

Decision support for crew rostering at NS

Anneke Hartog · Dennis Huisman ·
Erwin J.W. Abbink · Leo G. Kroon

Published online: 20 May 2009

© The Author(s) 2009. This article is published with open access at Springerlink.com

Abstract This paper describes a method for solving the cyclic crew rostering problem (CCRP). This is the problem of cyclically ordering a set of duties for a number of crew members, such that several complex constraints are satisfied and such that the quality of the obtained roster is as high as possible. The described method was tested on a number of instances of NS, the largest operator of passenger trains in the Netherlands. These instances involve the generation of rosters for groups of train drivers or conductors of NS. The tests show that high quality solutions for practical instances of the CCRP can be generated in an acceptable amount of computing time. Finally, we describe an experiment where we constructed rosters in an automatic way for a group of conductors. They preferred our—generated—rosters over their own manually constructed rosters.

1 Introduction

The assignment of work to individual crew members is a complex task for each public transport company. Traditionally, this process is split into two steps. In the first step,

D. Huisman (✉) · E.J.W. Abbink · L.G. Kroon
Department of Logistics, Netherlands Railways, P.O. Box 2025, 3500 HA Utrecht, The Netherlands
e-mail: huisman@few.eur.nl

D. Huisman
e-mail: huisman@ese.eur.nl

E.J.W. Abbink
e-mail: erwin.abbink@ns.nl

L.G. Kroon
e-mail: lkroon@rsm.nl

D. Huisman · L.G. Kroon
Erasmus Center for Optimization in Public Transport (ECOPT), Erasmus University Rotterdam,
P.O. Box 1738, 3000 DR Rotterdam, The Netherlands

duties are constructed where a duty is the work for one crew member on a single day. These duties have to fulfill a lot of requirements. For instance, there is an upper bound on the length of each duty and there should be a break in each duty. This process is called *crew scheduling*. In the next step, rosters are created where sequences of duties are constructed. These sequences are assigned to the individual crew members. This problem is called *crew rostering*.

Rostering can be done in several ways: (i) a roster for individual crew members can be created where crew specific characteristics (e.g. their vacations) can also be taken into account, (ii) a bid line can be constructed where individual crew members can bid for, or (iii) a cyclic roster can be constructed. The first two rostering approaches are mainly used in the airline industry (see Kohl and Karisch 2004 for an overview). However, many European public transport companies, including NS, use the concept of cyclic rosters. In the remainder of the paper, we refer to a cyclic roster if we use the word roster.

The remainder of this paper is organized as follows. The concept of cyclic rosters is explained in more detail in Sect. 2. In Sect. 3, we discuss the current crew planning process at NS. The rostering problem is solved in two steps: (i) the allocation of the duties among the different groups of crew members, and (ii) the actual construction of the rosters for each group. The focus in this paper is on the second step. However, in Sect. 4, we briefly discuss a straightforward solution approach for the allocation of the duties to the groups. In Sect. 5, we provide a mathematical formulation for the cyclic crew rostering problem. This formulation is solved with a standard commercial solver. Computational results that show the suitability of this approach are provided in this section as well. To test the approach in practice, we did an experiment for conductors in the crew base Utrecht of NS. The results of this experiment are described in Sect. 6. Finally, we end this paper with some concluding remarks.

2 The cyclic crew rostering problem

In the cyclic crew rostering problem (CCRP), rosters are created for a group of crew members, where crew members are in the same group if they have the same characteristics (e.g. drivers, full-time employees, same route knowledge). For such a group, one roster is constructed with a length in weeks equal to the number of crew members in the group. If a roster is of size k , then k indicates the number of weeks and the number of crew members in the roster.

The input for the CCRP consists of a set of duties for each day of the week. Since the roster is cyclic, all weeks have the same duties. Furthermore, a roster has to satisfy many labor rules related to days off, working time, etc.

Schematically, a roster can then be seen as a set of rows and columns, where the columns correspond to the different days of the week, and the rows correspond to the different weeks. Table 4 (Sect. 5.6) gives an example of a roster for 20 weeks which is to be carried out by 20 crew members. Crew member 1 starts in week 1 with the duties in the first row of the roster, while crew member 2 starts in the first week

with the duties in the second row. These duties are carried out by crew member 1 in the second week. The remaining part of the roster can be explained similarly. Notice hereby that the duties in the first row follow after the duties in the last row.

There are many rules indicating whether a roster is feasible or not. Those rules deal with the rest periods between two duties, the rest periods in a week, the number of weekend days off, the number of night duties, etc.

Only a few papers have studied the CCRP. For example, (Caprara et al. 1998) developed a heuristic based on a mixed integer programming formulation to determine a roster with a minimum number of weeks such that each duty is done once every day. A recent paper (Sodhi and Norris 2004) deals with the CCRP at the London Underground, which is a complex problem considering all kinds of hard practical constraints. The authors decompose the problem into two stages, where in the first stage a pattern of rest-days and duty types is created for each depot, and afterwards the individual duties are assigned to this pattern. The first phase is the most complex part, which is further decomposed into three steps. The most complicated step is to find the “optimal” rest-day pattern for each depot. This problem is solved as a mixed integer program. The second phase can be formulated straightforwardly and solved as an assignment problem with side constraints.

3 Crew planning at NS

In this section, we describe the crew planning process for drivers and conductors at NS. Other crew members (at ticketing offices, the call center, mechanics, etc.) fall outside the scope of this paper. The crew scheduling problem (CSP) is the problem of assigning tasks to anonymous duties. These tasks are given by the timetable and by the rolling stock schedule (see Huisman et al. 2005 for a discussion on all planning problems at NS). A duty is the work for one crew member from a specific crew base on a certain day. After the construction of the duties, the rosters are constructed. As mentioned before, the concept of cyclic rostering is used. Since each duty belongs to a certain crew base, the CCRP should be solved for each crew base separately. To increase the influence of the crew members on the final rosters, the rosters are constructed locally at the different crew bases themselves. An important aspect for the crew members is a fair division of the popular and the unpopular parts of the work.

Before we focus on the rostering process, we first discuss the crew scheduling process in more detail. At NS, the crew scheduling process has been split in two stages. First, the crew schedules for the annual plan are constructed. This plan deals with a generic Monday, Tuesday and so on. This generic annual plan is modified about 6 times a year. In the CSP that is solved for generating the generic annual plan, some rostering aspects are also taken into account. For instance, the average duty length over all duties on a certain crew base should not exceed 8 hours. The reason is that, if this time is exceeded, then it is impossible to construct rosters where the average working time per week is less than 36 hours (in principle each full-time crew

member works 9 days in two weeks). Moreover, it is important that, to obtain a fair division of the work for the different crew members, the work should be fairly spread over the different bases. The latter constraints are typical for the Dutch situation and are known as “Sharing Sweet & Sour” rules. They aim at allocating the popular and the unpopular work as fairly as possible among the different crew bases. For instance, some routes are more popular than others and Intercity trains are preferred over regional trains. For a detailed description of these rules, we refer to (Abbink et al. 2005).

The crew schedules for the specific days, e.g. Friday December 2, 2005, are based on the schedules for the corresponding generic day (in the example the schedules for Friday). However, for the specific days, the above mentioned complex rules do not need to be taken into account (see Huisman 2007). On the other hand, the rosters are not changed anymore for the specific days. Therefore, modifications in the generic crew schedules for adapting them to the specific days should fit in the rosters. This limits the possibilities in the CSP for the specific days. Only if the CSP cannot be solved otherwise, the rosters may be modified. As a consequence, the crew members know their required attendance at work already long time in advance.

The most important rostering problem is to be solved each year at the beginning of a new timetable year: then the basic rosters are constructed completely from scratch. During the year, the rosters are only slightly modified, as was described above. The latter is less complex than constructing a roster completely from scratch.

The basic rosters are constructed in two steps. First, the duties are allocated to the different groups such that the popular and the unpopular work is fairly divided among the groups. Moreover, requirements have to be taken into account to guarantee that a feasible roster can be constructed for each group afterwards. Examples are the average working time per group and the percentage of night duties per group.

In the largest Dutch crew base Utrecht, this step is currently solved as a kind of auction. Each group sends one representative to this auction and every representative tries to get as much popular work as possible for his group. Moreover, the different representatives check if the other groups do not get too much popular work. This is also checked by an independent person. After the duties have been allocated, each representative constructs a roster for his own group, thereby taking into account the labor rules following from the law and the collective labor agreements. This is a complex puzzle, which has to be solved manually. Usually, the representatives stop when they have found a feasible roster. When they cannot find one, they accept this (note that this does not necessarily mean that a feasible roster does not exist!), or they exchange duties with other groups.

4 Allocating the duties to groups

As was described in Sect. 3, the duties are first allocated to the different groups as fairly as possible. This can be seen as an assignment problem where duties should be assigned to different groups. The basic requirements are that each duty should be assigned to exactly one group, and that each group should not have more duties than the number that could be assigned to that group on each individual day. Moreover, some

restrictions should be taken into account such that each group has a large probability of being able to construct a feasible roster afterwards (see Sect. 5).

To allocate the work as fairly as possible to the groups, one could apply two strategies. The first one is that constraints are added which deal with the minimum (maximum) amount of (un)popular work. Alternatively, one could take as objective that the deviation from the average of popular and unpopular work should be as small as possible for each group. We choose for a combination of these two strategies.

The description above can be formulated in a straightforward way as a mixed-integer program (MIP) with 0/1 decision variables indicating whether a duty should be assigned to a group or not. We solve this MIP with the standard commercial solver Cplex 9.1, where we put the emphasis on finding good feasible solutions quickly. Moreover, we use Cplex's local branching heuristic in every fifth node of the branch-and-bound tree. In this way, we could easily find a fair division in a reasonable time. In Sect. 6, we provide some detailed results for an experiment in the crew base Utrecht.

5 Constructing the rosters

5.1 Introduction

When the duties have been allocated to the different groups, each group has to construct its own feasible roster. Here, one should take into account all kinds of constraints following from the labor rules and the collective labor agreements.

Next to duties and days off, certain other types of days should be scheduled. These are the so-called WTV, CO and RES days. A WTV day is in principle a day off for a crew member. However, a crew member can decide to sell this day off and go to work. For the rostering process this means that it should be possible to schedule a duty on a WTV day. A CO day is an extra day off that crew members get when they have worked a certain amount of time in the weekend, early in the morning or in the night. If an individual crew member does not reach a certain level of CO time (e.g. due to illness), he should work on this day. A RES day is a day where a crew member should be available for work in order to replace other crew members that are ill or are on vacation. In the following, a roster day represents either a duty, a day off, a WTV day, a CO day, or a RES day.

Below, we give some examples of rules that have to be taken into account in the CCRP. For an extensive overview, we refer to (Hartog 2005).

- The rest time between two duties is at least 12 hours if the first duty finishes before 2:00 AM, otherwise it is 14 hours.
- The maximum working time per week is 45 hours. For a period of 13 consecutive weeks, the average working time per week is at most 40 hours.
- A day off is a period of at least 30 hours between two duties.
- At least once in the three weeks, there is a so-called *Red Weekend*. This is a rest period of at least 60 hours, starting not later than Saturday at midnight and ending not earlier than Monday 4:00 AM.

- In a full-time roster of length n , there should be planned $2n$ days off and $n/2$ WTV-days.
- There are at most 10 duties per 4 weeks, and 32 per 13 weeks, which cover a part of the period between midnight and 6:00 AM.

The quality of a roster is determined by the order of the roster days and the variety in the duties. Some examples are:

- A series of duties after each other with the same type is preferred, where a type is either an early, late or night duty.
- Two or more adjacent days off (or CO-days) are preferred.
- WTV-days are preferably spread over the different days in the week and are preferably scheduled once in the two weeks.
- Similar duties should be spread over the roster. A lot of variety in the work with respect to trajectories, rolling stock types, etc. is preferred.

5.2 Split-up of the solution process

Since the CCRP is a hard problem to solve, we split it into two phases. In the first phase, we create a pattern where we assign to each day in the roster an early, late or night duty, a day off, a WTV day, a CO day, or a RES day. In the second phase, we assign the specific duties to the places in the roster. Note that this split is rather similar to the split that is described in (Sodhi and Norris 2004).

Both phases can be formulated as an assignment problem with additional constraints. The objective is to minimize the total sum of the penalties, which are determined by undesirable combinations of duties, days off, etc. The details are provided in the following sections.

5.3 Notation

In the following, T denotes the set of days in the roster, where $|T| = 7k$ for a roster of size k . Since the roster is cyclic, day $|T| + 1$ is the same day as day 1. We define I as the set of roster days. Note that in the first phase the specific duties are not considered. Therefore, only the following roster days exist: early, late and night duties, days off, and CO, WTV and RES days. All possible combinations of different types of roster days and different weekdays (i.e. Monday, Tuesday, etc.) are denoted by the set M , where for each combination m the number of roster days is denoted as b_m . For some type of roster days (e.g. WTV days), there is no split to the different weekdays a priori, in other words it still has to be decided in the CCRP. In that case, b_m gives the total number of these roster days and only one constraint for this type of roster days is added.

Finally, we define two types of decision variables. The assignment variable w_{it} is equal to 1 if roster day i is assigned to day t and 0 otherwise. Note that we only define these variables for feasible assignments, i.e. a roster day containing a Monday duty can only be assigned to a Monday. Therefore, we define the set TP as all possible combinations of assignments of roster days to days. TP_m is the subset of TP for combination m of a type of roster day and weekday, e.g. all early duties on Mondays.

The second type of decision variables deals with undesirable patterns of adjacent roster days. These patterns (denoted by the set P) are penalized in the objective function. For each possible pattern p that starts on day t , we define a binary variable y_{pt} , which is 1 if this pattern starts there and 0 otherwise. Its corresponding penalty is denoted by c_{pt} . Note that the value of this penalty can be different on different days of the week.

5.4 Mathematical formulation

In both phases, the mathematical formulation reads as follows, where the horizontal dots indicate constraints that are added and explained later on:

$$\min \sum_{p \in P} \sum_{t \in T} c_{pt} y_{pt}, \tag{1}$$

$$\sum_{i: (i,t) \in TP} w_{it} = 1, \quad \forall t \in T, \tag{2}$$

$$\sum_{(i,t) \in TP_m} w_{it} = b_m, \quad \forall m \in M, \tag{3}$$

⋮

$$y_{1,t} \geq \sum_{\substack{i: \text{earlyduty}, \\ (i,t) \in TP}} w_{it} + \sum_{\substack{j: \text{nightduty}, \\ (j,t+1) \in TP}} w_{j,t+1} - 1, \quad \forall t \in T, \tag{4}$$

⋮

$$w_{it} \in \{0, 1\}, \quad \forall (i, t) \in TP, \tag{5}$$

$$y_{pt} \in \{0, 1\}, \quad \forall p \in P, t \in T. \tag{6}$$

The objective (1) is to minimize the total sum of the penalties assigned to the different undesirable patterns. Constraints (2) and (3) assure that each day has exactly one roster day assigned to it and that each roster day occurs on a particular weekday exactly the right number of times, respectively. An example of an undesirable pattern is a night duty after an early duty. The corresponding set of constraints (4) says that if there is an early duty on day t and a night duty on day $t + 1$, then the corresponding y -variable to this pattern $y_{1,t}$ should be equal to 1. In that case, we count the corresponding penalty in the objective. Since the roster is cyclic, the first day is the successor of the last day. In a similar way, all other undesirable patterns can be defined. In our computations, the set P contained 48 undesirable patterns. Moreover, all labor rules can be added in a similar way. We discuss some examples in the next subsection.

5.5 Labor rules constraints

In the beginning of this section, we gave a few examples of labor rules. We will now show for these examples how we take them into account.

The first example deals with the rest time between two duties. Since in the first phase we do not take the specific duties into account, we cannot take this rule directly into account. However, we know that, if we do not take it into account at all, it is likely that we cannot find a feasible solution in the second phase. Therefore, we do not allow some combinations in the first phase that give a large probability of a too short rest time. Alternatively, we could allow them but against a high penalty. In other words, we could translate this rule in both phases into a rule, which says that duty i could not be followed by duty j on the next day, or in mathematical terms:

$$w_{it} + w_{j,t+1} \leq 1. \tag{7}$$

The maximum working time per week (13 weeks) is the second example. This rule is taken into account in the second phase. By defining a_i as the working time of roster day i , A as the maximum working time, and s as the number of days in this period, we can formulate this as

$$\sum_{n=t}^{t+s-1} \sum_{i:(i,n) \in TP} a_i w_{in} \leq A \quad \forall \text{Mondays } t. \tag{8}$$

The next example deals with the fact that a day off should cover at least 30 hours. In other words, it tells something about the time between two duties on days t and $t + 2$ with a day off in between. The following constraints are added to the second phase:

$$w_{ht} + \sum_{\substack{\text{dayoff } i: \\ (i,t+1) \in TP}} w_{i,t+1} + w_{j,t+2} \leq 2 \quad \forall t \in T, \tag{9}$$

where the start time of duty j is less than 30 hours after the end time of duty h and $(h, t), (j, t + 2) \in TP$.

The so-called ‘‘Red Weekend’’ is an example, which results in constraints in the first and second phase. In both phases, we add the constraints:

$$\sum_{p \in RW} y_{p,t} + y_{p,t+7} + y_{p,t+14} \geq 1 \quad \forall \text{Fridays } t, \tag{10}$$

where the set RW defines all patterns containing a Red Weekend. The difference between the first and second phase is in the definition of RW . In the second phase, we take the end time of the duty on Friday and the start time of the duty on Monday into account as well.

The number of days off and WTV days are taken into account directly in the basic model in the first phase. We do not add extra constraints here.

Our last example deals with the duties which cover a part of the period between midnight and 6:00 AM.¹ These duties form the set ZN . We define s as the period

¹ All night duties cover a part of this period. However, some early and late duties cover a part of this period as well.

length in days in which at most Z of these duties are allowed. This can be modeled as:

$$\sum_{n=t}^{t+s-1} \sum_{\substack{i \in ZN: \\ (i,n) \in TP}} w_{in} \leq Z \quad \forall \text{Mondays } t. \quad (11)$$

This constraint is taken into account in the second phase only.

5.6 Computational results

We have made some experiments for the crew base Utrecht of NS, which contains 12 groups of train drivers. Some characteristics of the groups are given in Table 1. These data correspond to the year 2005.

We have developed and tested our model on 7 of these groups. The computational results for the first phase are given in Table 2. In this table, the objective values, lower bounds and computation times are reported every time that a significantly better feasible solution is found. In addition, we report these values after a reasonable long computation time.

For group DO, we can see that we find after 1 minute a solution with value 12.4. At that moment, the lower bound is 4.1 and the gap is very large. However, for this instance we could prove after 17.5 hours of computation time that the solution with value 12.4 is indeed the optimal one. For the larger instances, we could not prove this. However, we noticed that the solution is never improving after 30 minutes of computation time, while the lower bound only slightly increases. We believe that at the end one could prove that the solution we found is indeed the optimal one (or very close to it). So we conclude that we find a “good” solution quickly.

Table 1 Driver groups of the crew base Utrecht

Group	Size	Average working time	Specifics
A	12	36 hours	no night duties
B	20	36 hours	no night duties
C	20	36 hours	no night duties
D	20	36 hours	all types
E	20	36 hours	all types
O1	20	36 hours	only early duties
O2	10	36 hours	only early duties
L1	20	36 hours	no early duties
L2	12	36 hours	no early duties
DO	10	32 hours	only early duties
DR	12	32 hours	all types
DL	12	32 hours	no early duties

Table 2 Results first phase

Group	Computation time	Objective value	Lower bound
A	1 min	37.3	0
	10 min	34	0.43
	1 hour	34	2.6
B	23 min	80.8	6.2
	45 min	58.9	6.6
	2 hours	58.9	7.5
D	48 min	99.9	6.2
	6 hours	82.6	6.9
	11 hours	68.9	7.5
	16 hours	68.9	7.5
O2	1 min	42.6	0
	16 min	27.4	3.7
	1 hour	27.4	6.6
L2	3 min	58	0
	30 min	37.1	0.5
	1 hour	37.1	1.3
DO	1 min	12.4	4.1
	17,5 hours	12.4 (optimal)	–
DR	1 min	37.3	2.3
	20 min	16.5	4.9
	1 hour	16.5	7

To help Cplex finding good feasible solutions early in the process and therefore to reduce computing time, we have added a redundant constraint:

$$\sum_{p \in P} \sum_{t \in T} c_{pt} y_{pt} \leq \alpha k \quad (12)$$

for an appropriate choice of α . This gives an upper bound on the objective function, which depends on the size of the roster as well as on the penalties for the undesirable patterns. α should be selected with care in order to assure that the feasible region is non-empty.

For the second phase, the results are given in Table 3. For almost all groups an optimal solution can be found within a reasonable amount of time. An exception is group B, where no optimal solution is found after more than 20 hours of computing time. Adding constraint (12) gives much better results for group B. After 7 minutes a solution with objective 231.6 and lower bound 67.8 is found and after two hours for the same solution the lower bound has increased to 100. This shows that adding the constraint can lead to finding better results in shorter computing time.

As an example of the solutions found we present the obtained roster for group D in Table 4. This roster is a full-time roster for 20 weeks (crew members) and contains all types of duties. Early, late and night duties are denoted by E, L and N, respectively. A day off is denoted by R. The solution for the first phase is found after 11 hours with

Table 3 Results second phase

Group	Computation time	Objective value	Lower bound
A	2.5 min	209	68
	1.7 hours	93.6	90.5
	8.8 hours	93.6 (optimal)	–
B	30 min	752.3	76.2
	4 hours	583.7	84.3
	20 hours	583.7	90
D	8 min	871.1	149
	10 min	245.5	170
	3.2 hours	217.5 (optimal)	–
O2	2 min	56	39
	3 min	48	40
	7 hours	48 (optimal)	–
L2	2 min	33	12.8
	17 min	27 (optimal)	–
DO	2 min	150	32.9
	3.6 hours	73	68
	11 hours	73 (optimal)	–
DR	1.5 min	129.3 (optimal)	–

Table 4 Example roster group D

Wk	Mo	Tu	We	Th	Fr	Sa	Su
1	RES	E	R	L	WTV	R	R
2	E	E	E	E	E	E	R
3	R	L	WTV	R	R	E	E
4	E	E	E	E	E	R	R
5	L	L	WTV	R	L	N	N
6	N	R	R	L	L	L	L
7	WTV	R	L	L	RES	R	R
8	E	E	E	E	E	R	R
9	L	WTV	R	R	N	N	R
10	R	L	L	RES	CO	E	E
11	WTV	R	E	E	E	R	R
12	L	N	N	R	L	L	L
13	R	R	RES	L	L	L	L
14	CO	L	L	L	WTV	R	R
15	L	L	L	L	L	R	R
16	E	WTV	R	R	L	L	L
17	R	RES	E	E	E	R	R
18	L	L	L	WTV	R	E	L
19	L	N	N	N	CO	R	R
20	L	L	L	WTV	R	E	E

an objective value of 68.9 and a lower bound of 7.5. For the second phase, the optimal solution is found after 3 hours with an objective value of 217.5. Only the results of the first phase are presented in the table.

First we give some remarks about the results of the first phase. Within the roster the WTV days are nicely spread over the weekdays and the weeks and are all followed by a rest day. Ten times there is a single day off, but in most of the cases days off are clustered. Three times, a night duty is scheduled after a late duty, and once a night duty is scheduled after an early duty. These are undesirable combinations. However, in general the sequencing of the duties is quite attractive. The next remarks are related to the results of the second phase. This roster contains only two weeks with an average working time above 36 hours. The roster contains two duties with a post time of 13 hours, seven with a post time of 14 hours, 15 with a post time of 15 hours and 57 duties with a post time of more than 16 hours. Most of the routes occur in every week in less than 3 duties (which spreads the routes equally over the weeks). Also the nice trains have been spread over the weeks fairly.

6 Results shadow planning train conductors

The conductors in the crew base Utrecht had some problems with the creation of the rosters in the previous year. Therefore, the management asked us to support the planning of the rosters for the year 2006. The idea was to generate rosters parallel to the manual planning process and to evaluate the differences.

As mentioned above, the first step in the rostering process is to allocate the duties to the different groups of conductors. The problem at hand consisted of 14 groups with 3 general, 3 late, 3 early and 5 part time rosters. Representatives of the different groups took 3 days to allocate the duties over the different groups. The popular and unpopular work was allocated reasonably fairly to the different groups. However, there was one group with a lot of unpopular work. By using the approach described in Sect. 4, we could allocate the duties in such a way that almost all the popular and unpopular work was allocated more or less fairly. The solution time to solve the optimization problem was about 20 minutes, which is much less than the 3 days needed by the representatives. Moreover, they modified their allocation when they saw our solution, such that the group with a lot of unpopular work got some other duties in order to increase the overall fairness of the allocation.

In the next step, the construction of the cyclic rosters, the main practical problem was related to the second phase of our approach. In principle, the groups use the same pattern for the rosters as in the previous year. In our approach the patterns are created in the first phase, so this phase was less important. Based on the patterns, an attractive solution for the second phase was found within one hour for all groups, except for group C. In this case it was not possible to find a feasible solution based on the given pattern. After generating a new pattern, we were able to find a good roster for group C. For the smaller or part time rosters we found an optimal solution almost instantly. The larger cases took some hours to be solved to optimality.

Manually, the planners were not able to find a feasible solution for three rosters within the available time. In these cases, we presented both the manual roster and

the roster generated by the model, where the first ones did not obey all labor rules. The violated rules were related to the official national regulations. However, these violations are not commonly known and hard to detect. We did not tell the personnel which rosters were generated by the model and which ones were generated manually. In all three cases the personnel preferred the rosters generated by the model. As a result of this, three of the rosters generated by the model have been implemented in practice. This shows that our model is not only successful in creating valid rosters but also creates attractive rosters.

7 Conclusions

In this paper we described a model for solving the CCRP. The model was tested on several instances involving groups of train drivers or conductors of NS, the largest operator of passenger trains in the Netherlands. The results were promising: rosters for the crew base Utrecht were generated in a fraction of the time that was required by the manual rostering process. Moreover, the rosters generated by the model satisfied all the relevant rules, which was not the case for the manually generated rosters. Moreover, the rosters generated by the model were preferred over the manually generated ones.

A next step in our research is to test the model also in the other crew bases of NS, where the rostering process may be slightly different than in the crew base Utrecht. Another next step is to add the model to the existing rostering system of NS. If these steps have been accomplished, then the model can be used at all crew bases of NS.

Summarizing, we have shown once more that the application of Operations Research techniques for solving a practical problem may lead to a shorter throughput time in the planning process, as well as to a higher quality of the obtained results. In particular, since the results are appreciated by the personnel, this will lead to more satisfied personnel. It can be expected that more satisfied personnel will lead to a higher productivity of the personnel and to a better punctuality of the railway services.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

- Abbink E, Fischetti M, Kroon L, Timmer G, Vromans M (2005) Reinventing crew scheduling at Netherlands railways. *Interfaces* 35(5):393–401
- Caprara A, Fischetti M, Toth P, Vigo D (1998) Modeling and solving the crew rostering problem. *Oper Res* 46(6):820–830
- Hartog A (2005) Wiskundige modellen voor het maken van dienstroosters voor het treinpersoneel. Master's thesis, Free University Amsterdam (in Dutch)
- Huisman D (2007) A column generation approach to solve the rail crew re-scheduling problem. *Eur J Oper Res* 180(1):163–173
- Huisman D, Kroon LG, Lentink RM, Vromans MJCM (2005) Operations research in passenger railway transportation. *Stat Neerl* 59(4):467–497
- Kohl N, Karisch SE (2004) Airline crew rostering: problem types, modeling and optimization. *Ann Oper Res* 127:223–257
- Sodhi M, Norris S (2004) A flexible, fast, and optimal modeling approach applied to crew rostering at London underground. *Ann Oper Res* 127:259–281