Integer Programming and Heuristic Approaches for a Multi-Stage Nurse Rostering Problem

Florian Mischek · Nysret Musliu

Abstract In the variant of the well studied nurse rostering problem proposed in the Second International Nurse Rostering Competition, multiple stages have to be solved sequentially which are dependent on each other. We propose an integer programming model for this problem and show that extensions in the form of additional constraints to deal with the incomplete information are necessary to achieve competitive results in this setting. Furthermore, we propose a local search framework based on a combination of Min-Conflict and Tabu search. We compare our solution approaches with the results obtained in the competiton.

Keywords Nurse rostering · INRC-II · Integer Programming · Local Search

1 Introduction

The automated generation of high quality staff schedules, in particular for hospitals, has been an important problem for over 40 years. Multiple variants and solution approaches exist (to be found e.g. in surveys by Ernst et al. [7] and Van den Bergh et al. [1]). A survey focused on variants of the Nurse Rostering Problem can be found in [3].

In 2015, Ceschia et al. [6] proposed a variant of the nurse rostering problem for the Second International Nurse Rostering Competition (INRC-II). In contrast to previous problem variants, a multi-stage formulation is used, where solutions for individual weeks have to be produced by the solver sequentially, without information about the requirements of later weeks. In [14], such a setting was denoted a *stepping horizon* approach.

This multi-stage setting poses two unique challenges for solvers: The dependencies between weeks make it necessary to take the solutions of previous weeks into account during the evaluation of the quality of a schedule. This is largely covered in the rules for constraint evaluation of the INRC-II. Other publications treating this issue are [8], [14] and [17].

Vienna University of Technology
Database and Artificial Intelligence group
E-mail: {fmischek,musliu}@dbai.tuwien.ac.at

Further, and not explored in the previously mentioned papers, is the the fact that due to the incomplete information during all weeks but the last, the generated solution can no longer be guaranteed to be optimal even if each week is solved to optimality. More so, a naive model that is not adapted to this setting will produce imbalanced schedules that incur large penalties in later weeks as options are restricted excessively by the solutions of the previous weeks.

There have been 15 submissions to the INRC-II, with seven of these advancing to the final round. The results are available on the competition website[1], but details about solution methods have not yet been published (apart from an abstract by the competition winners [13], who used a network flow-based approach.) While many approaches using integer programming (IP) formulations for other nurse rostering problems exist (e.g. [16], [5], [2]), to the best of our knowledge there are no publications using IP for this exact problem.

In fact, the only publication we are aware of dealing with this exact problem so far is by Santos et al. [15], who used a weighted constraint satisfaction approach.

In this paper, we first propose a basic IP formulation (Section 3) for the INRC-II problem (as defined in Section 2). We then extend this model with additional constraints to account for the multi-stage setting in Section 4. Additionally, we propose a local search based heuristic that uses the same additional constraints (Section 5). In Section 6, we evaluate our formulations using the instances provided for the INRC-II and show that the additional constraints significantly improve the quality of the generated solutions. We also compare with the results of our local search algorithm and those of the finalists in the INRC-II.

2 Problem Definition

In this section, we give a short overview of the problem used in the INRC-II. A detailed description of the problem structure and all constraints can be found in [6].

For each week in the instance (either 4 or 8), a schedule has to be found by the solver, using only the information provided in a global *scenario* file, containing information about the nurses and their contracts, *week data* about the requirements of the current week and a *history* with data concerning the last assignments of the previous weeks and some global counters. Information about the following weeks, in particular about the covering requirements, is not available until the solution for the current week has been fixed by the solver.

In the following, a *work stretch* denotes a period of consecutive working days for a nurse. *Rest stretch* and *shift stretch* are analogously defined for periods of consecutive days off and assignments to the same shift, respectively.

There are four hard constraints that have to be fulfilled by any solution to be regarded as feasible:

H1. Single assignment per day: Each nurse can only work a single shift using a single skill per day.

H2. Under-staffing: The minimum number of nurses required for each shift and skill must be present.

[1] http://mobiz.vives.be/inrc2/?page_id=226

H3. Shift type successions: Nurses must not have shifts on two consecutive days that form a forbidden sequence.
H4. Missing required skill: Nurses can only cover assignments for which they have the required skill.

Further, seven soft constraints are defined. Solutions should try to satisfy these constraints, but violating them only results in a penalty to the quality of the solution (weights are listed in the description of each constraint).

S1. Insufficient staffing for optimal coverage (30): The number of nurses assigned to each shift and skill should not be smaller than the optimum staffing. The penalty is multiplied by the number of missing nurses.
S2. Consecutive assignments (15/30): The length of each shift stretch (weight 15) and work stretch (weight 30) should be within the bounds defined for the shift type resp. the contract of the involved nurse. The penalty is multiplied by the number of missing or surplus assignments.
S3. Consecutive days off (30): As before, the length of each rest stretch should be within the bounds defined in each nurse's contract. The penalty is multiplied by the number of missing or surplus days off.
S4. Preferences (10): The requests of nurses for shifts (or days) off should be respected.
S5. Complete week-end (30): Nurses with the complete-weekend constraint in their contract should either work both days of the weekend or none.
S6. Total assignments (20): Over the whole planning horizon, each nurse's assignments should be within the bounds defined in their contract.
S7. Total working week-ends (30): Over the whole planning horizon, each nurse should not work more than the maximum number of weekends defined in their contract.

Constraints S6 and S7 are evaluated only after the solution for the last week has been fixed, although they should of course be respected by solvers during all weeks. All sequence constraints (H3, S2, S3) also use the border data from the solution for the previous week (this is provided in the history file).

3 Basic Model

3.1 Parameters

The first set of parameters contains values that stay the same over the whole planning horizon. These values are stored in the *scenario* file:

N	set of nurses
S	set of shifts
K	set of skills
$\|W\|$	number of weeks
$a_n^{[+/-]}$	maximum/minimum assignments for nurse n across planning horizon
$w_n^{[+/-]}$	maximum/minimum consecutive working days for nurse n
$f_n^{[+/-]}$	maximum/minimum consecutive days off for nurse n
t_n^+	maximum number of working weekends for nurse n across planning horizon
b_n	boolean, 1 iff either both days of a weekend should be worked by nurse n, or none
κ_{nk}	boolean, 1 iff nurse n has skill k
$\sigma_s^{[+/-]}$	maximum/minimum consecutive assignments of shift s
u_{st}	boolean, 1 iff shift t may be assigned the day after an assignment of shift s

The next set of parameters is defined for each *week*.

w	number of the current week
c_{sk}^d	minimum cover requirements for day d, shift s and skill k
o_{sk}^d	optimum cover requirements for day d, shift s and skill k
r_{ns}^d	boolean, 1 iff nurse n requested not to work in shift s on day d ($s = 0$ is day-off request)

Finally, these parameters specify values depending on the schedule of the previous week. This *history* is given for the first week and calculated from the solution of the last week for all subsequent weeks.

l_n^{id}	id of last shift worked by nurse n in previous week (0 if day off)
l_{ns}	consecutive shifts of type s worked by nurse n at the end of the previous week (0 if $s \neq l_n^{id}$)
l_n^w	consecutive working days for nurse n at the end of the previous week (0 if $l_n^{id} = 0$)
l_n^f	consecutive days off for nurse n at the end of the previous week (0 if $l_n^{id} \neq 0$)
a_n^{tot}	total number of assignments for nurse n so far
t_n^{tot}	total number of weekends worked by nurse n so far

3.2 Decision variables

$$x_{nsk}^d \in \{0,1\} \quad \forall n \in N, s \in S, k \in K, d \in \{1...7\}$$
$$W_n \in \{0,1\} \quad \forall n \in N$$

$x_{nsk}^d = 1$ if nurse n is assigned to shift s using skill k on day d, and 0 otherwise. The W_n variable indicates that nurse n works at least one day of the weekend.

The violation of soft constraints is measured using either non-negative or boolean surplus variables:

C^{S1}_{skd}	≥ 0	missing nurses for optimal coverage of shift s, skill k on day d
C^{S2a}_{nsd}	≥ 0	missing days in the block of shifts s starting on day d for nurse n
C^{S2b}_{nsd}	$\in \{0,1\}$	1 iff shift s of nurse n on day d violates maximum consecutive shifts
C^{S2c}_{nd}	≥ 0	missing days in the work block of nurse n starting on day d
C^{S2d}_{nd}	$\in \{0,1\}$	1 iff work of nurse n on day d violates maximum consecutive work days
C^{S3a}_{nd}	≥ 0	missing days in the free block of nurse n starting on day d
C^{S3b}_{nd}	$\in \{0,1\}$	1 iff day off of nurse n on day d violates maximum consecutive days off
C^{S4}_{nd}	$\in \{0,1\}$	1 iff assignment on day d violates a request of nurse n
C^{S5}_{n}	$\in \{0,1\}$	1 iff nurse n violates complete weekend constraint
C^{S6}_{n}	≥ 0	number of total shifts outside the allowed bounds for nurse n
C^{S7}_{n}	≥ 0	number of weekends worked above the maximum by nurse n

3.3 Objective function

The objective function is the weighted sum over all violations of each soft constraint:

$$
\begin{aligned}
minimize\ f = \quad & 30 * \sum_{\substack{s\in S\\ k\in K\\ d\in\{1\ldots 7\}}} C^{S1}_{skd} \\
& +15 * \sum_{\substack{n\in N\\ s\in S\\ d\in\{1\ldots 7\}}} (C^{S2a}_{nsd} + C^{S2b}_{nsd}) \\
& +30 * \sum_{\substack{n\in N\\ d\in\{1\ldots 7\}}} (C^{S2c}_{nd} + C^{S2d}_{nd}) \\
& +30 * \sum_{\substack{n\in N\\ d\in\{1\ldots 7\}}} (C^{S3a}_{nd} + C^{S3b}_{nd}) \\
& +10 * \sum_{\substack{n\in N\\ d\in\{1\ldots 7\}}} C^{S4}_{nd} \\
& +30 * \sum_{n\in N} C^{S5}_{n} \\
& +20 * \sum_{n\in N} C^{S6}_{n} \\
& +30 * \sum_{n\in N} C^{S7}_{n}
\end{aligned}
$$

3.4 Constraints

The following (in)equalities model the hard constraints, as described above.

H1 $\forall n \in N, d \in \{1\ldots 7\}$

$$\sum_{\substack{s\in S\\ k\in K}} x^d_{nsk} \leq 1 \tag{1}$$

$$\text{H2} \quad \forall s \in S, k \in K, d \in \{1 \dots 7\} \qquad \sum_{n \in N} x^d_{nsk} \geq c^d_{sk} \tag{2}$$

For constraint H3, any forbidden shift sequence ($u_{s_1 s_2} = 0$) must not be assigned to the same nurse on consecutive days. This must be ensured both within the week (a) and at the boundary of this week with the previous one (i.e. on the first day of the week, b).

$$\text{H3a} \quad \forall n \in N, s_1, s_2 \in S, k \in K, d \in \{1 \dots 6\} : u_{s_1 s_2} = 0 \qquad \sum_{k \in K} x^d_{ns_1k} + \sum_{k \in K} x^{d+1}_{ns_2k} \leq 1 \tag{3}$$

$$\text{H3b} \quad \forall n \in N, s \in S, k \in K : u_{l^{id}_n s} = 0 \qquad x^1_{nsk} = 0 \tag{4}$$

$$\text{H4} \quad \forall n \in N, s \in S, d \in \{1 \dots 7\}, k \in K : \kappa_{nk} = 0 \qquad x^d_{nsk} = 0 \tag{5}$$

The remaining inequalities deal with the soft constraints. Each inequality can be deactivated by setting the appropriate surplus variable to a value greater than zero, which results in a corresponding penalty in the objective function.

$$\text{S1} \quad \forall s \in S, k \in K, d \in \{1 \dots 7\} \qquad \sum_{n \in N} x^d_{nsk} \geq o^d_{sk} - C^{S1}_{skd} \tag{6}$$

S2 actually contains various different constraints that have to be modeled separately: consecutive assignments of the same shift (min (a)/ max (b)) and of work in general (min (c) / max (d)), both during and at the start of the week.

For the minimum consecutive shifts constraints, all patterns that compose a sequence shorter than the required length are prevented. For example, if the minimum number of consecutive night shifts (N) is 4, the patterns {xNx, xNNx, xNNNx}, where x is any other shift or a day off, should not appear.

Since each pattern incurs a penalty proportional to the number of missing assignments, (in the example, xNx would incur a penalty of 45, while xNNNx would incur a penalty of 15) the surplus variables are weighted correspondingly, to ensure that a value of at least the number of missing assignments is necessary to deactivate the constraint.

Equations 8 and 9 model the case where a stretch starts at the beginning of the week or towards the end of the previous week.

$$\text{S2a} \quad \forall s \in S, n \in N, b \in \{1 \dots (\sigma^-_s - 1)\}, d \in \{1 \dots 7 - (b+1)\}$$

$$\sum_{k \in K} (x^d_{nsk} + \sum_{i \in \{1 \dots b\}} (1 - x^{d+i}_{nsk}) + x^{d+b+1}_{nsk}) \geq 1 - \frac{C^{S2a}_{ns(d+1)}}{\sigma^-_s - b} \tag{7}$$

$$\forall s \in S, n \in N, b \in \{1 \dots (\sigma^-_s - 1 - l_{ns})\}$$

$$\sum_{k \in K} (\sum_{i \in \{1 \dots b\}} (1 - x^i_{nsk}) + x^{b+1}_{nsk}) \geq 1 - \frac{C^{S2a}_{ns1}}{\sigma^-_s - l_{ns} - b} \tag{8}$$

$$\forall s \in S, n \in N : l_n^{id} = s \land l_{ns} < \sigma_s^-$$
$$\sum_{k \in K} x_{nsk}^1 \geq 1 \;-\; \frac{C_{ns1}^{S2a}}{\sigma_s^- - l_{ns}} \tag{9}$$

The maximum consecutive shifts constraints is modeled like this: For each shift s with a maximum of σ_s^+ consecutive assignments, each block of $\sigma_s^+ + 1$ days must contain at least one day where s is not assigned. Note that contrary to the situation for S2a, violations of this constraint by more than one shift assignment result in multiple matches of the pattern and therefore it suffices to use boolean surplus variables.

As before, equations 11 model the case where a shift block started in the previous week.

S2b $\forall s \in S, n \in N, d \in \{1 \ldots (7 - \sigma_s^+)\}$
$$\sum_{k \in K} \sum_{i \in \{0 \ldots \sigma_s^+\}} x_{nsk}^{d+i} \leq \sigma_s^+ \;+\; C_{ns(d+\sigma_s^+)}^{S2b} \tag{10}$$
$$\forall s \in S, n \in N, b \in \{(\sigma_s^+ - l_{ns} + 1) \ldots \sigma_s^+\} : l_n^{id} = s$$
$$\sum_{k \in K} \sum_{i \in \{1 \ldots b\}} x_{nsk}^{i} \leq b - 1 \;+\; C_{nsb}^{S2b} \tag{11}$$

The inequalities modelling the maximum and minimum length of work stretches (S2c, S2d) function analogously to those for shift stretches. The only difference is that an assignment to any shift counts towards the length of the work stretch.

S2c $\forall n \in N, b \in \{1 \ldots (w_n^- - 1)\}, d \in \{1 \ldots 7 - (b+1)\}$
$$\sum_{\substack{s \in S \\ k \in K}} (x_{nsk}^d + \sum_{i \in \{1 \ldots b\}} (1 - x_{nsk}^{d+i}) + x_{nsk}^{d+b+1}) \geq 1 \;-\; \frac{C_{n(d+1)}^{S2c}}{w_n^- - b} \tag{12}$$
$$\forall n \in N, b \in \{1 \ldots (w_n^- - 1 - l_n^w)\}$$
$$\sum_{\substack{s \in S \\ k \in K}} (\sum_{i \in \{1 \ldots b\}} (1 - x_{nsk}^{i}) + x_{nsk}^{b+1}) \geq 1 \;-\; \frac{C_{n1}^{S2c}}{w_n^- - l_n^w - b} \tag{13}$$
$$\forall n \in N : l_n^{id} \neq 0 \land l_n^w < w_n^-$$
$$\sum_{\substack{s \in S \\ k \in K}} x_{nsk}^1 \geq 1 \;-\; \frac{C_{n1}^{S2c}}{w_n^- - l_n^w} \tag{14}$$

S2d $\forall n \in N, d \in \{1 \ldots (7 - w_n^+)\}$
$$\sum_{\substack{s \in S \\ k \in K}} \sum_{i \in \{0 \ldots w_n^+\}} x_{nsk}^{d+i} \leq w_n^+ \;+\; C_{n(d+w_n^+)}^{S2d} \tag{15}$$
$$\forall n \in N, b \in \{(w_n^+ - l_n^w + 1) \ldots w_n^+\} : l_n^{id} \neq 0$$
$$\sum_{\substack{s \in S \\ k \in K}} \sum_{i \in \{1 \ldots b\}} x_{nsk}^{i} \leq b - 1 \;+\; C_{nb}^{S2d} \tag{16}$$

S3 similarily contains two independent constraints: the minimum (a) and maximum (b) number of consecutive days off, again both during and at the start of the week.

The equations modelling these constraints are again analoguous to those from constraints S2c and S2d, except that days of work and days off were swapped.

S3a $\forall n \in N, b \in \{1 \ldots (f_n^- - 1)\}, d \in \{1 \ldots 7 - (b+1)\}$

$$\sum_{\substack{s \in S \\ k \in K}} ((1 - x_{nsk}^d) + \sum_{i \in \{1 \ldots b\}} x_{nsk}^{d+i} + (1 - x_{nsk}^{d+b+1})) \geq 1 - \frac{C_{n(d+1)}^{S3a}}{f_n^- - b} \tag{17}$$

$\forall n \in N, b \in \{1 \ldots (f_n^- - 1 - l_n^f)\}$

$$\sum_{\substack{s \in S \\ k \in K}} (\sum_{i \in \{1 \ldots b\}} x_{nsk}^i - x_{nsk}^{b+1}) \geq 0 - \frac{C_{n1}^{S3a}}{f_n^- - l_n^f - b} \tag{18}$$

$\forall n \in N : l_n^{id} = 0 \wedge l_n^f < f_n^-$

$$\sum_{\substack{s \in S \\ k \in K}} -x_{nsk}^1 \geq 0 - \frac{C_{n1}^{S3a}}{f_n^- - l_n^f} \tag{19}$$

S3b $\forall n \in N, d \in \{1 \ldots (7 - f_n^+)\}$

$$\sum_{\substack{s \in S \\ k \in K}} \sum_{i \in \{0 \ldots f_n^+\}} x_{nsk}^{d+i} \geq 1 - C_{n(d+f_n^+)}^{S3b} \tag{20}$$

$\forall n \in N, b \in \{(f_n^+ - l_n^f + 1) \ldots f_n^+\} : l_n^{id} = 0$

$$\sum_{\substack{s \in S \\ k \in K}} \sum_{i \in \{1 \ldots b\}} x_{nsk}^i \geq 1 - C_{nb}^{S3b} \tag{21}$$

To model nurse requests for shifts or days off, any assignment to an unwanted shift incurs the penalty.

S4 $\forall n \in N, s \in S, d \in \{1 \ldots 7\} : r_{ns}^d \vee r_{n0}^d$

$$\sum_{k \in K} x_{nsk}^d \leq C_{nd}^{S4} \tag{22}$$

For the complete weekends constraint, first the additional helper variables W_n are set if the nurse n works either of the days on the weekend. Equations 24 then ensure that if W_n is set, and the complete weekend constraint is present for the nurse, both days of the weekend should have work assigned.

S5 $\forall n \in N, d \in \{6, 7\}$

$$\sum_{\substack{s \in S \\ k \in K}} x_{nsk}^d \leq W_n \tag{23}$$

$\forall n \in N : b_n$

$$\sum_{\substack{s \in S \\ k \in K}} (x_{nsk}^6 + x_{nsk}^7) \geq 2W_n - C_n^{S5} \tag{24}$$

The constraint S6 (number of total assignments) is modeled slightly differently from its description in [6]. Originally, these constraints were evaluated only after the schedules of all weeks were fixed. In our model, the penalties are calculated immediately and added to the objective function value of the week in which they arise. This does not change the overall quality of the whole schedule, so results are still comparable, although the intermediate quality value of the individual weeks might be different.

$$\text{S6} \quad \forall n \in N \quad \sum_{\substack{s \in S \\ k \in K \\ d \in \{1 \ldots 7\}}} x^d_{nsk} \leq \max\{a^+_n - a^{tot}_n,\ 0\} \ + \ C^{S6}_n \tag{25}$$

$$\forall n \in N \quad \sum_{\substack{s \in S \\ k \in K \\ d \in \{1 \ldots 7\}}} x^d_{nsk} \geq \min\{a^-_n - 7 * (|W| - w),\ 7\} \ - \ C^{S6}_n \tag{26}$$

The equations for constraint S7 (maximum number of weekends worked) use the variable W_n, set in equations 23.

$$\text{S7} \quad \forall n \in N \quad t^{tot}_n + W_n \leq t^+_n \ + \ C^{S7}_n \tag{27}$$

4 Model Extensions

While the basic model described in Section 3 yields feasible solutions that are optimal for each week (if given enough time), the connections between weeks are mostly ignored. Because the weeks are solved individually, solutions are favored that give slightly better results in earlier weeks, at the cost of having potentially much larger penalties in later weeks.

In order to take this into account and improve the overall solution quality, we propose the following extensions to the model, in the form of additional (soft) constraints.

While constraints S6 in their modified form are already evaluated each week, they are trivially satisfied in the early weeks. For this reason, nurses get assigned too many or not enough shifts at first to satisfy other constraints, causing problems in later weeks. To alleviate this, we tried to keep the number of assignments per nurse roughly evenly distributed across all weeks.

S6*. Average assignments: The total number of assignments up to the current week must be within the bounds defined in the contract, multiplied by the fraction of weeks that have already passed.

This constraint tries to ensure that after p percent of the weeks have passed, also p percent of the available assignments per nurse have been used. To add it to the IP model, we introduced a new set of surplus variables:

$C_n^{S6*} \quad \geq 0$ number of assignments of nurse n outside the average bounds for the current week

The constraint can then be formalized as:

$$\text{S6*} \quad \forall n \in N$$

$$a_n^{tot} + \sum_{\substack{s \in S \\ k \in K \\ d \in \{1 \ldots 7\}}} x_{nsk}^d \leq \lfloor a_n^+ * \frac{w}{|W|} \rfloor \; + \; C_n^{S6*} \tag{28}$$

$$\forall n \in N$$

$$a_n^{tot} + \sum_{\substack{s \in S \\ k \in K \\ d \in \{1 \ldots 7\}}} x_{nsk}^d \geq \lceil a_n^- * \frac{w}{|W|} \rceil \; - \; C_n^{S6*} \tag{29}$$

Floor and ceiling functions were applied to ensure that any solution satisfying these constraints would also end up satisfying the original constraints S6, while at the same time keeping the penalty values integer. We also experimented with different variants, but this had no significant effect on the solution quality.

An alternative version of constraint S6* can be formulated as follows:

S6*-2. Average assignments, alternative: In each week, the remaining assignments (not yet used in previous weeks) should be divided equally among all remaining weeks.

It can be formalized in a similar way:

$$\text{S6*alt} \quad \forall n \in N$$

$$\sum_{\substack{s \in S \\ k \in K \\ d \in \{1 \ldots 7\}}} x_{nsk}^d \leq \lfloor (a_n^+ - a_n^{tot}) * \frac{1}{|W| - w + 1} \rfloor \; + \; C_n^{S6*} \tag{30}$$

$$\forall n \in N$$

$$\sum_{\substack{s \in S \\ k \in K \\ d \in \{1 \ldots 7\}}} x_{nsk}^d \geq \lceil (a_n^- - a_n^{tot}) * \frac{1}{|W| - w + 1} \rceil \; - \; C_n^{S6*} \tag{31}$$

Analogously, also constraints S7 (maximum total working weekends) can be extended to distribute these weekends evenly across the weeks. This results in

S7*. Average working weekends: In each week, the still available working weekends (not yet used in previous weeks) should be divided equally among all remaining weeks.

with the following formalization:

$C_n^{S7*} \quad \in \{0, 1\}$ 1 iff nurse n works on the weekend despite not having enough working weekends left

$$\text{S7*} \quad \forall n \in N$$

$$W_n \leq \lfloor (t_n^+ - t_n^{tot}) * \frac{1}{|W| - w + 1} \rfloor \; + \; C_n^{S7*} \tag{32}$$

Another point where the solution of one week can impact the solutions of subsequent weeks is at the end of the week. Since all sequence constraints (S2, S3) are also counted between weeks, a solution of one week can restrict the options to assign shifts to nurses without penalty at the beginning of the next week.
For example, let the minimum number of consecutive night shifts (σ^-_{N}) be 4 and the proposed solution for this week end with a single night shift on Sunday for a nurse (and any other shift or a day off on Saturday, compare Figure 1). Then we already know that any assignment for this nurse from Monday to Wednesday that is not a night shift, will inevitably incur a penalty (and depending on the rest of the schedule, assigning only night shifts on these three days could result in penalties of its own).
As another example, if the maximum number of consecutive night shifts is 5 and the proposed solution already contains a shift stretch of at least 5 night shifts in the days leading up to Sunday, this means that assigning a further night shift on Monday of the next week would incur a penalty for exceeding the maximum length.

	Sa	Su	Mo	Tu	We	
...	-	N	N?	N?	N?	...

Fig. 1 Example of the situation constraints S8* try to prevent. Assuming $\sigma^-_{\mathrm{N}} = 4$, a single night shift on Sunday will cause a penalty in the next week if any shifts other than additional night shifts have to be assigned between Monday and Wednesday.

In order to favor schedules that leave as many options as possible for the next week, we propose the following constraint:

S8*. Restriction of next week's assignments: Options for next week's schedule should not be restricted. The penalty is calculated as the total number of shifts that cannot be assigned in the next week without violating at least one sequence constraint.

As before, we use a new set of surplus variables to measure the violation of this constraint:

C^{S8a}_n	≥ 0	number of shifts restricted from next week's schedule of nurse n due to shift sequence constraints
C^{S8b}_n	≥ 0	number of shifts restricted from next week's schedule of nurse n due to work sequence constraints
C^{S8c}_n	≥ 0	number of shifts restricted from next week's schedule of nurse n due to day off sequence constraints

The equations to model this constraint are similar to those for constraints S2 and S3 (equations 7 to 21), except that the patterns are matched only at the end of the week.
The first set of equations (S8*a) deals with restrictions resulting from shift sequence constraints, S8*b deals with work sequence constraints and S8*c with day off sequence constraints. In each set, the first type of equations governs restrictions due to stretches smaller than the minimum length required, while the second type deals with restrictions resulting from reaching the maximum length of a stretch.

Additionally, any time a single shift type would be enforced (like in the first example above, where the assignment of additional night shifts is enforced), the weight of this pattern is multiplied by the number of shifts $|S|$ (since there are $|S|-1$ other shifts and a day off that cannot be assigned without penalty).

S8*a $\forall n \in N, s \in S, b \in \{1 \ldots (\sigma_s^- - 1)\}$

$$\sum_{k \in K} ((1 - x_{nsk}^{7-b}) + \sum_{i \in \{0 \ldots (b-1)\}} x_{nsk}^{7-i}) \leq b + \frac{C_n^{S8a}}{|S|(\sigma_s^- - b)} \tag{33}$$

$\forall n \in N, s \in S$

$$\sum_{k \in K} \sum_{i \in \{0 \ldots (\sigma_s^+ - 1)\}} x_{nsk}^{7-i} \leq \sigma_s^+ - 1 + C_n^{S8a} \tag{34}$$

S8*b $\forall n \in N, b \in \{1 \ldots (w_n^- - 1)\}$

$$\sum_{\substack{s \in S \\ k \in K}} ((1 - x_{nsk}^{7-b}) + \sum_{i \in \{0 \ldots (b-1)\}} x_{nsk}^{7-i}) \leq b + \frac{C_n^{S8b}}{w_n^- - b} \tag{35}$$

$\forall n \in N$

$$\sum_{\substack{s \in S \\ k \in K}} \sum_{i \in \{0 \ldots (w_n^+ - 1)\}} x_{nsk}^{7-i} \leq w_n^+ - 1 + \frac{C_n^{S8b}}{|S|} \tag{36}$$

S8*c $\forall n \in N, b \in \{1 \ldots (f_n^- - 1)\}$

$$\sum_{\substack{s \in S \\ k \in K}} (x_{nsk}^{7-b} - \sum_{i \in \{0 \ldots (b-1)\}} x_{nsk}^{7-i}) \leq 0 + \frac{C_n^{S8c}}{|S|(w_n^- - b)} \tag{37}$$

$\forall n \in N$

$$\sum_{\substack{s \in S \\ k \in K}} \sum_{i \in \{0 \ldots (f_n^+ - 1)\}} x_{nsk}^{7-i} \geq 1 - C_n^{S8c} \tag{38}$$

The changes to the model should also be reflected in the objective function, by adding the new surplus variables:

$$\begin{aligned} minimize\ f' = \quad & f \ + \ W^{S6*} * \sum_{n \in N} C_n^{S6*} \\ & + W^{S7*} * \sum_{n \in N} C_n^{S7*} \\ & + W^{S8*} * \sum_{n \in N} (C_n^{S8a} + C_n^{S8b} + C_n^{S8c}) \end{aligned}$$

where W^{S6*}, W^{S7*} and W^{S8*} are the weights of constraints S6* and S8* respectively.

After a solution has been fixed, the actual penalty has to be recalculated using the objective function of the basic model f, to ensure that the penalties from the additional constraints of the extended model are not included in the final result.

Obviously, all constraints introduced in this section should be ignored in the last week, as there is no further week to influence.

5 Local Search

We also implemented a local search framework for this problem, as a comparison with the results from the IP model. We used a combination of Tabu-Search (TS) [9] and the Min-Conflict heuristic (MC) [10], similar to the framework described in [11]. Starting from an initial solution, at each step, with probability p, one move of the TS strategy is applied, and with probability $1-p$, one move of the MC strategy is applied. After a fixed number of successive moves that did not improve the current best solution, the search is restarted from a new initial solution.

To generate the neighbourhood, we regarded two types of moves:

$change(n, d, s, k)$: sets the shift and skill assignment of nurse n on day d to s and k, respectively. If $s = 0$, this is treated as a day off. Only assignments to a legal skill for this nurse are allowed.

$swap(n, d, n2, l)$: swaps the assignments of the next l days, starting at day d, of nurse n with those of nurse $n2$. As before, only moves that don't result in conflicts with the skills of both nurses are allowed. Blocks up to length $l = 4$ are considered in each step.

To reduce the time until the first feasible solution is found, only *change* moves are used until the first feasible solution is found. After that point, both types of moves are considered.

In the problem treated in [11], the exact number of employees per shift was known in advance and therefore all staffing constraints could already be fulfilled by the initial solution, making *change* moves unnecessary. In our problem, not only can the number of nurses covering a single shift vary between the minimum and optimum staffing levels, but there are also skill restrictions to take into account that prevent some nurses from covering certain assignments. This makes it hard to generate an initial solution that already contains all assignments that appear in the final solution.

For the construction of such an initial solution, we experimented with various heuristics, including random assignment of shifts and skills to nurses, a randomized greedy algorithm that assigns a randomly chosen shift and skill to the nurse with the smallest penalty and a heuristic ordering similar to the one described in [4] (with weights adapted accordingly). There was no noticeable effect on the quality of the final solution, so we used a simple random assignment algorithm that tried to avoid overstaffing for all further experiments.

To ensure the satisfaction of the hard constraints, we transformed them into soft constraints with large penalty values (3000 for H2, 1000 for H3; H1 and H4 are satisfied in all solutions). While this does not guarantee that a generated solution will be feasible, it proved sufficient to find feasible solutions after only a few steps of the search algorithm.

As an alternative heuristic, we also experimented with using Random Walk (RW) instead of MC, but this reduced the quality of the results.

As basis for our evaluation function, we used the extended model described above, without constraints S8* and $W^{S6*} = W^{S7*} = 1$, as this configuration turned out to provide the best results also for the IP model (compare Section 6).

To tune the parameters of the framework (i.e. the probability p of using TS to determine the next move, the number of non-improving moves m before a restart and the length of the tabu list), we used IRACE [12], an automated parameter-tuning framework. The tuning was performed on a subset of the late instances[2] of the INRC-II (chosen to contain representatives of various instance sizes), with a limit of 1000 iterations. This gave us optimal values of $p = 0.5$, $m = 7000$ and a tabu list length of $7*|N|$, were $|N|$ is the number of nurses. Further experimentation showed that the overall results are very robust versus small changes of these values.

6 Experimental Results

All algorithms were implemented in Java 7, and we used the IBM ILOG CPLEX solver[3], version 12.6.3, to solve the IP models. All experiments were performed on an Intel Xeon 2.33GHz PC, using a single thread. The time limit for each week was set to the time alloted by the benchmarking script[4] provided for the INRC-II (on our machine, about 1 minute for the small instances and up to 5 minutes for larger instances).

We first evaluated our models using various different parameter settings, especially regarding the additional constraints S6*[alt], S7* and S8*, on a subset of the set of hidden instances[5] published for the INRC-II as well as four randomly generated instances from the set of late instances. This instance set was selected to provide results for instances of different sizes, yet still be small enough to allow a comparison of multiple models in reasonable time.

The results can be seen on Table 1.

Model	Basic	S6*		S6*alt		S8*		S6*&S8*	LS
W^{S6*} / W^{S8*}	-/-	1/-	10/-	1/-	10/-	-/1	-/10	1/1	
n030w4_0_8-1-4-3	2610	2275	2100	2200	2095	2430	2740	2290	2400
n030w8_1_5-4-1-2-1-2-3-3	4320	2850	2610	2910	2715	3555	3680	2795	3190
n050w4_1_3-0-1-2	2980	1995	1920	2070	1980	2375	2650	2050	2510
n080w4_1_3-0-1-2	5580	4140	4025	4325	4180	5895	6615	5430	4770
n035w4_2_8-8-7-5	2375	1595	1505	1545	1500	2375	2260	1865	1795
n035w8_2_9-7-2-2-5-7-4-3	5320	3240	3375	3220	3505	5175	5335	4450	3950
n070w4_0_3-6-5-1	4560	3380	3320	3485	3235	4815	5345	4265	3765
n070w8_2_9-3-5-2-2-9-2-0	10190	6275	6175	6445	6225	10775	12695	9820	7875
n110w4_0_1-4-2-8	6085	3165	3260	3215	3210	5970	5850	4225	3945
n110w8_2_8-5-7-3-9-8-8-5	11465	5775	5655	6310	6290	10960	10400	7375	6800

Table 1 Comparison of experimental results on a subset of the instances. For variants of the extended model (all columns except 'Basic' and 'LS'), the second row denotes the weights of the additional constraints S6* (S7*) and S8*, if used.

The first column shows results using the basic model only, without additional constraints. Subsequent columns contain results for extended models, including some or all of the additional constraints at various weights. Due to the similarity in structure and purpose, we used constraints S7* wherever constraints S6* were

[2] http://mobiz.vives.be/inrc2/wp-content/uploads/2014/08/late-instances.txt
[3] http://www-03.ibm.com/software/products/en/ibmilogcpleoptistud/
[4] http://mobiz.vives.be/inrc2/?page_id=245
[5] http://mobiz.vives.be/inrc2/wp-content/uploads/2014/08/hidden-instances.txt

used and assumed $W^{S7*} = W^{S6*}$. Finally, the last column has the results for our local search heuristic, in the configuration described in Section 5.

The results of Table 1 are visualized in Figure 2. The best results were obtained using one of the two variants for constraint S6*. While the results using constraint S8* slightly improve upon those for the basic model, the impact is only minor and using both S6* and S8* produces worse results than with S6* alone. This is the case even for instances where each week is solved to optimality within the time given, indicating that the increased complexity of the model is not the reason for this.

We also tried assigning much higher weights to the additional constrains (up to 50), but this resulted in solutions even worse than for the basic model and is thus not shown here.

The results of local search, if the additional constraints are taken into consideration, are promising, but still the exact method gives better results.

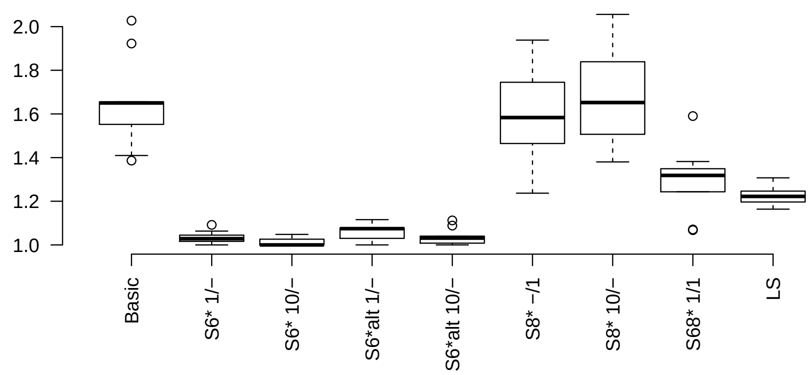

Fig. 2 Relative performance of different models and algorithms. Each result from Table 1 was scaled by the best result achieved for this instance

To reduce sample bias, we evaluated the performance of the best models over the whole set of hidden instances. The results, as well as a comparison with those of the finalists of the INRC-II, can be seen on Table 2. Figure 3 shows that all four models give similar results on average, with models with $W^{S6*} = 10$ performing slightly better.

As we can see from Table 2, the results of our best models are comparable to those of the finalists, although they couldn't improve upon the best results achieved.

Considering solution times, CPLEX was able to solve most weeks to optimality, even for the larger instances. Over the whole set of hidden instances, using the S6* 1/- model, an optimal solution could be found for 296 out of 360 weeks. For the remaining 64 weeks, the solver could not prove optimality within the given time limit. The average gap between the best solution found and the final lower bound

Instance	S6* 1/-	S6* 10/-	S6*alt 1/-	S6*alt 10/-	INRC-II Finalists Median	Best	Rank
n035w4_0_1-7-1-8	2095	1815	2005	**1795**	1756.5	1630	4.6
n035w4_0_4-2-1-6	2080	2110	2115	**2060**	2021.5	1800	4.9
n035w4_0_5-9-5-6	2085	**2055**	2070	2065	1928.5	1755	5
n035w4_0_9-8-7-7	1905	1755	1920	**1725**	1723.5	1540	4.9
n035w4_1_0-6-9-2	1770	**1725**	1810	1800	1737	1500	4.2
n035w4_2_8-6-7-1	1855	1795	**1785**	1825	1644.5	1490	5.4
n035w4_2_8-8-7-5	1595	1505	1545	**1500**	1407.5	1255	4.9
n035w4_2_9-2-2-6	2095	**2075**	2085	2085	1947.5	1705	5.3
n035w4_2_9-7-2-2	2170	2085	2200	**2025**	1970.5	1650	5.3
n035w4_2_9-9-2-1	2035	2025	**2015**	2030	1927.5	1620	5.2
n035w8_0_6-2-9-8-7-7-9-8	3530	**3370**	3690	3425	4171	3020	2.1
n035w8_1_0-8-1-6-1-7-2-0	3485	**3370**	3450	3430	4045.5	2770	3
n035w8_1_0-8-4-0-9-1-3-2	3545	3530	**3390**	3480	4019	2775	3.8
n035w8_1_1-4-4-9-3-5-3-2	**3440**	3480	3500	3580	3472.5	2805	4.7
n035w8_1_7-0-6-2-1-1-1-6	3810	3600	3910	**3425**	3548.5	2840	4.7
n035w8_2_2-1-7-1-8-7-4-2	3550	3505	3530	**3500**	4205	2910	2.6
n035w8_2_7-1-4-9-2-2-6-7	3860	**3485**	3730	3625	3699.5	2960	3.1
n035w8_2_8-8-7-5-0-0-6-9	3460	3400	3295	**3285**	3603	2815	3.1
n035w8_2_9-5-6-3-9-9-2-1	**3495**	3620	3625	3585	3659	3045	4
n035w8_2_9-7-2-2-5-7-4-3	3240	3375	**3220**	3505	3508	2715	3.2
n070w4_0_3-6-5-1	3380	3320	3485	**3235**	3151	2700	5
n070w4_0_4-9-6-7	2990	**2825**	3065	3010	2889	2430	4.1
n070w4_0_4-9-7-6	3180	**2875**	3200	3080	2948	2475	4.1
n070w4_0_8-6-0-8	3395	**3015**	3410	3075	3016	2435	4.5
n070w4_0_9-1-7-5	2945	2875	2830	**2810**	2864	2320	4.5
n070w4_1_1-3-8-8	3350	**3130**	3480	3160	3134.5	2700	4.4
n070w4_2_0-5-6-8	2960	**2870**	3100	2955	3012	2520	4
n070w4_2_3-5-8-2	3280	**2955**	3120	3020	3141.5	2615	3.8
n070w4_2_5-8-2-5	3135	2940	3030	**2915**	3005.5	2540	4.2
n070w4_2_9-5-6-5	3170	**2985**	3225	3140	3046	2615	4.1
n070w8_0_3-3-9-2-3-7-5-2	6435	**6105**	6650	6615	6222	5115	4.1
n070w8_0_9-3-0-7-2-1-1-0	6790	6665	6735	**6545**	6602	5390	4.8
n070w8_1_5-6-8-5-7-8-5-6	6815	**6295**	6910	6565	6236.5	5475	4.6
n070w8_1_9-8-9-9-2-8-1-4	**5840**	6260	5970	6105	6018.5	5100	5
n070w8_2_4-9-2-0-2-7-0-6	6600	**6125**	6390	6715	6259	5410	4
n070w8_2_5-1-3-0-8-0-5-8	**6355**	6615	6585	6785	6315	5280	5.1
n070w8_2_5-7-4-8-7-2-9-9	6925	6610	6795	**6505**	6317.5	5505	5.1
n070w8_2_6-3-0-1-8-1-5-9	6305	6390	6595	**6200**	6255	5120	4.8
n070w8_2_8-6-0-1-6-4-7-8	6455	6310	6675	**6290**	6890.5	5350	4
n070w8_2_9-3-5-2-2-9-2-0	6275	**6175**	6445	6225	6044.5	5320	4.9
n110w4_0_1-4-2-8	**3165**	3260	3215	3210	3539	2710	4
n110w4_0_1-9-3-5	3335	3420	**3255**	3460	3663	2920	4
n110w4_1_0-1-6-4	3545	3545	3445	**3385**	4030	2850	3.9
n110w4_1_0-5-8-8	**3275**	3325	3520	3450	3569.5	2820	3.9
n110w4_1_2-9-2-0	3950	**3755**	3920	3885	4092	3345	4
n110w4_1_4-8-7-2	3485	**3480**	3545	3500	3661	2805	4
n110w4_2_0-2-7-0	3730	**3580**	3740	3770	4198.5	3005	3.5
n110w4_2_5-1-3-0	**3290**	3395	3490	3470	3637.5	2925	4
n110w4_2_8-9-9-2	4050	4115	4070	**3960**	4025	3415	4.8
n110w4_2_9-8-4-9	3430	3475	3610	**3410**	3769	3135	3.6
n110w8_0_2-1-1-7-2-6-4-7	6550	**6340**	6505	6605	6596	5155	4
n110w8_0_3-2-4-9-4-1-3-7	**5765**	6045	6250	6460	6172.5	4805	4.2
n110w8_0_5-5-2-2-5-3-4-7	**5895**	5985	6095	6145	6227	4750	4
n110w8_0_7-8-7-5-9-7-8-1	**6090**	6235	6150	6165	6251.5	4855	4.4
n110w8_0_8-8-0-2-3-4-6-3	**5675**	5845	6115	6090	6146.5	4465	4
n110w8_0_8-8-2-2-3-2-0-8	6155	**6035**	6355	6295	6469	4865	4
n110w8_1_0-6-1-0-3-2-9-1	**6235**	6350	6425	6465	6514	5090	4.1
n110w8_1_4-1-3-6-8-8-1-3	**5805**	5835	6225	5820	6115.5	4315	4
n110w8_2_2-9-5-5-1-8-4-0	6325	6215	**6170**	6520	6222.5	4770	4.4
n110w8_2_8-5-7-3-9-8-8-5	5775	**5655**	6310	6290	5809	4360	4.1

Table 2 Results for S6* 1/-, S6* 10/-, S6*alt 1/- and S6*alt 10/-, compared with the median and the best result achieved by the competition finalists. Rank is calculated as the average rank (out of 8, seven finalists plus our results) over 10 runs of each instance for the configuration S6* 10/-. Results for the other configurations are similar.

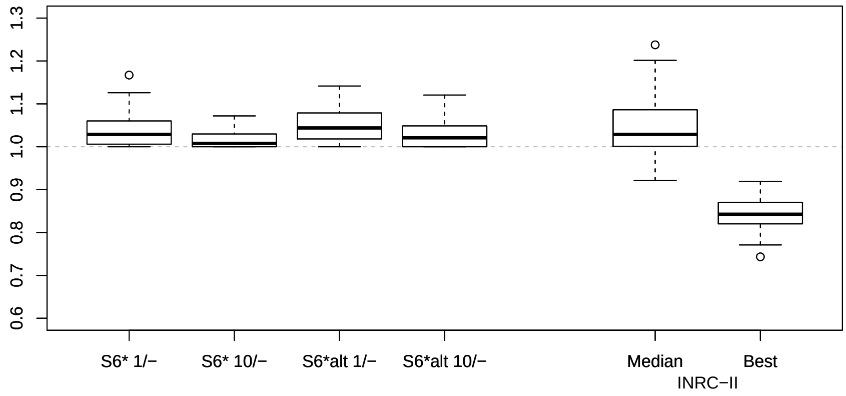

Fig. 3 Relative performance of S6* 1/-, S6* 10/-, S6*alt 1/- and S6*alt 10/- and comparison with the results of the INRC-II finalists. The baseline value for each instance is the best result achieved by our solvers.

for these weeks was only 1.20%, indicating that substantial improvements are not to be expected even with much longer running times.

7 Conclusions

In this paper, we have proposed and evaluated different extensions of standard IP formulations for nurse rostering problems in order to deal with multi-stage settings, as described for the INRC-II.

We have shown that our extensions significantly improve upon the results of the basic model and achieve competitive results compared to the finalists in the competition.

The fact that our model could be solved to (near) optimality in most cases, even under the strict time limits imposed by the challenge, indicates that major improvements cannot be expected from varying solution techniques alone. Instead, future research should be focused on further modifications of the model to distribute the penalties more equally between weeks and avoid blocking options for later weeks. Techniques that try to predict the requirements of yet unknown weeks or distinguish between nurses of different skill sets and contracts could result in even better models.

Acknowledgements This work was supported by the Austrian Science Fund (FWF): P24814-N23.

References

1. den Bergh, J.V., Beliën, J., Bruecker, P.D., Demeulemeester, E., Boeck, L.D.: Personnel scheduling: A literature review. European Journal of Operational Research **226**(3), 367–385 (2013). URL http://www.sciencedirect.com/science/article/pii/S0377221712008776
2. Brucker, P., Qu, R., Burke, E.: Personnel scheduling: Models and complexity. European Journal of Operational Research **210**(3), 467–473 (2011)
3. Burke, E.K., Causmaecker, P.D., Vanden Berghe, G., Landeghem, H.V.: The state of the art of nurse rostering. J. Scheduling **7**(6), 441–499 (2004). URL http://dx.doi.org/10.1023/B:JOSH.0000046076.75950.0b
4. Burke, E.K., Curtois, T., Post, G., Qu, R., Veltman, B.: A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem. European Journal of Operational Research **188**(2), 330–341 (2008). URL http://www.sciencedirect.com/science/article/pii/S0377221707004390
5. Burke, E.K., Li, J., Qu, R.: A hybrid model of integer programming and variable neighbourhood search for highly-constrained nurse rostering problems. European Journal of Operational Research **203**(2), 484–493 (2010). URL http://www.sciencedirect.com/science/article/pii/S0377221709005396
6. Ceschia, S., Thanh, N.D.T., Causmaecker, P.D., Haspeslagh, S., Schaerf, A.: Second international nurse rostering competition (INRC-II) - problem description and rules -. CoRR **abs/1501.04177** (2015). URL http://arxiv.org/abs/1501.04177
7. Ernst, A., Jiang, H., Krishnamoorthy, M., Sier, D.: Staff scheduling and rostering: A review of applications, methods and models. European Journal of Operational Research **153**(1), 3–27 (2004). URL http://www.sciencedirect.com/science/article/pii/S037722170300095X. Timetabling and Rostering
8. Glass, C.A., Knight, R.A.: The nurse rostering problem: A critical appraisal of the problem structure. European Journal of Operational Research **202**(2), 379–389 (2010). DOI 10.1016/j.ejor.2009.05.046. URL http://www.sciencedirect.com/science/article/pii/S0377221709003968
9. Glover, F., Laguna, M.: Tabu search. Kluwer Academic Publishers (1997)
10. Minton, S., Johnston, M.D., Philips, A.B., Laird, P.: Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems. Artificial Intelligence **58**, 161–205 (1992)
11. Musliu, N.: Heuristic methods for automatic rotating workforce scheduling. International Journal of Computational Intelligence Research **2**(4), 309–326 (2006)
12. nez, M.L.I., Dubois-Lacoste, J., Stützle, T., Birattari, M.: The irace package, iterated race for automatic algorithm configuration. Tech. Rep. TR/IRIDIA/2011-004, IRIDIA, Université Libre de Bruxelles, Belgium (2011). URL http://iridia.ulb.ac.be/IridiaTrSeries/IridiaTr2011-004.pdf
13. Römer, M.: A multi-commodity flow-based mixed-integer linear programming formulation for nurse rostering problems (abstract). OR2015-International Conference on Operations Research, Vienna (2015)
14. Salassa, F., Vanden Berghe, G.: A stepping horizon view on nurse rostering. In: Proceedings of the 9th international conference on the practice and theory of automated timetabling., pp. 161–173 (2012)
15. Santos, D., Fernandes, P., Cardoso, H.L., Oliveira, E.: A weighted constraint optimization approach to the nurse scheduling problem. In: Computational Science and Engineering (CSE), 2015 IEEE 18th International Conference on, pp. 233–239 (2015). DOI 10.1109/CSE.2015.46
16. Santos, H.G., Toffolo, T.A.M., Gomes, R.A.M., Ribas, S.: Integer programming techniques for the nurse rostering problem. Annals of Operations Research pp. 1–27 (2014). URL http://dx.doi.org/10.1007/s10479-014-1594-6
17. Smet, P., Salassa, F., Vanden Berghe, G.: Local and global constraint consistency in personnel rostering. Tech. rep., KU Leuven (2016)