companies is divided into four steps, which are carried out sequentially. Figure 1 shows the relationship between the four operational planning problems at a public bus transport company (from Freling [1]). *Line routes* and their *frequencies* are given as input for a *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. In *vehicle scheduling* vehicles are assigned to timetabled trips resulting in *vehicle blocks*. After



defining a sequence of *tasks* on each vehicle block, each task must be assigned to one *duty* for a working period (one day). A duty should satisfy several work regulations, such as the regulation of length and frequency of breaks within a duty. This process is defined as *driver scheduling*. *Rosters* (monthly work schedules) are constructed from daily duties considering some additional work regulations in the *driver rostering problem* (DRP). Each duty has its beginning time, ending time, the weekday it belongs to, the shift type (e.g. early shift) etc. Each driver can execute one duty per day. DRP addresses the task of assigning the duties to drivers while management considerations, labor laws, and the preferences of drivers have to be satisfied. After solving the DRP, a roster is generated for each driver. The optimal rosters are characterized by maximal satisfaction of drivers, minimal difference of overtime among all drivers, and minimal number of unassigned duties.

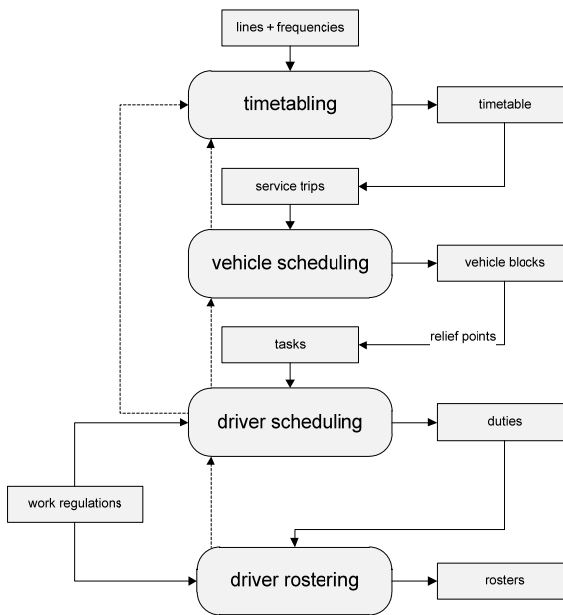


Figure 1: The planning process at a public transportation company (from Freling [1]).

There are different ways to generate rosters, namely cyclic roster, personalized roster and bidline roster. For each group of driver a *cyclic roster* is generated, which reflects the fairness in the following context that drivers within that group have the same duties including unpopular duties. A *personalized roster* is generated for a driver while special desires of single free day or vacations can be well considered. The personalized rostering approach is mostly used in the airline industry, especially in European airlines because of its fairness (Cappanera and Gallo [2], Kohl and Karisch [3] and Maenhout and Vanhoucke [4]). In recent years, generating a personalized roster in railways is

getting more attention (see De Pont [5]). In *bidline rostering*, which is used for example at North American airlines, a roster is generated subsequently for each individual crew member (see Kohl and Karisch [3] for an overview). This paper exclusively presents a cyclic roster, which is presented in Section 2. In the remainder of the paper, we refer a cyclic roster and roster interchangeably.

Traditionally, in public transport the DRP is solved sequentially due to its high complexity: *rota scheduling* and *duty sequencing* consecutively (Sodhi and Norris [6], Hartog *et al.* [7]). The rota scheduling aims at assigning a set of shifts, free days, and reserve shifts to driver groups (rotas are generated) while the duty sequencing problem concentrates on assigning the duties to driver groups according to the optimal rotas. In the sequential DRP, sub-optimal rosters can be generated even though rotas are optimal. In order to avoid the sub-optimality, this paper discusses an integrated DRP, which is solved for real-world instances. To the best of our knowledge, this paper presents the first approach to solve the two sequential problems simultaneously in public transit.

We conclude this section with an outline of remainder of this paper. Section 2 describes the rotation-based crew rostering problem in detail with an example, which will be used throughout the whole paper. In Section 3 we propose a new network model for the integrated DRP. Additionally, in order to compare the results of the integrated solution approach, a heuristic-embedded integrated approach is described. The mathematical formulation of the integrated solution approach is also described in this section. This formulation is solved with a standard commercial solver. Finally, in Section 4 we present computational results for comparing the integrated approach with the heuristic-embedded integrated approach.

2 Problem description

In rotation-based DRP, the desired rota for each driver group and a set of duties are considered as input data for the DRP. As mentioned before, each duty belongs to exactly one shift type depending on its starting time and ending time. *Shift types* can be roughly classified as early, midday and late shifts. However, some German bus companies define shift types more precisely, like very early shift (FFD), early shift (SFD), earlier midday shift (FMD), midday shift (SMD), earlier night shift (FSD) and night shift (SSD) etc. A sequence of consecutive shift types is called *Shift pattern*. Under a *backward shift pattern* the drivers would like to have a late shift at the beginning of a working week, then midday shifts, and an early shift at the end of the working week, in order to get longer free days before and after the working week, and vice versa for a *forward shift pattern*. A *steady shift pattern* contains the same or similar shifts within a working week.

A *desired rota* is a sequence of desired shift patterns, free days, reserve duties as well as some pre-assigned activities for a group of drivers. Table 1 shows an example of a desired rota for five weeks. The drivers within a *group* have the same qualification, same desires of the shift pattern and same route knowledge.

Table 1: An example of a desired rota.

| Week | Mon | Tue | Wed | Thu | Fri | Sat | Sun |
|------|-----|-----|-----|-----|-----|-----|-----|
| 1 | R | R | R | R | R | F | F |
| 2 | SSD | FSD | SMD | FMD | SFD | F | F |
| 3 | SSD | FSD | SMD | FMD | SFD | F | F |
| 4 | SSD | FSD | SMD | FMD | SFD | F | F |
| 5 | SSD | FSD | SMD | FMD | SFD | F | F |

The columns of a rota represent the consecutive days of a week, from Monday to Sunday, while the rows represent consecutive weeks, and the number of weeks is equal to the number of drivers in a roster. The first driver begins with the first row in the first week while the second driver begins with the second row, etc. The duties in the first row follow the duties in the last row. After they accomplished all duties of the rota, they begin the duties of their first week again.

One of the objectives for solving rotation-based DRP is to maximize the number of desired shifts. Therefore, undesired shifts are gradually punished. For example, another shift type as a desired early shift (SFD) is punished, but a very early shift (FFD) is less punished than earlier midday shift (FMD) (see Table 2 rows 1 and 2). Therefore, all admissible alternatives of a shift as well as admissible successors (Table 3) with the corresponding cost are defined. Note that a desired successor has the cost 0. The costs shown in Tables 2 and 3 are artificial costs used to satisfy the desires of drivers as far as possible.

After solving the rotation-based DRP, a roster is generated for each group of drivers. A *roster* is defined as a sequence of duties, free days as well as other

Table 2: An example of possible admissible shifts with the corresponding costs.

| | Shift 1 | Shift 2 | Cost |
|---|---------|---------|------|
| 1 | SFD | FFD | 50 |
| 2 | SFD | FMD | 100 |
| 3 | FMD | SMD | 50 |
| 4 | SMD | FMD | 50 |
| 5 | FSD | SSD | 50 |
| 6 | SSD | FSD | 50 |

Table 3: An example of admissible successor shifts with the corresponding costs.

| | Shift 1 | Shift 2 | Cost |
|---|---------|---------|------|
| 1 | SSD | FSD | 0 |
| 2 | SSD | SSD | 100 |
| 3 | FSD | SMD | 0 |
| 4 | FSD | FMD | 100 |
| 5 | SMD | FMD | 0 |
| 6 | SMD | SMD | 100 |
| 7 | FMD | SFD | 0 |
| 8 | FMD | FFD | 50 |

activities, such as reserve duties, and training. During the generation of a feasible roster, some hard regulations must be held, such as rest periods between two duties, maximal consecutive working days, maximum number of single day-off etc.

3 Integrated approach

In this section we discuss our solution approach for the integrated DRP. First of all, the day-offs and reserve duties are pre-assigned based on a given desired rota (see Section 3.1). In Sections 3.2 and 3.3 a desired shift-block network and a reduced desired shift-block network are described for the rest of the desired rota. For each desired/reduced desired shift-block network a duty-block is defined. We describe the duty-block network and mathematical formulation in Sections 3.4 and 3.5.

3.1 Pre-assignment of free days and reserve duties

The desired distributions of reserve duties and free days are checked for feasibility first. The goal is to retain the distribution to the largest possible extent, unless the following hard constraints cannot be satisfied.

- The minimum/maximum number of consecutive working days
- The maximum number of consecutive free days
- The minimum and maximum number of single/two consecutive free days
- The distance between two consecutive free days should be less than 14 days.

A similar mathematical formulation for the restrictions can be found in Emden-Weinert *et al.* [8] and Speer [9].

The distributions of reserve duties are currently evenly planned for all drivers without considering more detailed information such as historical or weekday-dependent sickness absence rates. A stochastic model for better planning of reserve duties is presented in Xie *et al.* [10].

3.2 Desired shift-block network

After pre-assignment of free days and reserve duties (F and R in Table 2), possible duties are required to assign to the rest of desired rota. We define the term *desired shift-block* as a sequence of consecutive desired shifts (e.g., in Table 1: SSD→FSD→SMD→FMD→SFD from Monday to Friday) in a desired rota. Note that the length of a block should be less than or equal to the maximum number of consecutive working days. Similar desired shift-blocks are gathered to determinate the frequency, and we generate a desired shift-block network for all of them. (e.g., in Table 1: the frequency of SSD→FSD→SMD→FMD→SFD from Monday to Friday is equal to four). Two blocks are *similar*, if they have the same shift pattern, and each of the shift types in the shift pattern is carried out on the same day. Based on the given possible alternative and successor shifts described in the example in Section 2, desired shift-block network (DSB-network) is generated as follows (see fig. 2). Each *node* in the network represents

either the desired shift type or an alternative shift type on a weekday. An *arc* between two nodes indicates that the combination of two nodes is possible (based on Table 3). A shift type of the first day without a successor or a shift type of the last day without a predecessor will be eliminated (e.g. FSD on Monday). A shift type of the rest of the days will also be eliminated, if it does not have any successor or predecessor (e.g. SSD on Tuesday).

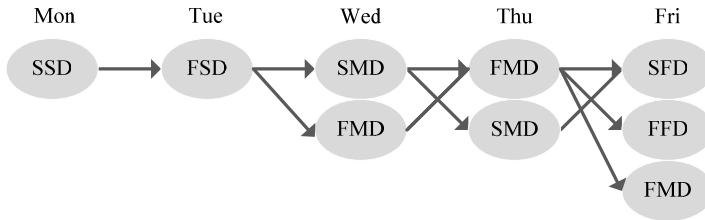


Figure 2: The desired shift-block network.

3.3 Reduced desired shift-block network

One heuristic is built for reducing the shifts-combinations in the DSB-Network shown in fig. 2.

Algorithm 1 (Heuristic for DSB-Network)

Step 0: Initialization

Let ST_t be a set of shift types on a day t . The differences $diff_{s,t}$ between the number of available shifts and the number of desired shifts on day t are calculated for each possible shifts $s \in ST_t$ on day t . If $diff_{s,t}$ is larger or equal to zero, then we define the shift s on day t is under-desired. Otherwise, the shift s on day t is defined as over-desired.

Step 1: Finding alternative shifts

Case 1.1: If the desired shift on day t of the DSB-Network is under-desired, then no alternative shifts are generated. Otherwise, all alternative shifts of that desired shift are generated.

Case 1.2: For all desired shift on day t of the DSB-Network the alternative shifts are generated, which are under-desired and have at least a possible predecessor shift on day $t-1$ (if t is not the first day in the DSB-Network).

Case 1.3: If the desired shift on day t of the DSB-Network is under-desired, then no alternative shifts are generated. Otherwise, the alternative shifts are generated, which are under-desired and have at least a possible predecessor shift on day $t-1$ (if t is not the first day in the DSB-Network).

Step 2: Generating a reduced desired shift-block network

The network structure is similar to how a desired shift-block network generates in Section 3.2.

We assume that all desired shifts but FMD on Thursday in this example are over-desired. FMD on Wednesday and FFD on Friday are over-desired. The rests are under-desired. Figure 3 presents the reduced desired shift-block network if

we choose the case 1.1 in step 1 of our heuristic. Because FMD on Thursday is under-desired, we do not generate any alternative shifts for it. The reduced desired shift-block network shown for case 1.2 presents in fig. 4 that FMD on Wednesday and FFD on Friday are not generated. Case 1.3 is considered as the combination between case 1.1 and 1.2, and it is shown in fig. 5.

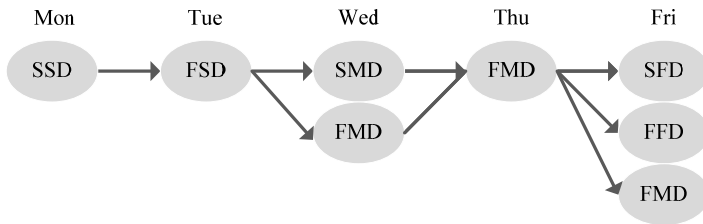


Figure 3: The reduced desired shift-block network for case 1.1.

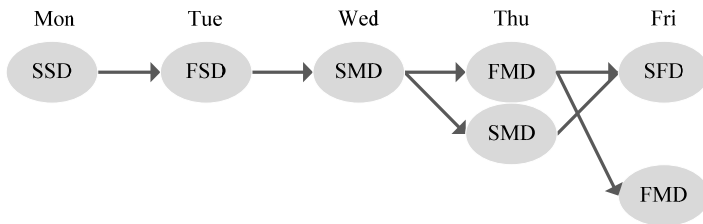


Figure 4: The reduced desired shift-block network for case 1.2.

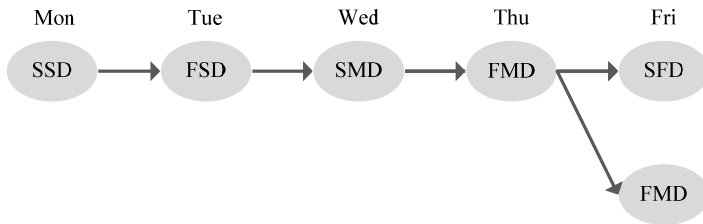


Figure 5: The reduced desired shift-block network for case 1.3.

3.4 Duty-block network

A *duty-block* includes all sequences of possible consecutive duties in a desired shift-block network, which one is described in the previous section. We denote the set of possible shift types of a day t by S_t , which can be generated from the desired shift-block network. The possible duties of a shift type s on a day t would be collected to a set $D_{s,t}$. Let the frequency of the desired shift-block be f .

For each duty-block a network is generated in the following way. A *node* in the network is a starting node s or an ending node t of the network or a beginning or ending node of a duty/dummy duty. The starting node s is connected to all

duties $d \in D_{s,t}$ with $s \in S_I$ and the dummy duty of the first day. The ending node t is connected to all duties $d \in D_{s,t}$ and $s \in S_S$ and the dummy duty of the last day. A *duty arc* represents a duty $d \in D_{s,t}$ with $s \in S_I$ and has at least one incoming and one outgoing arc, i.e. the upper bound is 1. A *dummy arc* is added for each day once to guarantee a feasible solution. The upper bound of this dummy arc is set to f . A *circulation arc* from t to s is added to guarantee that exactly f paths are generated and its lower and upper bound are set to f .

The connections between those arcs are needed to complete the network. Thus, a *connecting duty arc* is added where an arc is connected with a duty arc or between duty arcs, so its upper bound is equal to the upper bound of a duty arc, i.e. one. Two duty arcs $d1$ and $d2$ can only be connected if they are compatible. *Compatible* means that $d2$ is a possible successor of $d1$ (see Section 2) and the rest period between two duties is larger as the defined overtime rest period (e.g. 11 hours). A *connecting dummy arc* is only added between circulation arc and dummy arc or between dummy arcs, and its upper bound is equal to f .

Figure 6 depicts a duty-block network for the example which is shown in previous sections. The duty-block network for a reduced desired shift-block network (case 1.3) is shown in fig. 7. If a duty arc is not connected with any duty arcs, then it cannot be generated in fig. 7.

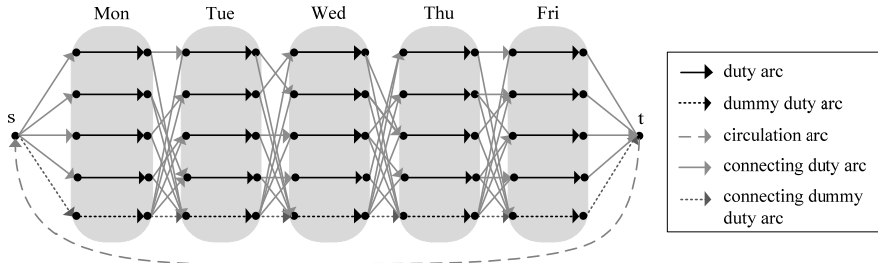


Figure 6: The duty-block network for the desired shift-block network in fig. 2.

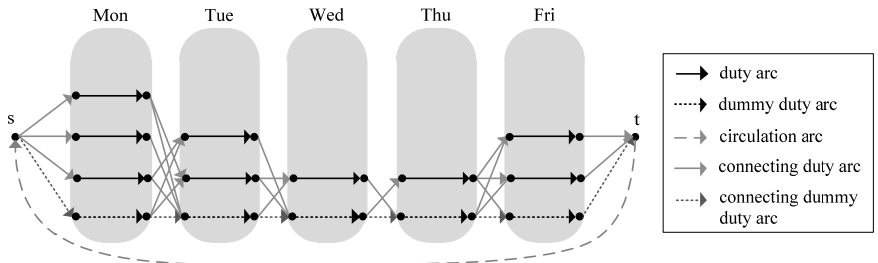


Figure 7: The duty-block network for the reduced desired shift-block network in fig. 5.

3.5 Mathematical formulation

In this section we present our mathematical formulation, which relies on the network structure described in Section 3.4. Let $D = \{1, 2, \dots, n\}$ be the set of n duties and $B = \{1, 2, \dots, m\}$ be the set of m duty-blocks. For each duty-block $b \in B$, $G^b = (N^b, E^b)$ defines a cyclic network as described earlier, with N^b as the set of nodes and E^b as the set of edges. We denote $DE^b \subset E^b$ as the set of duty arcs. The set of connecting duty arcs is denoted as $CDE^b \subset E^b$. Let DE^d be the set of arcs representing duty d in all duty-blocks. We introduce the dummy duties to ensure a feasible solution. Let $DUE^b \subset E^b$ be the set of dummy arcs and the set of connecting dummy arcs is denoted as $CDUE^b \subset E^b$. Binary decision variables x_e with $e \in DE^b \cup CDE^b$ indicate whether the edge e is selected or not. Besides that, we use linear decision variables x_e with $e \in DUE^b \cup CDUE^b$ indicating how many times the edge e can be selected. All linear decision variables have a maximum capacity f^b as described in the previous section. With each arc $e \in E^b$ we associate a cost c_e that is the total cost of adding the arc e . The cost includes the alternative or successor cost of adding that arc. The integrated crew rostering problem with multiple duty-blocks can be stated as follows:

$$\min \sum_{b \in B} \sum_{e \in E^b} c_e * x_e \quad (1)$$

$$\sum_{i|(i,j) \in E^b} x_{ij} - \sum_{i|(j,i) \in E^b} x_{ji} = 0 \quad \forall i \in N^b, b \in B \quad (2)$$

$$\sum_{e \in DE^d} x_e \leq 1 \quad \forall d \in D \quad (3)$$

$$x_e \in \{0,1\} \quad \forall e \in DE^b \cup CDE^b, b \in B \quad (4)$$

$$0 \leq x_e \leq f^b \quad \forall e \in DUE^b \cup CDUE^b, b \in B \quad (5)$$

$$x_{e=(i,s)^b} = f^b \quad \forall b \in B \quad (6)$$

The objective (1) minimizes the sum of arc costs, i.e. it maximizes the satisfaction of drivers. Constraints (2)-(3) correspond to a multicommodity flow formulation for the duty-network crew rostering problem where flow conservation is ensured for each duty-block in eqn. (2) and each duty can only be assigned for all duty-blocks at most once in eqn. (3). Due to restriction imposed by eqn. (3) it may have some unassigned duties in the solution, the number of which should be minimized. The costs of selected dummy arcs and connecting dummy duty arc are set to a large number. Therefore, it induces to minimize the number of dummy arcs. In other words, it minimizes the number of unassigned duties. Constraint (4) defines that each duty arc and connecting duty arc correspond to a binary variable, while constraint (5) defines the capacity of the dummy duty and connecting dummy duty arcs. Constraint (6) ensures that there are exactly f^b paths in the solution.

This mathematical formulation minimizes the dissatisfaction of drivers. The generated rosters can be improved with a simulated annealing approach for



minimizing the difference of overtime between all drivers. Xie *et al.* [11] shows the simulated annealing approach for the integrated model shown in this paper and the computational results indicate that it is worth to use simulated annealing to improve the quality of solutions.

4 Computational results

In this section we report computational results of our modeling and solution approach. All experiments were conducted on a Dell Latitude E5510 personal computer running Windows 7 64 bits with an Intel Core i5 processor @2.53GHz and 4 GB of main memory. Our integrated model and heuristic-based integrated model shown in Section 3 are solved by the MLP solver ILOG CPLEX 12.0.

We tested our approaches on two real-world medium-sized instances from public transport companies, namely 256_1013_9 and 256_899_9. The instance name contains information about the problem data: The number of drivers, the number of duties, and the number of different driver groups. The both instances we chose have the same number of drivers but different number of duties, which are needed to be assigned. Additionally, there are less alternative shifts for the desired shifts in the instance 256_899_9 compared to the instance 256_1013_9.

In Table 4 we report results of the integrated approach, which are compared with the heuristic-based integrated approach based on the following criteria.

- The running time;
- The number of unassigned duties;
- The number of duties, which are assigned exactly as desired (DAaD);
- The number of duties, which are assigned but not as desired (DAaA).

Table 4: The comparison of integrated model with and without heuristic.

| criteria | 256_1013_9 | | | | 256_899_9 | | | |
|--|----------------|-------------|-------------|---------|----------------|-------------|-------------|-------------|
| | with heuristic | | | without | with heuristic | | | witho ut |
| | Case 1.1 | Case 1.2 | Case 1.3 | | Case 1.1 | Case 1.2 | Case 1.3 | |
| running time (in sec.) | 266.2 | 119.4 | 49.8 | 516.6 | 5.3 | 2.0 | 2.0 | 41 |
| # unassigned duties | 32 | 51 | 65 | 23 | 138 | 316 | 277 | 103 |
| # duties assigned as desired (DAaD) | 643 | 672 | 677 | 726 | 358 | 140 | 212 | 626 |
| # duties assigned as alternatives(DAaA) | 338 | 291 | 271 | 264 | 406 | 443 | 410 | 170 |

Note that the running time of the integrated approach without considering the heuristic is much longer than the running time of the heuristic-based integrated approach. However, the running time of the integrated approach without heuristic still can be solved within about 8 minutes for each instance. As mentioned before in this section, there are less alternative shifts in the instance 256_899_9, therefore, it can be solved faster compared to the instance 256_1013_9. That is also the reason why the number of unassigned duties is higher in this instance, although the number of duties, which are required to be

assigned, is smaller. In general, the integrated approach outperforms the heuristic-embedded integrated approach with regard to solution quality, i.e. more assigned duties, and more duties assigned as desired (see Table 4).

Additionally, we test our instance with three cases in step 1 shown in Algorithm 1. We notice that if we use case 1.1 in our heuristic in both instances, the number of unassigned duties is clearly less than the results of using the case 1.2 and 1.3. The reason is that all alternative shifts of all over-desired shifts are generated. However, due to the same reason, the number of DAaD is less than the other cases of step 1 in the instance 256_1013_9. In the instance 256_899_9 the result of the case 1.1 is the best one compared to other cases in the Algorithm 1 according to all criteria except the running time, because the most desired shifts in this instance are over-desired. Moreover, the under-desired alternative shifts are not generated for an over-desired shift in case 1.3. That is the reason why it brings the most DAaD in the instance 256_1013_9 than all other cases in Algorithm 1.

The comparison of the solutions of our integrated and sequential approaches as well as the description of the sequential approach can be found in Xie *et al.* [11]. It is shown that our integrated solution approach outperformed the sequential approach in terms of solution quality.

5 Summary

This paper presents a new modeling approach for the integrated crew rostering problem in public transport. The formulation relies on a new duty-block network. The results demonstrated that the medium-sized real-world instances could be solved very efficiently even without the heuristic which we have developed.

References

- [1] Freling, R., Models and Techniques for Integrating Vehicle and Crew Scheduling, PhD thesis, Erasmus University of Rotterdam, 1997.
- [2] Cappanera, P. and Gallo, G., A multicommodity flow approach to the crew rostering problem, *Operations Research*, **52**(4), pp. 583–596, 2004.
- [3] Kohl, N. and Karisch, S., Airline Crew Rostering: Problem Types, Modeling, and Optimization, *Annals of Operations Research*, **127**, pp. 223–257, 2004.
- [4] Maenhout, B. and Vanhoucke, M., A hybrid scatter search heuristic for personalized crew rostering in the airline industry, *European Journal of Operational Research*, **206**, pp. 223–257, 2010.
- [5] De Pont, G., Personalized crew rostering at Netherlands Railways, Master's thesis, University of Tilburg, Netherland, 2006.
- [6] Sodhi, M.S. and Norris, S., A flexible, fast, and optimal modeling approach applied to crew rostering at London underground, *Annals of Operations Research*, **127**, pp. 259–281, 2004.
- [7] Hartog, A., Huisman, D., Abbink, E.J.W. and Kroon, L.G., Decision support for crew rostering at NS, *Public Transport*, **1**, pp. 121–133, 2009.



- [8] Emden-Weinert, T., Kotas, H.G., Speer, U., DISSY-A driver scheduling system for public transport, <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.33.7675>
- [9] Speer, U., Turnusoptimierung im öffentlichen Personennahverkehr, Master's thesis, Technische Universität Dortmund, Germany, 2000.
- [10] Xie, L., Naumann, M. and Suhl, L., A stochastic model for rota scheduling in public bus transport, working paper, University of Paderborn, Germany, 2012.
- [11] Xie, L., Kliewer, N. and Suhl, L., Integrated driver rostering problem in public bus transit, working paper, University of Paderborn, Germany, 2012.

