Quantifying and enforcing robustness in staff rostering

Toni I. Wickert[1,2], Pieter Smet[1], Greet Vanden Berghe[1]

[1]*KU Leuven, Department of Computer Science, CODeS*
[2]*Institute of Informatics, Federal University of Rio Grande do Sul (UFRGS)*

Abstract

Employee absences are inevitable in practice due to illness, heavy workloads or accidents. These unforeseen events result in the disruption of employee shift rosters, which are then typically repaired using re-rostering methods. Despite their widespread use, repair methods often require last-minute changes to rosters, negatively affecting employees' personal lives. Robust rosters are thus crucial when it comes to minimizing the negative impact of these unforeseen events. The academic literature, however, currently lacks a metric capable of measuring the robustness of staff rosters. This study introduces two complementary approaches for quantifying roster robustness without the use of simulation. The first metric enables an a priori estimation of how well unexpected employee absences can be accommodated, while the second metric approximates the costs incurred by repair methods. An integer programming-based methodology is proposed for generating robust rosters in a controlled manner. A computational study using public instances based on real-world scenarios demonstrates the effectiveness of the proposed methodology. A significant reduction of costs is observed when enforcing an appropriate level of robustness, compared against when robustness is ignored.

Keywords: Robustness, Integer programming, Personnel rostering

1 Introduction

Employee absenteeism is the result of a variety of causes such as illness, injuries, childcare and bullying (Forbes, 2013). A recent study reports that Belgian organizations suffer from a short-term (less than one month) absenteeism rate of 2.64% (SD Worx, 2016). Other studies report absence rates between 3% and 6% in Europe and 2.8% in the United States (Edwards and Greasley, 2010; Bureau of Labor Statistics, 2017). The direct and indirect effects of employee absenteeism are significant (Kocakulah et al., 2016). The Society for Human Resource Management (2016) reports that 67% of the questioned organizations perceived absences to have a moderate to significant impact on productivity, while 62% said absences disrupted the work of others, and 51% reported them to increase stress. Moreover, employees with supervisory responsibility spent on average 3.3 hours per week responding to absences by searching for replacements, adjusting workflow and providing additional training.

Unforeseen absences typically have a significant impact on staff rosters, quickly rendering them sub-optimal or even infeasible (Drake, 2014). The academic literature has introduced several re-rostering algorithms for repairing roster infeasibility caused by disruptions (Moz and Pato, 2003; Maenhout and Vanhoucke, 2011; Bäumelt et al., 2016; Wickert et al., 2019). Although these procedures have proven effective, they always negatively impact employees' personal lives due to the inevitable last-minute changes to the roster they demand (Williams et al., 2017). Flexible rosters, that is, robust rosters, are essential with respect to mitigating such negative effects by being less sensitive to disruptions (Bertsimas and Sim, 2004). In staff rostering, common approaches to improve the robustness of a roster include maximizing employee substitutability, designing effective hiring and overtime policies, using reserve shifts (Ingels and Maenhout, 2017b), and efficiently employing available staff (Guo et al., 2014).

Despite this diversity of methods for enforcing robustness, the academic literature currently lacks an exact and unanimously agreed upon manner by which roster robustness can be calculated. The main contributions of this paper are two approaches for quantifying the robustness of a roster, taking into account the complex skill structures which typically characterize rostering problems. The first metric serves as an abstract indicator of how well two types of buffers (capacity and reserve shift buffers) in a roster can help accommodate disruptions before becoming infeasible. The second metric approximates the expected cost associated with repairing disruptions using these buffers. While there are a number of ways in which robustness could be formalized and quantified, the metrics proposed in this paper are presented for a general problem model and can thus be widely deployed across personnel rostering problems.

Following the proposed quantification of roster robustness, an integer programming-based methodology is introduced for generating rosters which guarantee a minimum level of robustness. The integer programming formulation is designed such that the generated rosters remain feasible for every possible realization of the reserve shifts. In other words, reserve shifts may be either removed from the roster or converted into a working shift without violating any employee's contractual constraints. The proposed methodology is validated using a three-stage procedure which simulates employee absences based on a given probability distribution and repairs disruptions using a re-rostering algorithm. A computational study involving nurses in a hospital ward demonstrates the trade-off between roster robustness and the operational costs incurred by repairing disruptions. Finding the optimal trade-off point allows for significant cost reductions to be realized by increasing the robustness of rosters.

The remainder of the paper is organized as follows. Section 2 reviews literature related to robust staff rostering. Section 3 defines the staff rostering problem, while Section 4 introduces the first metric which quantifies robustness for rosters with both single- and multi-skilled employees. Section 5 details the methodology employed to generate robust rosters and to validate the proposed metric. Section 6 introduces the second metric which approximates the expected re-rostering cost: the cost incurred by repairing disruptions. Section 7 discusses the results of a computational study. Finally, Section 8 concludes the paper and identifies future research possibilities.

2 Related work

Van den Bergh et al. (2013) identified three main categories of uncertainty in staff rostering: *(i)* uncertainty of capacity in which actual available manpower may not be the same as the planned, *(ii)* uncertainty of demand in which the expected workload is not deterministic and *(iii)* uncertainty of arrival in which the arrival pattern of the workload is not deterministic. Uncertainty of capacity is characteristic of every organization working with human resources given the unavoidable occurrence of employee absenteeism. Uncertainty of demand may occur in hospitals since the workload depends on patient admissions, however, most hospitals operate under the assumption of fixed (seasonal) demand. Uncertainty of arrival is strongly related to uncertainty of demand and has an impact only if individual tasks require scheduling.

Two general approaches currently exist for introducing robustness in staff rosters: those focusing on horizontal robustness and those focusing on vertical robustness. Horizontal robustness may be obtained by extending shift durations or by permitting overtime to compensate for employee shortages (Ingels and Maenhout, 2018). Vertical robustness is introduced through different types of buffers which manage uncertainty of demand and capacity through the use of surplus resources. Capacity buffers involve assigning more resources than originally required by the nominal problem by, for example, assigning employees up to a preferred staffing level rather than a minimum level (Ingels and Maenhout, 2017b). A similar strategy is the use of reserve shifts, where employees are assigned to special reserve shifts which, if necessary, are converted into working shifts to cover disruptions (Ingels and Maenhout, 2015). Alternative strategies include introducing replaceability among employees such that absences may be covered without affecting roster feasibility (Ingels and Maenhout, 2017a) or by enabling overlap between working shifts (Lusby et al., 2012).

The different approaches introduced in the academic literature typically have one or more parameters responsible for indirectly controlling the robustness of a roster. For example, the number of reserve shifts required in a solution determines how robust a roster will be against uncertainty of capacity and demand (Ingels and Maenhout, 2015). The best value of this parameter is set based on an extensive empirical study of different strategies specific for the considered experimental setting. Similarly, Lusby et al. (2012) control the level of robustness by increasing the expected workload and by adding delays to the start time of tasks. However, there is no direct quantifiable relation between these parameters and the degree to which solutions are immune to unforeseen events.

Although there is a history of strategies for enforcing robustness in existing literature, these approaches all lack a formal metric for quantifying the robustness of a roster. The present research remedies this shortcoming by introducing two complementary ways of quantifying the robustness of staff rosters a priori using analytical functions, rather than relying on the outcome of simulations to evaluate robustness. Objectively quantifying robustness has significant importance as it enables organizations to estimate how immune their rosters are to unexpected employee shortages as well as the impact of such disruptions.

3 The staff rostering problem

The staff rostering problem considered in this paper is based on a general problem definition from the literature which captures various problem characteristics from practice (Ceschia et al., 2019). The goal is to find an assignment of shifts to employees subject to a variety of personal and organizational constraints. The organization expects the minimum staffing requirements to be met while assigning at most one shift per day to each employee. The remaining constraints are related to the contracts and personal preferences of the employees. For example, there are forbidden shift successions which prevent certain shift combinations from being assigned on two consecutive days. Other constraints include shift or day off requests, skill requirements, maximum number of consecutive working days and maximum number of consecutive night shifts worked. Employees are allowed to work overtime by working more days than the maximum contracted. Undertime is not permitted. Note that in order to correctly evaluate these constraints for a given scheduling period, data from the preceding scheduling period must be taken into account to avoid constraint violations at the beginning. To avoid issues regarding infeasibility, constraints related to the staffing requirements and overtime are relaxed to soft constraints. The objective function is a weighted sum of the soft constraint violations along with employee wage costs and overtime expenses. Table 1 provides an overview of all hard and soft constraints in the considered problem. For a detailed discussion concerning these constraints we refer interested readers to Ceschia et al. (2019) and Smet et al. (2014).

Table 1: Overview of hard and soft constraints

Hard constraints
An employee can be assigned to at most one shift per day
A shift requiring employees with a given skill must always be fulfilled by an employee having that skill
Forbidden shift succession
Minimum number of days worked per employee
Maximum number of consecutive working days
Minimum number of consecutive working days
Maximum number of consecutive night shift assignments
Shift or day off requests
Soft constraints
Minimum staffing requirements for each day, shift and skill
Maximum number of days worked per employee

Table 2 defines the sets, parameters and decision variables used in an integer programming formulation of the problem. The integer programming problem is as follows:

Table 2: Indices, sets and variables used in the integer programming formulation.

Symbol	Definition
Sets and parameters	
$n \in N$	set of all employees, indexed by n
$d \in D$	set of days in the planning horizon, indexed by d
$s \in S$	set of all shifts, including the reserve shift, indexed by s
$S^w \subset S$	subset of all working shifts which does not include the reserve shift
s_n	index of the night shift in set S
$k \in K$	set of all skills, indexed by k
$\tilde{K}_n \subset K$	subset of skills for which employee n is not qualified
m_{dsk}	minimum number of employees required on day d for shift s with skill k
$(s_1, s_2) \in \tilde{S}$	set of shift pairs which define forbidden shift successions
$(n, d, s) \in U$	set of tuples which define forbidden assignments
β_n^1	maximum number of consecutive working days of employee n
β_n^2	maximum number of consecutive night shifts worked of employee n
β_n^3	minimum total number of working days of employee n
β_n^5	maximum total number of working days of employee n
$\omega_n^1 \in \mathbb{N}_{\geq 0}$	wage cost of a working day of employee n
$\omega_n^5 \in \mathbb{N}_{\geq 0}$	overtime cost of employee n per additional day worked over the allowed maximum
$\omega^6 \in \mathbb{N}_{\geq 0}$	understaffing cost
Decision variables	
$x_{ndsk} \in \{0, 1\}$	1 if employee n is assigned to shift s on day d with skill k, 0 otherwise
$v_n^5 \in \mathbb{N}_{\geq 0}$	number of days worked over the maximum contracted for employee n
$v_{dsk}^6 \in \mathbb{N}_{\geq 0}$	number of employees under the minimum required for day d, shift s and skill k

$$min \sum_{n\in N}\sum_{d\in D}\sum_{s\in S^w}\sum_{k\in K} x_{ndsk}\omega_n^1 + \sum_{n\in N} v_n^5\omega_n^5 + \sum_{d\in D}\sum_{s\in S^w}\sum_{k\in K} v_{dsk}^6\omega^6 \tag{1}$$

$$s.t. \sum_{s\in S}\sum_{k\in K} x_{ndsk} \leq 1 \qquad \forall n\in N, d\in D \tag{2}$$

$$\sum_{n\in N} x_{ndsk} + v_{dsk}^6 \geq m_{dsk} \qquad \forall d\in D, s\in S^w, k\in K \tag{3}$$

$$\sum_{k\in K}(x_{nds_1k} + x_{n(d+1)s_2k}) \leq 1 \qquad \forall n\in N, d\in D\setminus\{|D|\}, (s_1,s_2)\in \tilde{S} \tag{4}$$

$$\sum_{d\in D}\sum_{s\in S} x_{ndsk} = 0 \qquad \forall n\in N, k\in \tilde{K}_n \tag{5}$$

$$\sum_{d'=d}^{\beta_n^1+d}\sum_{s\in S}\sum_{k\in K} x_{nd'sk} \leq \beta_n^1 \qquad \forall n\in N, d\in\{1,\ldots,|D|-\beta_n^1\} \tag{6}$$

$$\sum_{d'=d}^{\beta_n^2+d}\sum_{k\in K} x_{nd's_nk} \leq \beta_n^2 \qquad \forall n\in N, d\in\{1,\ldots,|D|-\beta_n^2\} \tag{7}$$

$$\sum_{k\in K} x_{ndsk} = 0 \qquad \forall (n,d,s)\in U \tag{8}$$

$$\sum_{d\in D}\sum_{s\in S^w}\sum_{k\in K} x_{ndsk} \geq \beta_n^3 \qquad \forall n\in N \tag{9}$$

$$\sum_{d\in D}\sum_{s\in S^w}\sum_{k\in K} x_{ndsk} - v_n^5 \leq \beta_n^5 \qquad \forall n\in N \tag{10}$$

Objective function (1) minimizes a weighted sum of the costs associated with employee wages, overtime and understaffing. Constraints (2) assign at most one shift per day for each employee. Constraints (3) require a minimum number of employees on each day, shift and skill. Constraints (4) enforce the forbidden shift succession restrictions. Constraints (5) ensure an employee can only be assigned to a shift and skill if they have the required skill. Constraints (6) limit the maximum number of consecutive days worked. Constraints (7) ensure the maximum number of consecutive assignments to night shifts is not exceeded. Constraints (8) ensure undesired shifts or days are not assigned. Constraints (9) and (10) limit the minimum and maximum number of assignments in the planning horizon, respectively.

4 Measuring robustness

The primary objective of measuring robustness is to estimate how efficiently a given roster can accommodate disruptions before becoming infeasible. This section begins by offering a definition of robustness before subsequently introducing the first metric for quantifying robustness in staff rosters with single- and multi-skilled employees. Different variations of this metric are introduced to measure robustness at different levels of granularity: roster-wide, per day, per shift or per skill.

4.1 Definition of robustness

A roster is considered robust whenever it is constructed in such a way that disruptions due to uncertainty of capacity can be repaired without leading to infeasibility. This paper considers two strategies for obtaining this type of robustness: *(i)* assigning more employees than the minimum required (capacity buffers) and *(ii)* assigning employees to reserve shifts. The first strategy does not require absent employees to be replaced given that more than the minimum are already rostered for a given shift. The second strategy provides greater flexibility, as employees in reserve shifts may be deployed to replace employees in any working shift. Note that a reserve shift may be converted into any of the working shifts without violating the employee's contractual constraints. This is not a trivial assumption and requires a dedicated solution approach to generate such rosters, as will be discussed throughout Section 5.

4.2 Single-skilled employees

Rosters are usually constructed respecting a constraint which enforces a minimum number of employees required for each day and shift. The proposed metric is expressed relative to this minimum requirement: 0 indicates that none of the employees assigned to working shifts can be replaced and 1 indicates sufficient excess staff to substitute all employees assigned to working shifts. Values higher than 1 may be obtained when a disruption is covered by multiple employees. Table 3 details the notation used in the definition of the robustness measure for problems with single-skilled employees.

Table 3: Symbols and definitions for problems with single-skilled employees

Symbol	Definition
s'	index of the reserve shift in set S
$\hat{x}_{nds} \in \{0,1\}$	equals 1 if employee n is assigned to working shift s on day d, 0 otherwise
$\hat{x}_{nds'} \in \{0,1\}$	equals 1 if employee n is assigned to reserve shift s' on day d, 0 otherwise
m_{ds}	minimum number of employees required on day d, shift s
α_d	weight of robustness on day d when computing complete roster robustness
$\hat{r}_d \in \mathbb{R}_{\geq 0}$	robustness on day d from reserve shift buffers
$\hat{r} \in \mathbb{R}_{\geq 0}$	reserve shift-robustness for the complete roster
$\hat{o}_{ds} \in \mathbb{N}_{\geq 0}$	robustness on day d and shift s from capacity buffers
$\hat{o}_d \in \mathbb{R}_{\geq 0}$	capacity buffer-robustness on day d
$\hat{o} \in \mathbb{R}_{\geq 0}$	capacity buffer-robustness for the complete roster
$rob_d \in \mathbb{R}_{\geq 0}$	robustness on day d
$rob \in \mathbb{R}_{\geq 0}$	robustness of the complete roster

Equation (11) defines the robustness on day d induced by reserve shifts. The numerator equals the number of employees assigned to reserve shifts on day d, while the denominator equals the total minimum demand summed over all working shifts. As the minimum demand is, de facto, a hard constraint, $\hat{r}_d$ expresses how many employees are assigned to reserve shifts relative to the number of employees assigned to regular working shifts.

$$\hat{r}_d = \frac{\sum_{n\in N} \hat{x}_{nds'}}{\sum_{s\in S^w} m_{ds}} \tag{11}$$

Equation (12) generalizes the daily reserve shift-robustness calculated by Equation (11) by summing $\hat{r}_d$ for each $d \in D$ multiplied by a weighting factor α_d. This allows information concerning the distribution of absences across different days of the planning horizon to be included in the metric. If this information is not known, one can assume the probability of an absence occurring is uniformly distributed over all days in the planning horizon and set $\alpha_d = 1/|D|$.

$$\hat{r} = \sum_{d\in D} \alpha_d \hat{r}_d \tag{12}$$

Robustness obtained by capacity buffers may be quantified per day and per shift. Equation (13) determines, for day d and working shift s, the number of employees assigned more than the minimum required, again expressed relative to the minimum demand. From a robustness point-of-view, the higher this number, the more unexpected absences on this day and shift may be covered using the surplus employees.

$$\hat{o}_{ds} = \frac{\sum_{n\in N} \hat{x}_{nds} - m_{ds}}{m_{ds}} \tag{13}$$

This capacity buffer-robustness may be averaged over all working shifts for a given day to obtain a more general calculation of this type of robustness. Equation (14) calculates this averaged robustness for day d.

$$\hat{o}_d = \frac{\sum_{s\in S^w} \hat{o}_{ds}}{|S^w|} \tag{14}$$

Similar to how Equation (12) generalizes daily reserve shift-robustness, $\hat{o}_d$ may be summed over all days in D with a weighting factor α_d to calculate robustness of the complete roster. This results in

Equation (15).

$$\hat{o} = \sum_{d \in D} \alpha_d \hat{o}_d \tag{15}$$

We assume that both reserve shifts and capacity buffers contribute equally to a roster's overall robustness. The complete roster robustness *rob* is calculated by combining $\hat{r}$ and $\hat{o}$. While Equation (16) defines this value as the sum of the two types of robustness, alternative operators such as $\max(\hat{r}, \hat{o})$ or $\hat{r} \times \hat{o}$ may also be considered, albeit at the expense of potentially larger compensation effects.

$$rob = \hat{r} + \hat{o} \tag{16}$$

Both types of robustness, $\hat{r}_d$ and $\hat{o}_d$, may also be combined per day resulting in Equation (17).

$$rob_d = \hat{r}_d + \hat{o}_d \tag{17}$$

4.3 Multi-skilled employees

Skills represent an additional challenge when rostering an organization's staff which cannot be ignored since employees are seldom qualified for all tasks (De Bruecker et al., 2015). In hospitals, for example, a regular nurse may be qualified for medical tasks for which a caregiver may not have had the necessary training. In such situations, caregivers cannot assume the work of a regular nurse. However, more complex situations may occur when, for example, there exists a hierarchy of qualifications or an arbitrary skill structure in which each employee has an individual subset of skills. The robustness quantifications introduced in Section 4.2 will now be extended for multi-skilled employees where each individual employee may have one or more skills. Table 4 introduces the additional notation that is required.

Table 4: Additional symbols and definitions for problems with multi-skilled employees

Symbol	Definition
$\hat{x}_{ndsk} \in \{0,1\}$	1 if employee n is assigned to working shift s on day d with skill k, 0 otherwise
$\hat{x}_{nds'k} \in \{0,1\}$	1 if employee n is assigned to reserve shift s' on day d with skill k, 0 otherwise
m_{dsk}	minimum number of employees required in shift s on day d with skill k
$\hat{r}_{dk} \in \mathbb{R}_{\geq 0}$	robustness on day d and skill k from reserve shift buffers
$\hat{o}_{dsk} \in \mathbb{N}_{\geq 0}$	robustness on day d and shift s for skill k from capacity buffers
$rob_{dk} \in \mathbb{R}_{\geq 0}$	robustness on day d for skill k

We first consider robustness obtained from reserve shift buffers. Note that employees with more than one skill are counted only once when they are assigned to reserve shifts. The re-rostering procedure, however, is not limited to use these employees only for the skill they were counted for, allowing them to be assigned to any shift for which they are qualified. Equation (18) calculates the robustness for day d and skill k. Note that this is almost identical to the single-skilled case except for the skill index.

$$\hat{r}_{dk} = \frac{\sum_{n \in N} \hat{x}_{nds'k}}{\sum_{s \in S^w} m_{dsk}} \tag{18}$$

Equation (19) calculates the daily reserve shift-robustness for day d by averaging $\hat{r}_{dk}$ over all skills. As with previous generalizations, this step makes the quantification less accurate due to compensation which may occur among different skills.

$$\hat{r}_d = \frac{\sum_{k \in K} \hat{r}_{dk}}{|K|} \tag{19}$$

Daily reserve shift-robustness is summed over all days using weights α_d to calculate the reserve shift-robustness of the complete roster, as shown in Equation (20).

$$\hat{r} = \sum_{d \in D} \alpha_d \hat{r}_d \tag{20}$$

The capacity buffer-robustness on day d, shift s and skill k is calculated by Equation (21).

$$\hat{o}_{dsk} = \frac{\sum_{n\in N}\hat{x}_{ndsk} - m_{dsk}}{m_{dsk}} \tag{21}$$

As with the previous equations, $\hat{o}_{dsk}$ can be averaged over all working shifts, skills and finally summed over all days resulting in aggregated, albeit less accurate, robustness quantification, as shown in Equations (22) - (24).

$$\hat{o}_{dk} = \frac{\sum_{s\in S^w}\hat{o}_{dsk}}{|S^w|} \tag{22}$$

$$\hat{o}_{d} = \frac{\sum_{k\in K}\hat{o}_{dk}}{|K|} \tag{23}$$

$$\hat{o} = \sum_{d\in D}\alpha_d\hat{o}_d \tag{24}$$

Finally, Equation (25) calculates the complete roster robustness for problems with multi-skilled employees by summing $\hat{r}$ and $\hat{o}$.

$$rob = \hat{r} + \hat{o} \tag{25}$$

Reserve shift- and capacity buffer-robustness may also be combined per day or per day and skill, resulting in Equations (26) and (27).

$$rob_d = \hat{r}_d + \hat{o}_d \tag{26}$$

$$rob_{dk} = \hat{r}_{dk} + \hat{o}_{dk} \tag{27}$$

4.4 Numerical example

Table 5 presents an example which considers a hospital setting with ten multi-skilled nurses and a planning horizon of seven days. The shifts are *early* (E), *late* (L) and *night* (N), and there are two skills: *head nurse* (H) and *nurse* (N). For each shift, at least one *head nurse* and one *nurse* are required. Nurses 1–5 are qualified for both *head nurse* and *nurse* skills, whereas Nurses 6–10 only have the *nurse* skill. This means that Nurses 6–10 may be replaced with Nurses 1–5, while the opposite is not true.

On Monday, $\hat{r}_d = 0.33$ since only Nurses 5 and 9 are assigned to a reserve shift. Although Nurse 5 is qualified for both the *head nurse* and *nurse* skills, they are only counted towards the highest qualification (in this case *head nurse*). Therefore, in accordance with Equation (18), $\hat{r}_{dH} = 0.33$ and $\hat{r}_{dN} = 0.33$ for the skills *head nurse* and *nurse*, respectively. Specifically, for the skill *head nurse*, $\hat{r}_{dH} = \frac{1}{3}$, where the numerator is the number of reserve shifts for the skill *head nurse* and the denominator is the sum of the required *head nurses* over all shifts. Equation (19) determines the average reserve shift robustness $\hat{r}_d = \frac{0.33+0.33}{2}$, where the numerator is the sum of the robustness considering *head nurses* and *nurses*, while the denominator is the number of skills.

The robustness on Wednesday is calculated by applying Equations (22) and (23). Equation (22) results in $\hat{o}_{dH} = \frac{2}{3}$ and $\hat{o}_{dN} = \frac{2}{3}$, where the numerator is the number of nurses in capacity buffers for skills *head nurses* and *nurses* and the denominator is the number of shifts. The next row provides the results of Equation (23), $\hat{o}_d = \frac{0.67+0.67}{2}$, where the numerator is the robustness considering capacity buffers and the denominator is the number of skills.

The last column in the example (Sunday), has both types of robustness: capacity buffers and reserve shifts. Nurse 8 and 10 are assigned to reserve shifts, resulting in $\hat{r}_d = 0.33$. Nurse 9 is scheduled in excess of the minimum required for the skill *nurse* on the *night* shift, thus resulting in the presence of a capacity buffer. Variable $\hat{o}_d$ subsequently assumes a value of 0.17. The total robustness on Sunday therefore is $rob_d = 0.33 + 0.17 = 0.50$. The general roster robustness is calculated by multiplying the daily robustness values by α_d, resulting in a value of $rob = 0.3643$.

Table 5: Example calculating roster robustness considering multi-skilled employees.

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
N01 [H, N]	E[H]	E[H]	E[H]	N[H]	N[H]	N[H]	-
N02 [H, N]	L[H]	L[H]	L[H]	-	**R**	-	E[H]
N03 [H, N]	N[H]	N[H]	N[H]	-	**R**	-	L[N]
N04 [H, N]	-	**R**	*N[H]*	L[H]	L[H]	L[H]	L[H]
N05 [H, N]	**R**	-	*E[H]*	E[H]	E[H]	E[H]	N[H]
N06 [N]	E[N]	E[N]	E[N]	-	**R**	-	N[N]
N07 [N]	L[N]	L[N]	L[N]	L[N]	L[N]	L[N]	E[N]
N08 [N]	N[N]	N[N]	N[N]	-	**R**	-	**R**
N09 [N]	**R**	-	*E[N]*	E[N]	E[N]	E[N]	*N[N]*
N10 [N]	-	**R**	*N[N]*	N[N]	N[N]	N[N]	**R**
$\hat{r}_{dH}$	0.33	0.33	0.00	0.00	0.67	0.00	0.00
$\hat{r}_{dN}$	0.33	0.33	0.00	0.00	0.67	0.00	0.67
$\hat{r}_{d}$	0.33	0.33	0.00	0.00	0.67	0.00	0.33
$\hat{o}_{dH}$	0.00	0.00	0.67	0.00	0.00	0.00	0.00
$\hat{o}_{dN}$	0.00	0.00	0.67	0.00	0.00	0.00	0.33
$\hat{o}_{d}$	0.00	0.00	0.67	0.00	0.00	0.00	0.17
rob_d	0.33	0.33	0.67	0.00	0.67	0.00	0.50
α_d	0.29	0.21	0.17	0.17	0.12	0.02	0.01
rob	0.36						

5 Generating and validating robust rosters

With a standardized means of calculating a given roster's robustness, it is possible to formulate additional constraints which enforce a minimum level of robustness when generating rosters. Constraints based on the proposed robustness quantification are added to the integer programming formulation presented in Section 3. For example, if a minimum capacity buffer robustness ρ_d is required on each day of the planning horizon in a multi-skilled environment, Constraints (28) are added to model (1) - (10).

$$\frac{\sum_{s\in S^w}\sum_{k\in K}\left(\frac{\sum_{n\in N} x_{ndsk}-m_{dsk}}{m_{dsk}}\right)}{|S^w||K|} + v^r_d \geq \rho_d \qquad \forall d \in D \qquad (28)$$

$$v^r_d \geq 0 \qquad \forall d \in D \qquad (29)$$

To avoid issues regarding infeasibility which may arise when there is only a limited number of employees available, these additional constraints are modeled as soft constraints. In the example above, an integer penalty variable v^r_d is added to the problem's objective function, resulting in objective function (30). A weight ω^r is associated with not meeting the required minimum robustness level.

$$min\ (1) + \sum_{d\in D} \omega^r v^r_d \qquad (30)$$

To validate a roster's ability to cope with uncertainty concerning capacity, a three-phase procedure similar to the approaches of Abdelghany et al. (2008) and Ingels and Maenhout (2015) is employed. First, a baseline roster is generated using the aforementioned methodology. Note that, in addition to minimizing soft constraint violations, objective function (1) also includes employee wages. Introducing robustness in this phase thus increases the objective value. By assigning employee n to a reserve shift, an additional wage cost ω^7_n is incurred, regardless of whether they are called in or not. The generated rosters ensure that every reserve shift can be converted into any working shift by carefully modeling the employee's contractual and personal constraints.

The second step involves simulating employee absences which lead to roster disruptions. These disruptions are manifested as employees who become unavailable on one (or more) days in the scheduling period, thereby possibly introducing infeasibility by violating their contractual constraints, and leading to violations of the demand requirements.

The third step applies a re-rostering method which attempts to restore the solution's feasibility in terms of demand. The integer programming model introduced below is employed in this re-rostering phase. Solving this optimization problem converts reserve shifts into working shifts in addition to other modifications, such as changing shift assignments or converting a day off into a working day. The main objective is now to minimize the overall cost of the applied re-rostering operations. Table 6 shows an overview of the parameters and decision variables employed in the re-rostering integer programming formulation.

Table 6: Additional indices, sets and variables for the re-rostering formulation.

Symbol	Definition
Sets and parameters	
$\hat{N}$	set of absent employees
$\hat{c}_{nd} \in \{0,1\}$	1 if employee n is absent on day d, and 0 otherwise
c_{ndsk}	1 if employee n is assigned to shift s on day d with skill k in the original roster, and 0 otherwise
$\omega_n^2 \in \mathbb{N}_{\geq 0}$	cost for employee n to change shift (excluding the reserve shift)
$\omega_n^3 \in \mathbb{N}_{\geq 0}$	cost for employee n to convert a reserve shift into a working shift
$\omega_n^4 \in \mathbb{N}_{\geq 0}$	cost for employee n to convert a day off into a working shift or vice versa
Decision variables	
$y'_{nds} \in \{0,1\}$	1 if employee n works on the original schedule or on the new roster, 0 otherwise
$y''_{nds} \in \{0,1\}$	auxiliary variable to calculate the number of changes compared to the original roster
$y'''_{nds} \in \{0,1\}$	auxiliary variable to calculate the number of changes compared to the original roster
$v_{nd}^2 \in \mathbb{N}_{\geq 0}$	number of shift changes compared to the original roster excluding the reserve shifts for employee n on day d
$v_{nd}^3 \in \mathbb{N}_{\geq 0}$	number of changes from the reserve shift to any working shift for employee n on day d
$v_{nd}^4 \in \mathbb{N}_{\geq 0}$	number of times a day off is replaced with a working shift and vice versa

$$min\ (1) + \sum_{n \in N} \sum_{d \in D} \sum_{i \in \{2,..,4\}} v^i_{nd} \omega^i_n \tag{31}$$

$$s.t.\ \hat{c}_{nd} + \sum_{s \in S} \sum_{k \in K} x_{ndsk} \leq 1 \qquad \forall n \in N, d \in D \tag{32}$$

$$\sum_{k \in K} (c_{ndsk} + x_{ndsk}) \leq 2y'_{nds} \qquad \forall n \in N \setminus \hat{N}, d \in D, s \in S \tag{33}$$

$$\sum_{k \in K} (c_{ndsk} + x_{ndsk}) + y''_{nds} \geq 2y'_{nds} \qquad \forall n \in N \setminus \hat{N}, d \in D, s \in S \tag{34}$$

$$\sum_{s \in S} y''_{nds} - 2y'''_{nd} \leq 0 \qquad \forall n \in N \setminus \hat{N}, d \in D \tag{35}$$

$$\sum_{s \in S^w} \sum_{k \in K} (c_{ndsk} + x_{ndsk}) - 1 - v^2_{nd} \leq 1 - y'''_{nd} \qquad \forall n \in N \setminus \hat{N}, d \in D \tag{36}$$

$$\sum_{s \in S} \sum_{k \in K} x_{ndsk} + \sum_{k \in K} c_{nds'k} - 1 - v^3_{nd} \leq 1 - y'''_{nd} \qquad \forall n \in N \setminus \hat{N}, d \in D \tag{37}$$

$$\sum_{s \in S} \sum_{k \in K} c_{ndsk} + \sum_{s'' \in S^w} \sum_{k \in K} x_{nds''k} + v^4_{nd} \geq 2(y'''_{nd} - \sum_{k \in K} c_{nds'k}) \qquad \forall n \in N \setminus \hat{N}, d \in D \tag{38}$$

$$\sum_{s \in S} \sum_{k \in K} x_{ndsk} \geq \sum_{k \in K} c_{nds'k} \qquad \forall n \in N \setminus \hat{N}, d \in D \tag{39}$$

Objective function (31) minimizes both the original objective function (1) and the costs associated with re-rostering. Constraints (32) ensure that an absent employee is not assigned to any shift. Constraints (33), (34) and (35) calculate the number of changes compared to the original roster and store this information in auxiliary variables. Constraints (36) calculate the shift changes excluding the reserve shifts. Constraints (37) count how many times a reserve shift is changed into a working shift. Constraints (38) calculate the day off to working day changes and vice versa. Finally, constraints (39) ensure reserve shifts are not replaced with a day off.

6 Expected re-rostering cost

The metric proposed in Section 4 serves as an abstract indicator of robustness which quantifies the degree to which a given roster is immune to disruptions without explicitly considering the re-rostering costs which will be incurred when resolving those disruptions. This section proposes a complementary quantification of roster robustness which approximates the re-rostering cost as defined by objective function (31).

Equations (40)-(43) introduce four auxiliary variables which measure various characteristics of a solution defined by the $\hat{x}_{ndsk}$ variables. Let R_d be the size of the reserve shift buffer on day d, and T_{dsk} the total number of employees assigned to shift s on day d with skill k. The size of the capacity buffer on day d for shift s and skill k is denoted by B_{dsk}. For each day d, shift s and skill k, the sum of all other working shifts' capacity buffers is denoted by Q_{dsk}.

$$R_d = \sum_{n \in N} \sum_{k \in K} \hat{x}_{nds'k} \qquad \forall d \in D \tag{40}$$

$$T_{dsk} = \sum_{n \in N} \hat{x}_{ndsk} \qquad \forall d \in D, s \in S^w, k \in K \tag{41}$$

$$B_{dsk} = \max(T_{dsk} - m_{dsk}, 0) \qquad \forall d \in D, s \in S^w, k \in K \tag{42}$$

$$Q_{dsk} = \sum_{s'' \in S^w \setminus \{s\}} \sum_{k \in K} B_{ds''k} \qquad \forall d \in D, s \in S^w, k \in K \tag{43}$$

Let $P_{dsk}(X=a)$ be the probability that a employees are absent for skill k and shift s on day d. Given the probability σ_{dsk} that an employee is absent for skill k and shift s on day d, $P_{dsk}(X=a)$ is computed as shown in Equation (44).

$$P_{dsk}(X=a) = (\sigma_{dsk})^a \times (1-\sigma_{dsk})^{T_{dsk}-a} \times \binom{T_{dsk}}{a} \tag{44}$$

The proposed approximation of the total re-rostering cost is based on the assumption that disruptions are first repaired using the capacity buffers and reserve shift buffers before changing existing working shifts or calling in employees who had a day off. In other words, the costs associated with these various repair operations are assumed to have the following relation[1]: $\omega_n^3 < \omega_n^2 < \omega_n^4$. A second assumption is that there will always be a sufficient number of employees who have a day off to cover absences which cannot otherwise be resolved. Equation (45) calculates an approximation of the expected total re-rostering cost.

$$\begin{aligned}\sum_{d\in D}\sum_{s\in S^w}\sum_{k\in K}\sum_{a=0}^{T_{dsk}} P_{dsk}(X=a)\times(\quad & \omega^3\times \quad \min(R_d, \max(0, a-B_{dsk})) + \\ & \omega^2\times \quad \min(Q_{dsk}, \max(0, a-B_{dsk}-R_d)) + \\ & \omega^4\times \quad \max(0, a-B_{dsk}-R_d-Q_{dsk}))\end{aligned} \tag{45}$$

The term $\min(R_d, \max(0, a-B_{dsk}))$ counts how many employees in the reserve shift buffer are used based on the size of this buffer and how many absences were already covered using the capacity buffer $(a-B_{dsk})$. Any remaining absences are then resolved with at most the number of employees in the reserve shift buffer. The term $\max(0, a-B_{dsk}-R_d)$ counts how many absences are covered using other shifts' capacity buffers. The final term $\max(0, a-B_{dsk}-R_d-Q_{dsk})$ counts the remaining absences which are repaired by calling in employees who have a day off. Each term is multiplied by its respective re-rostering cost. To arrive at the expected cost, the total sum is multiplied by probability $P_{dsk}(X=a)$, for each $d \in D$, $s \in S^w$, $k \in K$ and $a = \{0,\ldots,T_{dsk}\}$.

Given the way Equation (45) is constructed, it is possible that a single reserve shift is used to repair disruptions in different shifts. Similarly, a single nurse's existing assignment may be changed to cover multiple absences which occur in different shifts. Section 7.5 experimentally evaluates the gap between the cost computed by Equation (45) and the true expected cost obtained after simulation.

7 Computational study

The primary purpose of the computational study is to investigate both the accuracy of the proposed robustness quantification and the impact of capacity and reserve shifts on operational costs. Moreover, insights are provided concerning various strategies for introducing robustness to a roster: *(i)* specifying a general minimum robustness level leaving it to the solver to distribute it among the days and employees, or *(ii)* specifying a minimum robustness level per day while an optimization solver determines over which skills it should be distributed, or *(iii)* specifying a minimum robustness level per day and skill.

7.1 Experimental setup and data

All models were implemented in Java and compiled with OpenJDK 1.8. The experiments were conducted on an AMD FX-8150 eight-core processor at 1400 MHz with 32GB of RAM memory running Linux Ubuntu 16.04.3 64-bit. The integer programming problems were solved using CPLEX 12.7.1 with default parameters. No time limit was imposed and all problems were solved to optimality.

[1]We assume these costs are identical for all employees and therefore drop the subscript n in the remainder of this section.

The computational experiments were conducted using problem data from the second International Nurse Rostering Competition (Ceschia et al., 2019) as they represent a collection of constraints and problem characteristics often encountered in practice. The employed problem instance consists of 35 nurses and a planning horizon of four weeks. There are four regular shifts *early (E)*, *late (L)*, *day (D)* and *night (N)* in addition to the *reserve shift (R)*. In the multi-skill setting, the considered skills are $K = \{H,N,C,T\}$ which correspond to *head nurse*, *nurse*, *caretaker* and *trainee*, respectively. Table 7 provides the weight settings used in the experiments. It is preferable for a reserve shift to be converted into a working shift rather than converting a day off into a working shift or changing the shift of an already scheduled nurse. The reason behind this assumption is that unexpected last-minute calls to nurses with days off or changes to their existing work schedules negatively impact their personal lives to a significant degree.

Table 7: Rostering and re-rostering weights.

Description	Cost
Rostering costs	
Base wage per day worked ω_n^1	100, 70, 50, 30 [1]
Overtime ω_n^5, 150% of base wage	$150, 105, 75, 45$ [1]
Understaffing ω^6	1×10^6
Reserve shift assignment ω_n^7, 10% of base wage	$10, 7, 5, 3$ [1]
Not meeting required robustness level ω^r	1×10^3
Re-rostering costs	
Changing an employee's assigned shift ω_n^2, 100% of base wage	$100, 70, 50, 30$ [1]
Calling an employee from a reserve shift ω_n^3, 10% of base wage	$10, 7, 5, 3$ [1]
Calling an employee from a day off or cancel a working shift ω_n^4, 150% of base wage	$150, 105, 75, 45$ [1]

[1] Wage cost for head nurse, nurse, caretaker and trainee, respectively, as determined by the employee's highest skill.

The simulation step generates disruptions using a Bernoulli distribution (Ingels and Maenhout, 2015). This study considers an average absenteeism rate of 2.64%, acquired from a general survey of absenteeism in Belgian organizations (SD Worx, 2016). An additional study showed that absences are more likely to occur on Monday and Tuesday than on other days (SD Worx, 2016). This data was combined to derive the daily absence probabilities detailed in Table 8. The weight α_d used to calculate complete roster robustness in Equations (12), (15), (20) and (24) were also set according to this data.

Table 8: Daily absence probabilities

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
5.43%	3.93%	3.17%	3.08%	2.29%	0.42%	0.16%

7.2 Robustness measure evaluation

The first part of the computational study analyzes the degree to which the metric proposed in Section 4 is able to approximate the roster's true robustness. The correlation between these two values is investigated by varying a roster's robustness level, as calculated by the metric, and comparing these values against how exactly the disruptions were repaired. All reported results are averages obtained by repeating the simulation of disruptions and re-rostering 100 times, as detailed in Section 5.

7.2.1 Single-skilled nurses

The first set of experiments involves single-skilled employees and considers five levels of required robustness $R = \{0.00\%, 2.64\%, 5.28\%, 7.92\%, 10.56\%\}$. These values may be reached by assigning zero to four employees to reserve shifts or capacity buffers on each day. For the studied problem instance, more employees, and thus higher robustness levels, were not possible to attain due to the limited number of available employees.

Figure 1a details the results of the first robustness strategy which enforces a minimum robustness level by adding a general constraint based on Equation (16) to model (1) - (10). Dark-gray bars represent the relative number of disruptions solved by converting days off into working shifts, while light-gray bars correspond to disruptions which were solved by converting reserve shifts into working shifts. Without proactively enforcing robustness upon the roster, disruptions can only be solved by converting days off into working shifts. However, by increasing the required robustness, the number of unexpected employee calls is reduced and disruptions are primarily solved by converting the reserve shifts into working shifts. For the highest robustness level of 10.56%, close to 85% of the disruptions were repaired by using reserve shifts.

Figure 1b plots the results of the second strategy by enforcing robustness on each day in the planning horizon using constraints based on Equation (17). In contrast to Equation (16), which enforces robustness as a general constraint, Equation (17) ensures the required robustness per day. The results show that with a robustness level of 10.56%, almost all the disruptions could be repaired using reserve shifts. For the single-skill setting, it can therefore be concluded that adding robustness constraints on each day reduces the number of unexpected employee calls the most, as reserve shifts are distributed over the planning horizon according to the disruption probability. Moreover, the correlation between the proposed metric and the roster's actual robustness is clearly observable: higher robustness levels considerably reduce the number of unexpected calls necessary to repair roster infeasibility.

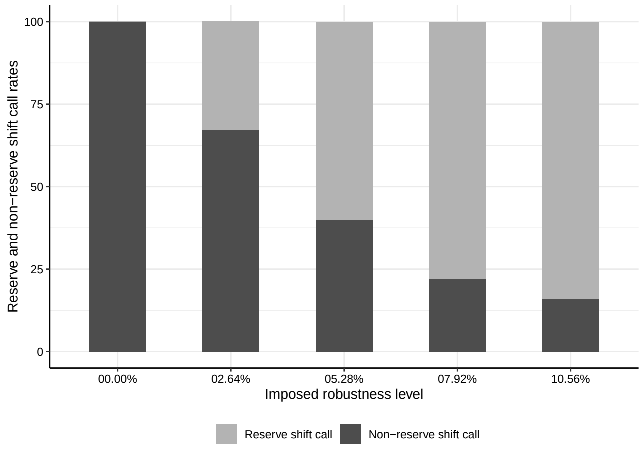
Table 11 in Appendix A reports computation times for generating both the initial and the re-rostering solutions considering these different robustness levels. There is a tendency that higher robustness levels require more computational time. For example, without considering any robustness, an optimal solution was found in 0.6 seconds, while with a robustness level of 10.56%, the computation time increased to 1.4 seconds. In the re-rostering phase, computation times varied between 1.3 and 1.6 seconds, which appeared to be independent of the roster's robustness level. Note that the reported values include the time required by the MIP solver to prove optimality of the solutions.

7.2.2 Multi-skilled nurses

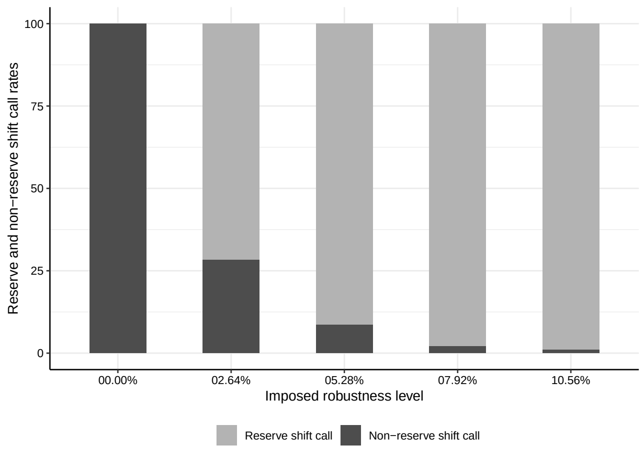
The second series of experiments analyze the impact of robustness in a multi-skill setting. All graphs presented in this section are divided into four groups, corresponding to the four (sub)sets of skills which nurses may be qualified for: $K_1 = \{H\}$, $K_2 = \{H,N\}$, $K_3 = \{H,N,C\}$ and $K_4 = \{H,N,C,T\}$. In the first group, only nurses with the skills in K_1 can be assigned to reserve shifts. For the second group, the nurses are restricted to those with skills in K_2, and so on. Four minimum robustness levels are considered $R = \{2.64\%, 5.28\%, 7.92\%, 10.56\%\}$. For some of the skill subsets, such as K_1, K_2 and K_3, high robustness levels in R could not be attained due to the limited number of nurses who were qualified for these skills.

Figure 2a details the results obtained by requiring a minimum level of robustness using a constraint based on Equation (25). For each of the skill groups, the parameter K in Equations (18) and (19) was set appropriately. The results demonstrate a reduction of unexpected nurse calls (dark-gray bars) when the robustness is increased. Analyzing the effect of increasing robustness over skill groups confirms the intuition that it is more beneficial to assign head nurses to reserve shifts rather than caretakers or trainees. For example, for a robustness level of 2.64%, the best result was obtained when only employees with skill subset K_1 were assigned to reserve shifts, while a decrease of reserve shifts calls was observed when employees of all skills were included.

Figure 2b details the results when robustness is enforced on each day separately using constraints based on Equation (26). The results exhibit the same tendency as when robustness is enforced as an



(a) General

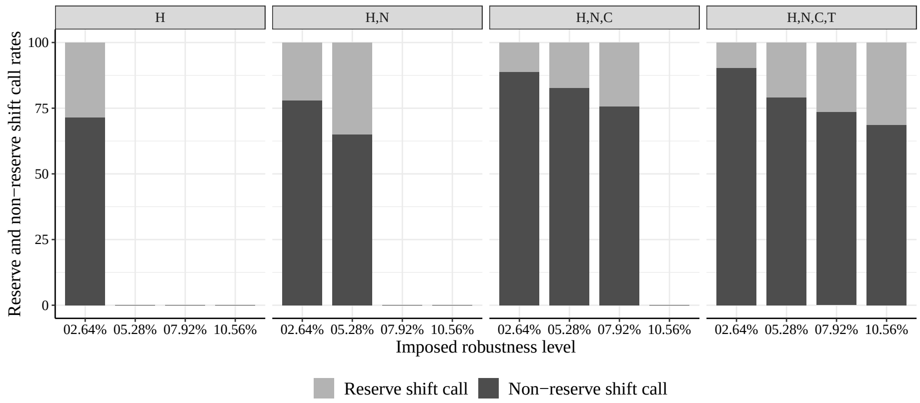


(b) Per day

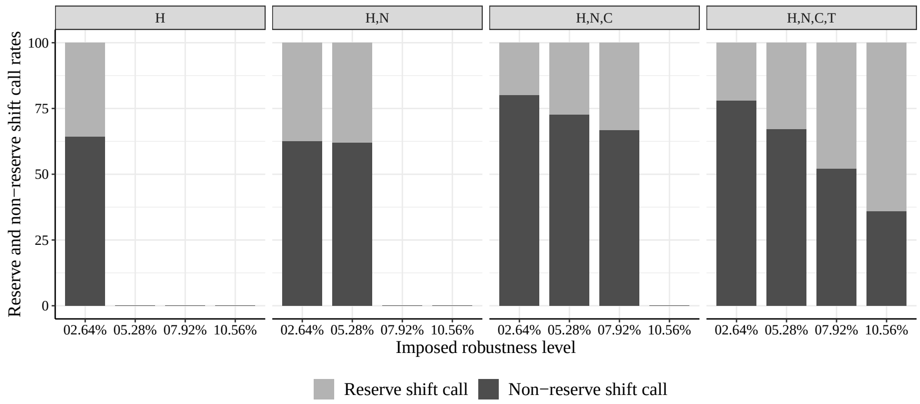
Figure 1: Effect of enforcing robustness on how disruptions were solved.

average over all days of the planning horizon. However, more disruptions were repaired using reserve shifts when enforcing robustness per day compared to enforcing it as a general constraint.

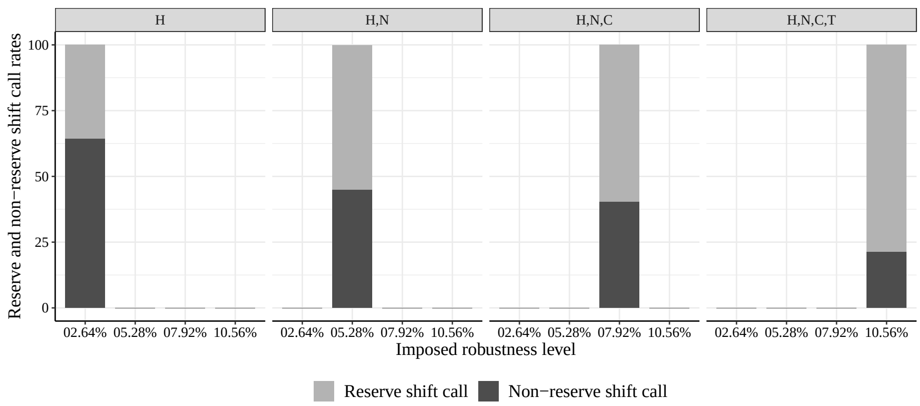
The impact of specifying the required robustness in the most detailed fashion, that is, on each day and each skill separately, is demonstrated in Figure 2c. Here, constraints were added based on Equation (27). Note that the possible robustness values which could be realized per day and per skill are even more restricted due to the limited number of nurses. As before, there is a clear correlation between the required robustness and the number of unexpected employee calls.



(a) General



(b) Per day



(c) Per day and skill

Figure 2: Effect of enforcing robustness on how disruptions were solved.

The results for problems with multi-skilled employees all follow a similar trend: increasing the robustness reduces the number of unexpected calls which necessitated changing a day off into a working shift. These results confirm the validity of how robustness is quantified as introduced in Section 4, since there is a clear correlation with the robustness defined as a constraint for each experiment. Overall, defining robustness constraints in a more detailed manner, per day and skill, generated better results compared against when they were only defined per day or as a general constraint. Moreover, assigning highly-skilled employees to reserve shifts is always better as these employees may assume the work of more employees compared to those with only few skills.

Detailed computation times for the multi-skilled scenarios are reported in Table 12 in Appendix A. The observed trend is similar to the single-skill problems: increasing the robustness level typically requires more computation time to generate the baseline rosters. Compared to the single-skill problems, more computation time was required for finding solutions when considering multiple skills.

7.3 Impact on operational costs

The previous analysis focused on the repair mechanism used to restore roster feasibility after disruptions while ignoring the operational costs induced by the employees' wages associated with a roster. This section investigates the trade-off between managing the operational costs and robustness of a roster. As in Section 7.2, the impact on problems with both single- and multi-skilled employees is evaluated.

7.3.1 Single-skilled nurses

Figure 3a shows the initial cost of the baseline roster and the final cost after disruptions were solved, for different robustness levels specified as a general constraint. The difference between the initial and final costs is due to the additional wage costs incurred by calling employees to cover the absences. Note that the higher the robustness level, the higher the initial cost. While the roster without enforced robustness has the lowest initial cost, it is ultimately associated with the highest final cost. The best result was obtained with a robustness level of 7.92%. Increasing the robustness level to 10.56% generated higher initial costs, while final costs were also higher than those obtained with a robustness level of 8.33%, indicating that more robustness was introduced than necessary.

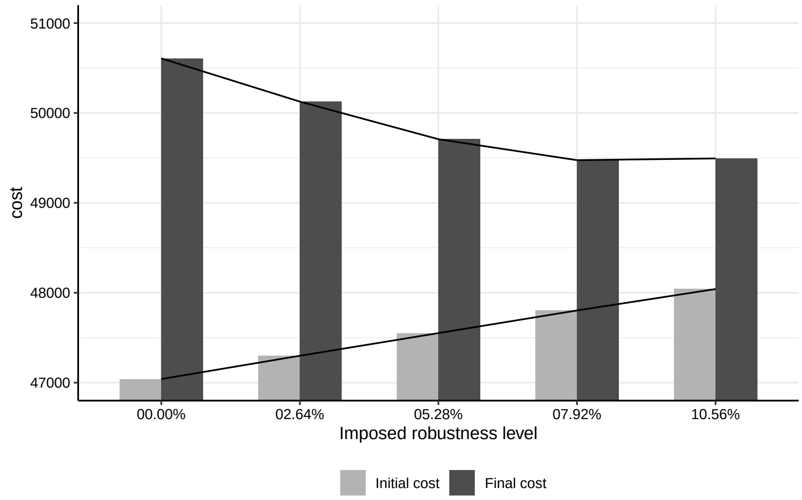
Figure 3b presents the same type of analysis, however, instead of adding a general robustness constraint, the daily robustness constraint forces the solver to evenly distribute the buffers over all days in the planning horizon. The best results were generated with robustness levels of 5.28% and 7.92%. However, the total cost reduction was higher compared to defining robustness as a general constraint. This observation confirms the results from Section 7.2.1 which showed fewer unexpected calls when a robustness constraint was added per day, rather than on a general level.

7.3.2 Multi-skilled nurses

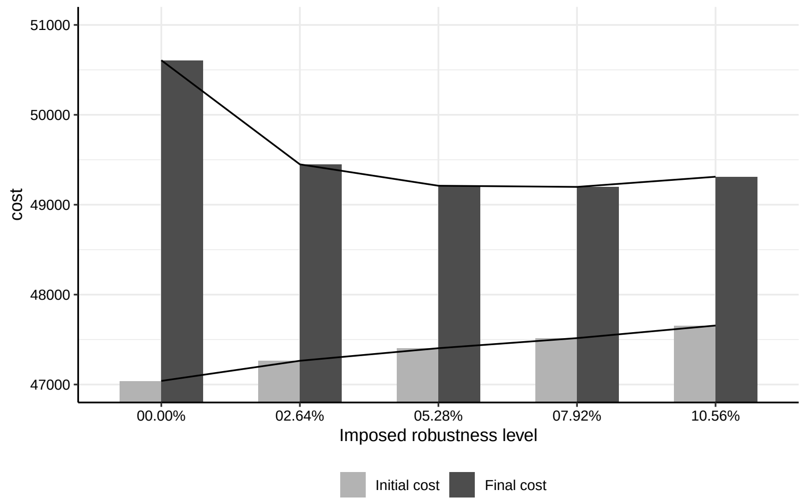
Figure 4a compares the final costs for different robustness levels for the subsets of skills detailed in Section 7.2.2. The leftmost set of bars represents the final cost without any robustness in the roster and is used as a baseline for the comparison. Bars are not shown when the required robustness level could not be attained due to the limited number of available skilled nurses. For a robustness level of 2.64%, the best results were obtained when only head nurses are assigned to reserve shifts. When increasing the robustness level to 5.28%, it was best to allow head nurses and nurses to be assigned to reserve shifts. Such results are expected due to the greater number of disruptions that can be covered by highly-skilled nurses.

Figure 4b compares the final costs when robustness is enforced as a constraint per day. Similar to previous experiments, only assigning highly-skilled employees to reserve shifts resulted in lower final costs. For example, when enforcing a robustness level of 5.28% with only head nurses and nurses, considerably better results are observed compared to when caretakers and trainees were also assigned to reserve shifts.

Figure 4c compares final costs when the required robustness level constraints are specified per day and skill. The lowest final cost was achieved using employees with head nurse, nurse, caretaker and trainee skills for a robustness level of 10.56%. Despite that fact that trainees cannot contribute towards



(a) General



(b) Per day

Figure 3: Comparison of robustness levels and respective initial cost and final cost after repair.

covering disruptions of other skills, a lower final cost was obtained compared to assigning only higher skilled nurses to reserve shifts.

7.3.3 Comparison of robustness strategies

Figure 5 summarizes the performance of the different strategies used to increase robustness. The best results were obtained when constraints were specified per day and skill, while a general robustness constraint (averaged over all days and skills) resulted in the smallest cost reduction. The primary reason for these results is that constraints enforcing robustness per day, or per day and skill, generate a distribution of reserve shifts that better correlates with the disruption probability over the planning horizon, which avoids any possible compensation effects when considering an aggregate metric. For example, employing a general constraint may result in multiple reserve shifts on some days while leaving others without coverage.

Specifying constraints considering employees' skills generated better results compared to incorporating daily robustness constraints since good solutions tend to assign reserve shifts to less qualified employees who have comparably lower costs associated with them. When employees' skills are considered, higher qualified employees are assigned to reserve shifts. Since they also have a higher probability of being able to replace absent employees, this strategy generally performs better.

7.4 Cost of reserve shifts

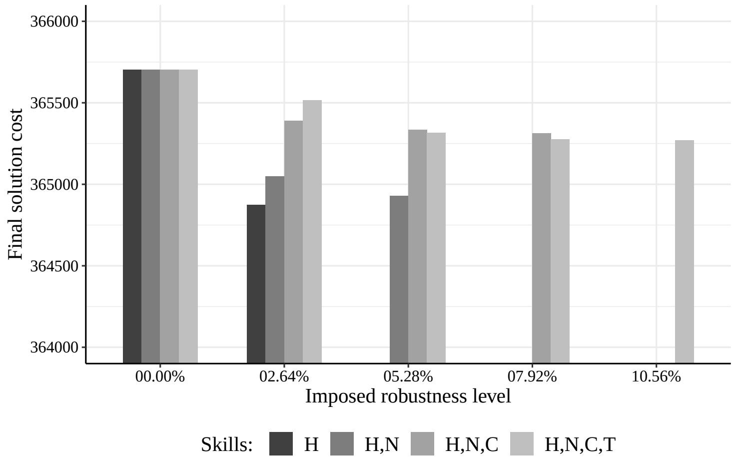
As shown in Table 7, a cost of 10% of a nurse's base wage was incurred when assigning them to a reserve shift. This experiment investigates whether the investment in robustness is still worthwhile when the cost of assigning nurses to reserve shifts increases to 20% or 30% of their base wage. Figure 6 compares the final cost of a roster when robustness is enforced as a constraint per day. For all robustness levels, the final cost was lower compared to rosters without robustness, indicating that, even when the wage cost increases, reserve shifts are still a good investment to ensure robustness in a roster. Only when increases the cost to 30% and enforcing a robustness level of 5.28% with head nurses, nurses and caretakers, the final cost becomes comparable to a roster without robustness.

7.5 Expected re-rostering cost

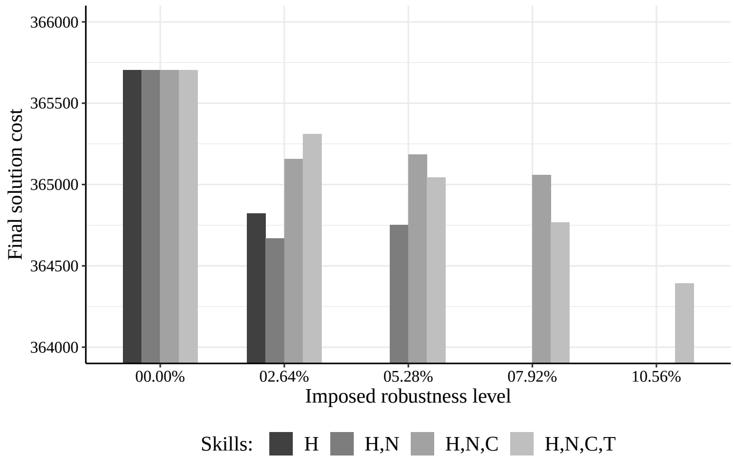
Tables 9 and 10 compare, for different robustness levels, the expected total re-rostering cost $\mathbb{E}(C)$ and the final cost for problems with single-skilled and multi-skilled employees, respectively. Equation (45) was used to compute the expected total re-rostering cost while the final cost was obtained after simulation. Note that only instances for which the minimum staffing requirements were satisfied after re-rostering were considered in these results. Problem instances for which disruptions lead to solutions with understaffing, and thus large penalties ω^6, were not included.

When considering single-skilled employees, the gap between the expected cost and the final cost was always less than 3.4%. When increasing robustness levels, this gap decreased. Moreover, when imposing the required robustness level on each day of the planning horizon, the gaps are, on average, smaller than when employing a general robustness constraint. Similar trends are observed for problems with multi-skilled employees, where the gap between the expected cost and the final cost was at most 5.1%.

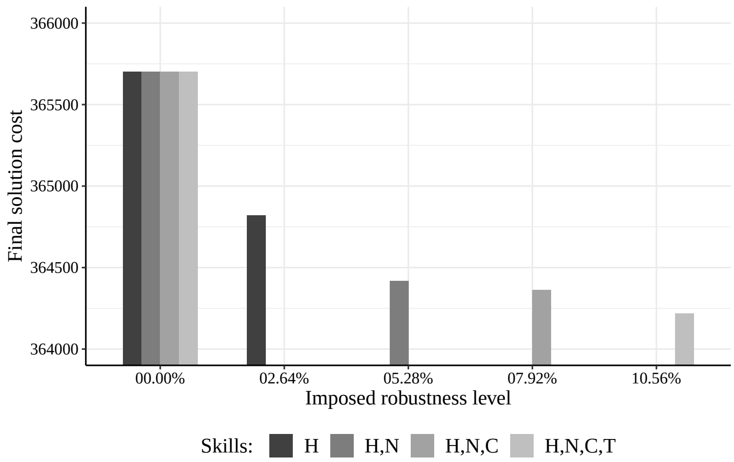
As discussed in Section 6, Equation (45) approximates the true expected cost by allowing disruptions in different shifts to be covered using the same employees. Given that the gaps reported in Tables 9 and 10 are relatively small, it can be concluded that typically few of the costly re-rostering operations are necessary and that most disruptions can be resolved using reserve shifts. This hypothesis is supported by the costs associated with the different buffer types. Reserve shifts only cost 10% of an employee's base wage, making them considerably cheaper than regular working shifts. Capacity buffers are thus rarely installed due to the much higher initial costs associated with them. Moreover, they lead to higher



(a) General



(b) Per day



(c) Per day and skill

Figure 4: Comparison of robustness levels and respective costs.

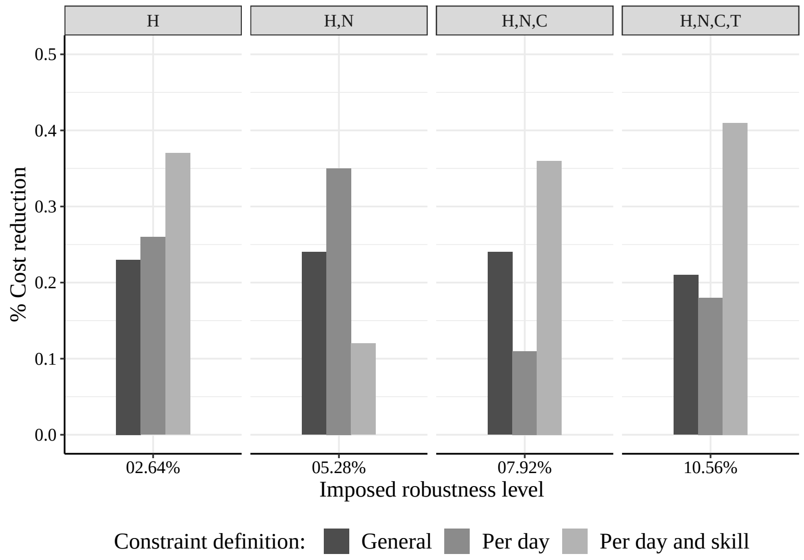


Figure 5: Comparison of robustness strategies for multi-skilled employees

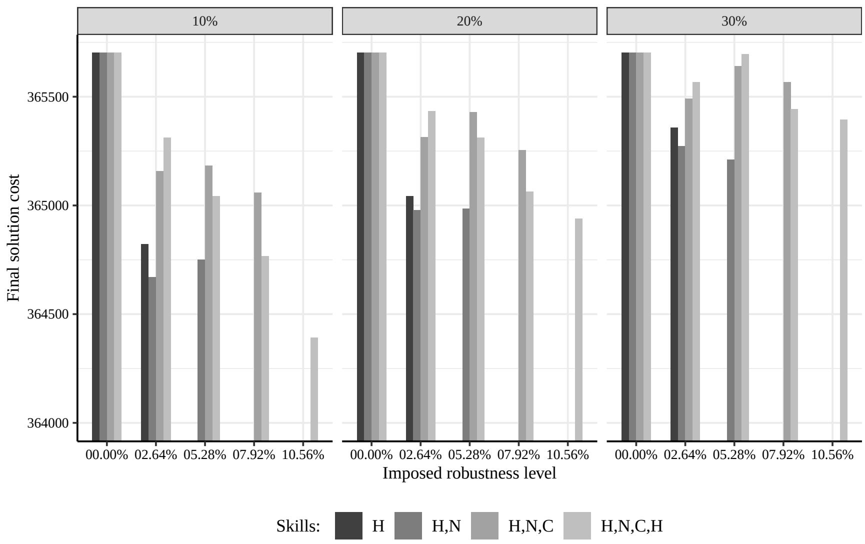


Figure 6: Robustness levels per day with reserve shift costs of 10%, 20% and 30% of nurses' base wage.

costs in the re-rostering phase: 100% of the base wage for changing a regular working shift to another type, compared to 10% for converting a reserve shift to a working shift.

Table 9: Expected and final re-rostering cost for problems with single-skilled employees

	General constraint			Per day constraint		
Robustness	$\mathbb{E}$(C)	Final cost	Gap (%)	$\mathbb{E}$(C)	Final cost	Gap (%)
0.00%	48903	50607	3.37	47040	50607	3.37
2.64%	48744	50126	2.76	47264	49449	1.50
5.28%	48846	49708	1.73	47404	49211	1.04
7.92%	49056	49475	0.85	47516	49198	0.87
10.56%	49285	49494	0.42	47656	49311	0.83

Table 10: Expected and final re-rostering cost for problems with multi-skilled employees

	General constraint			Per day constraint			Per day and skill constraint		
Robustness	$\mathbb{E}(C)$	Final cost	Gap (%)	$\mathbb{E}(C)$	Final cost	Gap (%)	$\mathbb{E}(C)$	Final cost	Gap (%)
0.00%	43351	45682	5.10	43351	45682	5.10	43351	45682	5.10
2.64%	43471	44851	3.08	43451	44765	2.93	43461	44765	2.91
5.28%	43576	44894	2.94	43257	44683	3.19	43466	44379	2.06
7.92%	43644	45295	3.65	43228	45072	4.09	43476	44307	1.88
10.56%	43771	45268	3.31	43357	44381	2.31	43485	44196	1.61

8 Conclusions

This paper sought to remedy a shortcoming in the literature by introducing two complementary functions for quantifying the robustness provided by capacity and reserve shift buffers in staff rostering problems. The first metric provides an a priori estimation of how well a roster can accommodate unexpected employee shortages, while the second metric approximates the expected total re-rostering cost. In addition, an integer programming formulation was introduced with the option of imposing a required minimum level of robustness.

A computational study demonstrated how enforcing relatively low robustness levels appeared sufficient to result in the lowest final cost for problems with single-skilled employees. Higher levels of robustness resulted in higher initial and final costs as too many employees were unnecessarily assigned to reserve shifts. Similar results were observed for multi-skilled employees. However, since employees can only substitute for others when they share the same skills, multi-skilled environments require a higher robustness level than single-skilled environments. In this case, assigning more highly qualified employees to reserve shifts generated better results than assigning, for example, trainee nurses. Despite the fact that highly qualified employees are associated with higher costs, they are also capable of remedying a far greater percentage of the personnel shortages which may occur.

The proposed approximation of the expected total re-rostering cost was shown to be relatively accurate, in particular for problems with single-skilled employees. An analysis of the results of a computational study indicated that installing reserve shift buffers is generally preferable over capacity buffers due their lower wage costs and higher flexibility when repairing disruptions.

Future research on staff rostering will benefit from the standardized quantification of roster robustness proposed in this paper. The proposed approximation of the total re-rostering cost may be used to determine the optimal size of capacity and reserve shift buffers. While there are some computational challenges associated with this, it would greatly benefit organizations when it comes to protecting themselves against disruptions caused by employee absenteeism.

Acknowledgment

We acknowledge the support provided by FWO as part of ORDinL (Data-driven Logistics) under the Strategic Basic Research (SBO) program, and KU Leuven C24/14/012 (Dynamic Combinatorial Optimization). Editorial consultation provided by Luke Connolly (KU Leuven).

A Additional computational details

Tables 11 and 12 provide the computation times for the single- and multi-skilled problems, respectively.

Table 11: Computational time for single-skill instances, in seconds.

Robustness level	Initial solution: General constraint	Initial solution: Robustness constraint per day	Re-rostering solution: General constraint	Re-rostering solution: Robustness constraint per day
0.00%	0.6	0.6	1.4	1.3
2.64%	1.0	1.0	1.4	1.3
5.28%	0.9	1.2	1.5	1.4
7.92%	1.5	1.8	1.5	1.5
10.56%	1.4	2.5	1.6	1.5

Table 12: Computational time for multi-skill instances, in seconds.

	Initial solution											
Robustness	General constraint				Robustness constraint per day				Robustness constraint per day and skill			
level	H	H,N	H,N,C	H,N,C,T	H	H,N	H,N,C	H,N,C,T	H	H,N	H,N,C	H,N,C,T
2.64%	1.5	1.7	2.9	1.3	78.9	2.4	2.3	1.5	76.2	-	-	-
5.28%	-	1.5	4.7	1.7	-	2.4	3.3	1.9	-	212.1	-	-
7.92%	-	-	2.1	2.2	-	-	1.7	3.9	-	-	194.7	-
10.56%	-	-	-	14.3	-	-	-	9.5	-	-	-	300.9
	Re-rostering solution											
Robustness	General constraint				Robustness constraint per day				Robustness constraint per day and skill			
level	H	H,N	H,N,C	H,N,C,T	H	H,N	H,N,C	H,N,C,T	H	H,N	H,N,C	H,N,C,T
2.64%	2.6	2.7	3.2	2.6	2.6	2.2	2.3	2.2	2.3	-	-	-
5.28%	-	2.7	2.6	3.0	-	2.3	2.1	2.5	-	2.0	-	-
7.92%	-	-	2.8	2.7	-	-	2.2	2.1	-	-	2.0	-
10.56%	-	-	-	2.8	-	-	-	1.9	-	-	-	2.0

References

Abdelghany KF, Abdelghany AF, Ekollu G (2008) An integrated decision support tool for airlines schedule recovery during irregular operations. European Journal of Operational Research 185(2):825 – 848, DOI 10.1016/j.ejor.2006.12.045

Bertsimas D, Sim M (2004) The price of robustness. Operations Research 52:35–53

Bäumelt Z, Dvořák J, Šůcha P, Hanzálek Z (2016) A novel approach for nurse rerostering based on a parallel algorithm. European Journal of Operational Research 251(2):624 – 639

Bureau of Labor Statistics (2017) Absences from work of employed full-time wage and salary workers by occupation and industry. URL https://www.bls.gov/cps/cpsaat47.htm, [Online; accessed 06-June-2018]

Ceschia S, Dang N, De Causmaecker P, Haspeslagh S, Schaerf A (2019) The second international nurse rostering competition. Annals of Operations Research 274(1):171–186

De Bruecker P, Van den Bergh J, Beliën J, Demeulemeester E (2015) Workforce planning incorporating skills: State of the art. European Journal of Operational Research 243(1):1 – 16, DOI 10.1016/j.ejor.2014.10.038

Drake RG (2014) The ‘robust’roster: exploring the nurse rostering process. Journal of advanced nursing 70(9):2095–2106

Edwards P, Greasley K (2010) Absence from work. URL https://www.eurofound.europa.eu/observatories/eurwork/comparative-information/absence-from-work, [Online; accessed 06-June-2018]

Forbes (2013) The causes and costs of absenteeism in the workplace. URL https://www.forbes.com/sites/investopedia/2013/07/10/the-causes-and-costs-of-absenteeism-in-the-workplace, [Online; accessed 04-May-2018]

Guo M, Wu S, Li B, Rong Y (2014) Maximizing the efficiency of use of nurses under uncertain surgery durations: A case study. Computers & Industrial Engineering 78:313 – 319, DOI 10.1016/j.cie.2014.04.014

Ingels J, Maenhout B (2015) The impact of reserve duties on the robustness of a personnel shift roster: An empirical investigation. Computers & Operations Research 61(Supplement C):153 – 169, DOI 10.1016/j.cor.2015.03.010

Ingels J, Maenhout B (2017a) Employee substitutability as a tool to improve the robustness in personnel scheduling. OR Spectrum 39(3):623–658, DOI 10.1007/s00291-017-0476-0

Ingels J, Maenhout B (2017b) Optimised buffer allocation to construct stable personnel shift rosters. Omega DOI 10.1016/j.omega.2017.12.006

Ingels J, Maenhout B (2018) The impact of overtime as a time-based proactive scheduling and reactive allocation strategy on the robustness of a personnel shift roster. Journal of Scheduling 21(2):143–165, DOI 10.1007/s10951-017-0512-6

Kocakulah MC, Kelley AG, Mitchell KM, Ruggieri MP (2016) Absenteeism problems and costs: causes, effects and cures. The International Business & Economics Research Journal (Online) 15(3):89

Lusby R, Dohn A, Range TM, Larsen J (2012) A column generation-based heuristic for rostering with work patterns. Journal of the Operational Research Society 63(2):261–277

Maenhout B, Vanhoucke M (2011) An evolutionary approach for the nurse rerostering problem. Computers & Operations Research 38(10):1400 – 1411

Moz M, Pato MV (2003) An integer multicommodity flow model applied to the rerostering of nurse schedules. Annals of Operations Research 119(1):285–301

SD Worx (2016) Absenteeism in belgium. URL https://www.sdworx.be/~/media/sd%20worx/blog/ziekteverzuim%20in%20belgie%202016.pdf?la=nl-be&hash=25E78DDB6D5B444C6177D4A3945FF6A2E37852FE, [Online; accessed 23-November-2020]

Smet P, Bilgin B, De Causmaecker P, Vanden Berghe G (2014) Modelling and evaluation issues in nurse rostering. Annals of Operations Research 218(1):303–326

The Society for Human Resource Management (2016) Total financial impact of employee absences. URL https://www.shrm.org/hr-today/trends-and-forecasting/research-and-surveys/Documents/Total%20Financial%20Impact%20of%20Employee%20Absences%20Report.pdf, [Online; accessed 03-July-2018]

Van den Bergh J, Beliën J, De Bruecker P, Demeulemeester E, De Boeck L (2013) Personnel scheduling: A literature review. European Journal of Operational Research 226(3):367 – 385

Wickert TI, Smet P, Vanden Berghe G (2019) The nurse rerostering problem: Strategies for reconstructing disrupted schedules. Computers & Operations Research 104:319 – 337, DOI 10.1016/j.cor.2018.12.014

Williams JC, Lambert S, Kesavan S (2017) How the gap used an app to give workers more control over their schedules. URL: https://hbr.org/2017/12/how-the-gap-used-an-app-to-give-workers-more-control-over-their-schedules [Online; accessed 25-April-2018]