Achieving compromise solutions in nurse rostering by using automatically estimated acceptance thresholds

Böðvarsdóttir, Elín Björk; Smet, Pieter; Vanden Berghe, Greet; Stidsen, Thomas Jacob Riis

Published in:
European Journal of Operational Research

Link to article, DOI:
10.1016/j.ejor.2020.11.017

Publication date:
2021

Document Version
Peer reviewed version

Link back to DTU Orbit

Citation (APA):
Böðvarsdóttir, E. B., Smet, P., Vanden Berghe, G., & Stidsen, T. J. R. (2021). Achieving compromise solutions in nurse rostering by using automatically estimated acceptance thresholds. *European Journal of Operational Research*, *292*(3), 980-995. https://doi.org/10.1016/j.ejor.2020.11.017

Achieving compromise solutions in nurse rostering by using automatically estimated acceptance thresholds[☆]

Elín Björk Böðvarsdóttir[a,*], Pieter Smet[b], Greet Vanden Berghe[b], Thomas J. R. Stidsen[a]

[a]*Department of Technology, Management and Economics, Technical University of Denmark, Anker Engelundsvej 1, 2800 Kgs. Lyngby, Denmark*

[b]*KU Leuven, Department of Computer Science, CODeS, Gebroeders De Smetstraat 1, Gent 9000, Belgium*

Abstract

Despite the multi-objective nature of the nurse rostering problem (NRP), most NRP formulations employ a single evaluation function that minimizes the weighted sum of constraint violations. When solving the NRP in practice, the focus should be on obtaining compromise solutions: those with appropriate trade-offs between different constraints. Due to the real-world characteristics of the problem, appropriate trade-offs may vary substantially across instances, and quantifying these trade-offs does not necessarily translate well into a single evaluation function. This paper introduces a new multi-objective approach for the NRP that promotes controlled trade-offs and guides the solver towards acceptable compromise solutions. The method consists of two phases. The first phase quantifies the characteristics of acceptable compromise solutions by estimating acceptance thresholds that implicitly incorporate trade-offs. This quantification is performed automatically by drawing upon the instance at hand and identifying appropriate trade-offs. The second phase solves the NRP by employing these acceptance thresholds in a lexicographic goal programming framework. By automatically estimating instance-specific acceptance thresholds, we not only require minimal information from the user but also obtain a realistic prediction for solution quality. A case study shows that the methodology produces rosters with little or no deviations from acceptance thresholds, within only a few minutes. Furthermore, this methodology provides the user with clear reasoning behind the trade-offs made, as opposed to methods employing a single evaluation function.

Keywords: Timetabling, Nurse rostering, Multi-objective optimization, Compromise solutions, Lexicographic goal programming

[☆] Declarations of interest: none

[*] Corresponding author

Email addresses: `ebod@dtu.dk` (Elín Björk Böðvarsdóttir), `pieter.smet@kuleuven.be` (Pieter Smet), `greet.vandenberghe@cs.kuleuven.be` (Greet Vanden Berghe), `thst@dtu.dk` (Thomas J. R. Stidsen)

1. Introduction

The nurse rostering problem (NRP) is the task of assigning nurses to shifts to generate a feasible work schedule. This problem belongs to the class of personnel scheduling problems, which consider different professions and scenarios. Researchers have studied these problems for several decades and offered multiple formulations and solution methods (Van Den Bergh et al., 2013). The NRP is typically formulated using a set of hard and soft constraints, where the hard constraints may not be violated and where the violation of soft constraints should be minimized.

The NRP generally includes numerous constraints, many of which can conflict with one another. Thus, minimizing the violation of soft constraints is not a trivial task and one may identify different solutions as being optimal depending on the formulation of the evaluation function. Even though this function should measure the quality of a solution, its formulation is inherently subjective and often based on abstract concepts such as objective weights. As rosters can have a significant effect on nurses' health and happiness (Gärtner et al., 2018), the evaluation function should promote *compromise solutions* that avoid negative effects for individual nurses while simultaneously meeting the overall needs of the ward.

In an extensive review of the literature, Burke et al. (2004) identified several promising directions for nurse rostering research. One direction was an increased focus on multi-objective approaches to comply with the multi-objective nature of the problem. Another direction considered improving the ease of use of the developed algorithms, for example, by reducing the number of required parameters or by using simpler parameters that nurse rostering practitioners understand. While a few researchers have addressed the first of these directions (see Section 3), the second aspect poses an intriguing challenge of immense value in practice, for which hardly any methodology exists.

In this paper, we introduce a new multi-objective approach for the nurse rostering problem that guides the solver in exploring promising areas of the search space, i.e., areas including appropriate compromise solutions that the end-users approve. Rather than using abstract concepts to evaluate the quality of a solution, we use the terms *targets* and *acceptance thresholds* (or just *thresholds*) introduced by Böðvarsdóttir et al. (2019c). The methodology consists of two phases, where the first estimates acceptance thresholds for each target and the second produces a solution by drawing upon those thresholds. This methodology is applied to a case from Danish hospitals, which includes 13 different targets.

The solution space for nurse rostering problems is large. One must also consider trade-offs between numerous conflicting aspects. In practice, identifying compromise solutions with appropriate trade-offs is not trivial, as they may differ substantially between nurses or between rostering horizons. Therefore, the first phase of the proposed methodology automatically employs instance-specific information to estimate acceptance thresholds that incorporate the appropriate trade-offs, thereby indicating the promising areas of the search space. The second phase investigates the feasibility of the acceptance thresholds from the first phase, thereafter enforcing them as hard constraints (without causing infeasibility). This enables us to cut away non-promising areas of the search space and reduce its size. This reduction is performed iteratively based on the priorities of the targets, adding a single cut in each iteration.

Even though nurse rostering is concerned with multiple conflicting criteria, very few researchers have addressed the problem with multi-objective techniques. Instead, researchers typically minimize the weighted sum of different constraint violations with a single evaluation function and the majority of the formulations require manual interventions to temper this function. Several nurse rostering researchers and practitioners have criticized this approach, basing their criticism on observations from practice (Gärtner et al., 2018; Petrovic and Vanden Berghe, 2012; Böðvarsdóttir et al., 2019c).

Researchers focusing on the theory of multi-objective decision making have also criticized the use of the weighted sum objective. Branke et al. (2008) identify serious limitations: the promotion of an imbalance between different objectives and the assumption that a trade-off can be made between any two objectives.

Various methods exist for addressing multi-objective optimization problems. These methods are generally categorized into the following four categories (Branke et al., 2008). First, *a priori methods* where the users define their preferences before the optimization and the solution process tries to find a solution satisfying those preferences as well as possible. Second, *a posteriori methods* where the solution process generates a set of Pareto optimal solutions (where no objective can be improved without the solution becoming worse w.r.t. another objective) for the users to choose from. Third, *interactive methods*, where users are closely involved in the solution process by iteratively specifying and adjusting their preferences while learning from the process. Fourth, *no-preference methods* for finding neutral solutions when no preference information is available, which is not relevant for nurse rostering problems.

This paper introduces a new a priori method for the NRP, where users provide targets with distinct priorities. One of the main drawbacks of a priori methods is the difficulty of setting realistic expectations beforehand (Luque et al., 2009). The proposed methodology overcomes this drawback by setting appropriate acceptance thresholds based on instance-specific information. Compared to the other three categories, this a priori method requires significantly less computational effort than a posteriori methods. Additionally, it requires minimal human effort, as the users neither need to evaluate multiple solutions themselves (as in a posteriori methods) nor continuously interact and exchange information with the solver (as is the case with interactive methods).

Böðvarsdóttir et al. (2019c) introduced the concepts of *targets* and *acceptance thresholds* to nurse rostering research in a paper that focused on making nurse rostering models more accessible to practitioners. They discussed four common nurse rostering targets and presented specific techniques for estimating their acceptance thresholds. Moreover, they briefly discussed different opportunities for incorporating these concepts in nurse rostering formulations, including lexicographically.

The main contributions of this paper are threefold: First, we introduce *general* strategies for automatically estimating acceptance thresholds for various targets by utilizing instance-specific information. This estimation takes place before the solution process by identifying conflicts and specifying trade-offs. Second, we present an alternative modeling approach that requires minimal information (i.e., parameters) from users. Third, instance-specific acceptance thresholds guide the search towards appropriate compromise solutions, reducing the length of time wasted on exploring unacceptable solutions.

The remainder of this article is structured as follows. Section 2 provides a general description of nurse rostering problems while Section 3 presents a review of multi-objective approaches for such problems. Section 4 introduces the methodology and Section 5 presents the case we analysed to validate it. Section 6 presents the results and, finally, Section 7 concludes the paper.

2. The nurse rostering problem

The nurse rostering problem focuses on scheduling nurses according to some pre-defined coverage constraints. These constraints are generally defined in terms of shifts, where a structure with three shifts (day, evening, and night) is common in the literature. Most often, coverage constraints describe hard lower bounds for the number of nurses assigned to shifts. To ensure the feasibility of these constraints, we allow the temporary use of external personnel, known as *float nurses*, in addition to the fixed personnel in the ward. Float nurses are employees that are not restricted to working in a specific ward.

In its essence, the NRP is an assignment problem that assigns nurses to shifts. Nevertheless, the problem is complicated by the fact that it involves human resources, requiring various constraints that are irrelevant for other assignment problems.

Different formulations of the NRP vary in complexity (Smet et al., 2016). Some only consider the most basic needs, such as overall restrictions on working hours or assigning days off. Others include more complex aspects, such as the nurses' competencies and preferences as well as constraints on the sequence of assignments on consecutive days. De Causmaecker and Vanden Berghe (2011) provide an $\alpha|\beta|\gamma$-notation for categorizing nurse rostering problems based on the types of constraints included and the objective function to be optimized. This categorization displays the variability of formulations presented throughout the years and highlights the different levels of complexity.

Real-world nurse rostering problems require many constraints. Some of these constraints are legally binding, such as restrictions on working hours and minimal requirements for rest. Other constraints go further to meet the nurses' needs by trying to regulate their sleeping patterns and providing a good work-life balance. Overall, the majority of the constraints are designed to improve the nurses' well-being. These terms are difficult to quantify and, as a result, the exact constraints used to formulate the problem may differ substantially.

Moreover, the real-world problem has some characteristics that impact either the feasibility or preferability of some assignments. For example, nurses may have private obligations preventing them from working a certain shift on a specific day. These characteristics impose additional constraints on the solution space and are often captured using *requests*. These requests can be either hard (impacting the feasibility) or soft (describing the preferability).

In general, nurse rostering problems consist of numerous conflicting constraints, and generating a feasible solution requires multiple trade-offs between different aspects. To generate a compromise solution we need to identify trade-offs that are not only appropriate for the ward as a whole but also acceptable to the nurses as individuals. For example, a nurse whose requests are repeatedly refused will perceive the roster as poor and perhaps favorable to others. Therefore, a good roster should promote the perception of fairness by balancing unfavorable trade-offs between the nurses.

Even though the term trade-off implies making a sacrifice, the loss is not always significant. For example, a general constraint promoting compact work schedules and a nurse's request for an isolated work shift are conflicting. Nonetheless, making a trade-off by "sacrificing" the general constraint does not constitute a loss to this nurse, as her specific request is accommodated. Mihaylov et al. (2016) indicated the weakness of weighted sum objective functions and showed that setting appropriate weights may not be intuitive. This paper pursues alternative solution methods to capture the underlying multi-objective nature of the problem.

3. Literature review

In this review, we consider multi-objective approaches to model and solve personnel rostering problems. As the focus is on alternative modeling approaches, we exclude any papers that apply weighted-sum evaluation functions (including weighted goal programming).

Burke et al. (2002) present a compromise programming approach where the objective function evaluates different solutions by comparing them to an ideal point. To set this point, they find the best value that each objective can take by optimizing it without considering any other objectives. Despite being infeasible for many problems, the ideal point indicates what degree of slack is necessary. In addition to the ideal point, Burke et al. (2002) also define an anti-ideal point based on the worst possible values for all objectives along with weights to reflect the importance of each objective. To evaluate a solution, they measure its distance to the ideal point, relative to the anti-ideal point. By doing so, they account for a difference in measurement units between the objectives, along with the ranges of possible violations for each objective. By combining compromise programming with user-defined weights, Burke et al. (2002)'s methodology categorizes as an a priori method.

Li et al. (2012) also present a compromise programming method, where they identify the ideal point by solving a goal programming model for each of the objectives one by one. Then, they apply an a posteriori method where they use a meta-heuristic to generate an approximation of the Pareto set. Due to the computational complexity of the problem and the vast search space, they are unable to use exact methods to achieve the true Pareto set. After generating the approximated set, they reduce its size by applying filtering techniques that draw upon a general preference ordering of the objectives. Thus, the final output of their approach is a set of less than one hundred "highly-preferable" rosters for a manager to choose from.

Burke et al. (2012) present a Pareto-based search methodology for the same problem formulation and instances as Li et al. (2012). After generating a feasible starting solution, Burke et al. (2012) use a meta-heuristic to optimize the different objectives and generate an approximation of the Pareto set. For comparison, they employ both the standard weighted-sum evaluation function and a domination-based evaluation function. In their results, they report an approximated set of up to two thousand non-dominated solutions.

The largest practical drawback of a posteriori methods is the difficulty users face when choosing a single solution from all the possibilities they are presented with (Branke et al., 2008). Li et al. (2012) touch upon these issues, by applying general preferences to reduce the number of solutions that they present. Nonetheless, choosing a single roster from up to a hundred alternatives is a cognitively challenging task, which Li et al. (2012) do not acknowledge. Furthermore, neither Li et al. (2012) nor Burke et al. (2012) address how to present the solutions such that they are informative enough to make accurate comparisons while still being easy to understand. Tackling these critical issues is necessary for a posteriori methods to be beneficial in practice.

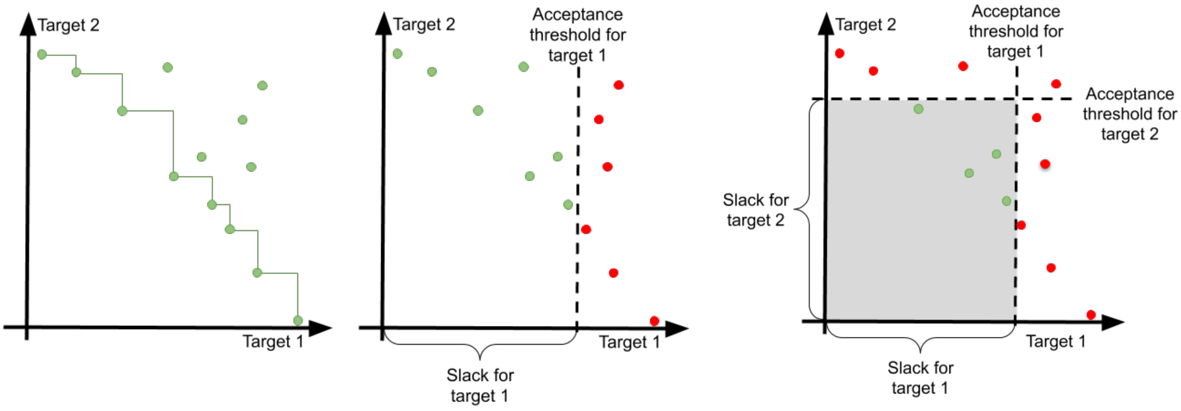
Rihm and Baumann (2018) introduce a new formulation for employee scheduling problems, namely the staff assignment problem with lexicographically ordered acceptance levels (SAP-LAL). Although this is an a priori method, they do not rely on user-defined weights but instead, use manually defined functions to map measured constraint violations into so-called *acceptance levels* (ranging from 0 to 100). Each soft constraint can have one or more acceptance levels describing the severity of its violation. For example, an acceptance level of 100 for the first violation (meaning that it is fully acceptable), a level of 60 for the second violation, and a level of 0 for three or more violations (making those solutions infeasible). After setting these acceptance levels, Rihm and Baumann (2018) solve the problem lexicographically by minimizing violations based on increasing order of the acceptance levels, to ensure that the solution does not consider a trade-off between constraint violations of different severity. Although this method resembles the method we introduce (especially in the terms that are used), we will highlight significant differences in Section 4.

This review presents multi-objective alternatives for modeling and solving the nurse rostering problem. All of these methods require intuitive information from the users, in the form of weights, mapping functions, or a large number of possible solutions to choose from.

In the remainder of this paper, we present a multi-objective methodology that automatically employs instance-specific information to find an appropriate compromise solution, without requiring any user interactions.

4. Methodology

This section introduces a new multi-objective methodology for addressing real-world nurse rostering problems. This methodology considerably restricts the solution space as demonstrated in Figure 1. The methodology draws upon the concepts of *targets* and their *acceptance thresholds*. We define a *target* as a measurable solution characteristic that practitioners review when evaluating the quality of a roster. For each target, we define its *acceptance threshold* (or just *threshold*) as the maximum acceptable value for that target (assuming minimization). If the value is below the threshold, we will refer to it as *slack* and if the value exceeds the threshold, we refer to it as a *target violation*.



(a) Before introducing any cuts. (b) After introducing a cut for the first target. (c) After introducing a cut for both targets.

Figure 1: The evolution of the search space when iteratively enforcing acceptance thresholds (dashed lines) as hard constraints, where the dots (·) represent different solutions. The green dots represent solutions that belong to the feasible region, while the red solutions exceed the acceptance threshold for some targets.

The methodology consists of two phases presented in Sections 4.1-4.2, respectively. The first phase automatically estimates acceptance thresholds by drawing upon instance-specific information. The second phase is a lexicographic goal programming approach that addresses the targets in a prioritized order, verifying the feasibility of the corresponding acceptance thresholds and imposing them as hard constraints before moving towards lower priority targets.

Figure 1 illustrates how the second phase iteratively reduces the search space, allowing the solver to focus on exploring promising areas that include appropriate compromise solutions. In Figure 1a we indicate the Pareto front (green line) and note that several of the Pareto optimal solutions become infeasible when introducing the acceptance thresholds as hard constraints (Figures 1b-1c).

The grey area (Figure 1c) represents the feasible region after cutting away solutions that exceed the threshold for either target.

A lexicographic framework has two main advantages: it prevents a direct comparison between targets with different measurement units and it has the potential to increase the control over how we explore the search space. However, the accuracy of the acceptance thresholds is crucial with respect to obtaining increased control. For each target, we reduce the size of the search space by cutting away solutions where constraint violations exceed the corresponding acceptance threshold. While too tight thresholds will unnecessarily restrain the search by cutting away acceptable solutions, too loose thresholds will result in a larger search space and even the generation of unacceptable solutions.

The methodology we introduce is in many ways related to interactive multi-objective methods (e.g., reference point approaches), where the users do not know which levels they can obtain beforehand and gradually learn, thus adjusting their expectations (Branke et al., 2008). Both methodologies are easy to use and the information exchanged with the users is clear. The method we present does not require direct interactions with the users during the solution process, as opposed to interactive approaches. Instead, it automatically employs instance-specific information to estimate accurate thresholds. These estimations replace the explorations that users would have to do themselves for every single instance when employing interactive approaches. We assert that automating the process and requiring limited involvement from users is crucial in an environment with diverse instances (both multiple wards and changes between rostering horizons). Furthermore, we centralize the knowledge within the algorithms, making the approach more robust in practice, as practitioners may come and go.

At last, we acknowledge that this methodology bears some resemblance to Rihm and Baumann (2018)'s work. Both methods employ lexicographic goal programming along with acceptance thresholds (or acceptance levels). However, the key difference between the two is that Rihm and Baumann (2018) employ manually-defined mapping functions to generate their acceptance levels, while we automatically estimate appropriate acceptance thresholds based on the instance at hand.

4.1. Estimating acceptance thresholds

In this section, we will discuss several alternatives for estimating acceptance thresholds, the simplest being to set them manually. Often practitioners have some rules of thumb (e.g., aiming for no slack) that we could use as general acceptance thresholds. As the NRP is tightly constrained, most often some slack for a few constraints is inevitable and practitioners' rules of thumb may be infeasible. Furthermore, the appropriate value for a given acceptance threshold may differ, both between nurses and between rostering horizons. Thus, if we employ manually set thresholds, we risk that the previous challenge of *tuning the weights* becomes the challenge of *tuning the thresholds*.

Therefore, we should focus on deriving attainable acceptance thresholds for the instance at hand. Estimating thresholds by drawing upon the characteristics of a given instance does not only result in instance-specific but also nurse-specific thresholds. These thresholds capture the individual wants and needs of each nurse, even as they change between rostering horizons.

Finding attainable thresholds relates to evaluating bounds on the slack needed for soft constraints. As an example, for a constraint k with an upper bound n_k, a corresponding acceptance threshold would be $n_k + \delta_k$ for some non-negative slack $\delta_k \geq 0$ intended to ensure that a feasible solution exists. Nonetheless, choosing this amount of slack is challenging, as the thresholds should neither be too tight (causing infeasibility) nor too loose (unnecessarily compromising the quality of the rosters). We note that using the bound n_k corresponds to how Burke et al. (2002) and Li et al. (2012) identified the ideal point for their compromise programming approaches.

To arrive at accurate acceptance thresholds, we should not only consider a single constraint. Instead, we should pragmatically choose relevant components of the problem, as interactions between constraints might affect the attainability of different thresholds. Considering thresholds for a component of the problem is directly linked to finding *minimal unsatisfiable subsets*, i.e., a set of constraints such that no feasible solution exists whereas all strict subsets have a feasible solution (Liffiton et al., 2016). Thus, we should estimate the thresholds for multiple constraints simultaneously in order to avoid unsatisfiable subsets caused by conflicting constraints.

For the NRP, we consider two types of targets, both individual targets specific to each nurse and targets for the ward as a whole. The type of target affects the amount of information we need to estimate the corresponding acceptance threshold accurately, and thus, influences the methods that we apply. We address these two types in the following sections, where Section 4.1.1 focuses on individual targets and Section 4.1.2 on ward targets. We let $\mathcal{T}$ denote the set of targets and $\mathcal{N}$ the set of nurses. For individual targets $t \in \mathcal{T}$, we let $AT_{t,n}$ denote the acceptance threshold for nurse $n \in \mathcal{N}$. Similarly, we let AT_t denote the threshold for ward target $t \in \mathcal{T}$.

4.1.1. Individual targets

This section introduces general techniques for estimating individual acceptance thresholds in an environment where the appropriate trade-offs (and thus thresholds) vary. When setting a threshold, we accommodate the nature of the problem by drawing upon the nurses' requests and present the following three techniques for estimating individual acceptance thresholds depending on the characteristics of the specific target:

1. Setting the acceptance threshold to zero and deactivating the target when conflicting with requests (either only hard requests or all requests).
2. Setting a general acceptance threshold (e.g., no slack) and increasing it as needed after measuring conflicts with requests (either only hard requests or all requests).

3. Solving individual optimization models to generate acceptance thresholds that will not result in unsatisfiable subsets.

Algorithms 1-2 present the first two techniques, respectively. For simplification, we consider conflicts with requests in general without specifying whether they are hard or soft. This simplification does not alter the structure of the algorithms. To evaluate whether a given target conflicts with the requests, we consider a relevant period of the rostering horizon. This period includes all the days that may impact the target. As an example, the relevant period for a given series target is a set of consecutive days corresponding to the length of the series. We analyze the requests put forth during this period to estimate whether combining them with the target would result in an unsatisfiable subset.

We let $\mathcal{D}$ denote the days of the rostering horizon and $v_{t,n} \geq 0$ denote the slack needed for target $t \in \mathcal{T}$ and nurse $n \in \mathcal{N}$. Furthermore, we let $s_{t,n} \geq 0$ denote an excess slack for target $t \in \mathcal{T}$ and nurse $n \in \mathcal{N}$. Most often, this excess is consistent between the nurses and is assessed based on known conflicts with other constraints (besides requests). Nonetheless, we may need the excess slack to be individual for targets where the nurses differ significantly due to different contracts (resulting in different constraints being active or the tightness of constraints being incomparable).

Algorithm 1 Deactivating target $t \in \mathcal{T}$ for nurse $n \in \mathcal{N}$ based on conflicts with requests.

```
for $d \in \mathcal{D}$ do
    if target conflicts with requests then
        Deactivate target on day $d \in \mathcal{D}$ for nurse $n \in \mathcal{N}$.
    end if
end for
```

In both algorithms, we loop through the days and identify conflicts. We note that we may replace days with other time units (e.g., weeks) as appropriate for the specific target. In Algorithm 1, we deactivate the target when conflicts occur. While we fully deactivate counter targets, we only deactivate the relevant set of days for series targets, i.e., those days that impose a conflict with the requests. With this deactivation, we can use an acceptance threshold of zero, preventing other occurrences. In Algorithm 2, we do not deactivate the target, but instead measure the slack that the identified conflicts will require (and store in $v_{t,n}$). We then adjust the acceptance threshold, drawing upon both the measured and excess slack.

We distinguish between *series* and *counter* constraints for personnel rostering. While counter constraints are evaluated by comparing the number of assignments against a given value, series constraints consider the occurrence of consecutive assignments. Smet et al. (2017) introduced the concepts of *local* and *global consistency* for the evaluation of series and counter constraints across multiple rostering horizons. For targets that require local consistency, we replace the set $\mathcal{D}$ with

Algorithm 2 Increasing the individual acceptance threshold corresponding to nurse $n \in \mathcal{N}$ and target $t \in \mathcal{T}$ based on conflicts with requests.

```
Initialize $AT_{t,n}$ as a general acceptance threshold for the target
Initialize measure for needed slack $v_{t,n} = \begin{cases} 0 & \text{if target considers a single horizon} \\ v^0_{t,n} & \text{if target requires global consistency} \end{cases}$
Set excess slack $s_{t,n}$ based on conflicts with other constraints
for $d \in \mathcal{D}$ do
    if target conflicts with requests then
        Increase $v_{t,n}$ based on conflict
    end if
end for
if $v_{t,n} + s_{t,n} > AT_{t,n}$ then
    $AT_{t,n} = v_{t,n} + s_{t,n}$
end if
```

$\mathcal{D}^{all}$, which includes the days of the current rostering horizon along with adjacent days required to ensure feasibility across the boundary of two rostering horizons. For targets that require global consistency, we let $v^0_{t,n}$ denote the slack for target $t \in \mathcal{T}$ and nurse $n \in \mathcal{N}$ that should be carried over from previous rostering horizons, by drawing upon the value of the target in that horizon. For example, this slack could be the value of the corresponding target at the end of the previous horizon.

Algorithms 1-2 are designed to estimate acceptance thresholds for targets where the main conflicts occur due to specific requests. The main difference between the two is that the former provides local control over trade-offs (by deactivating a target on a specific day) while the latter provides global control (referring to the entire rostering horizon).

The third technique is presented with Algorithm 3. This technique is designed to address targets that have a complicated relationship with other targets and hard constraints, where the main source of conflict is not clear. There, we generate individual optimization models that focus on generating feasible rosters for each nurse, without considering the staffing requirements of the ward. To avoid unsatisfiable subsets, the model includes all other individual targets for the specific nurse and assigns the nurse to shifts throughout the rostering horizon.

When building the optimization models in Algorithm 3, we incorporate acceptance thresholds for other targets as hard constraints. Thus, if Algorithm 3 is needed, it should only be employed after estimating thresholds for relevant targets with the first two algorithms. Moreover, inaccurate estimates for other thresholds may impact the estimates obtained with this algorithm, making it quite sensitive. For example, if the hard constraints are too tight, they either constitute an unsatisfiable subset (i.e., no feasible combination of thresholds exist) or Algorithm 3 will produce

Algorithm 3 Individual optimization models to estimate the acceptance threshold for nurse $n \in \mathcal{N}$ and target $t \in \mathcal{T}$.

```
1: Generate an optimization model for nurse $n \in \mathcal{N}$ including relevant targets with their acceptance thresholds as hard constraints
2: Minimize the slack for target $t \in \mathcal{T}$
3: Set $AT_{t,n}$ as the objective value (along with appropriate excess slack $s_{t,n}$).
```

a threshold that is too loose. Conversely, employing Algorithm 3 with unnecessarily loose hard constraints may result in a loose lower bound for the needed slack, leading to an unrealistic estimate of the corresponding threshold.

Figure 2 indicates how we choose the appropriate estimation technique based on the characteristics of a given target. Initially, we analyze whether we can identify a relevant period for evaluating a direct conflict between the target and the requests. If we cannot identify this period, then we must employ Algorithm 3. However, if we can identify the period, then we can employ either of the first two algorithms. In that case, the appropriate algorithm depends on the manager's perception of trade-offs, i.e., whether he requires local control over the trade-offs or allows for the flexibility of global control.

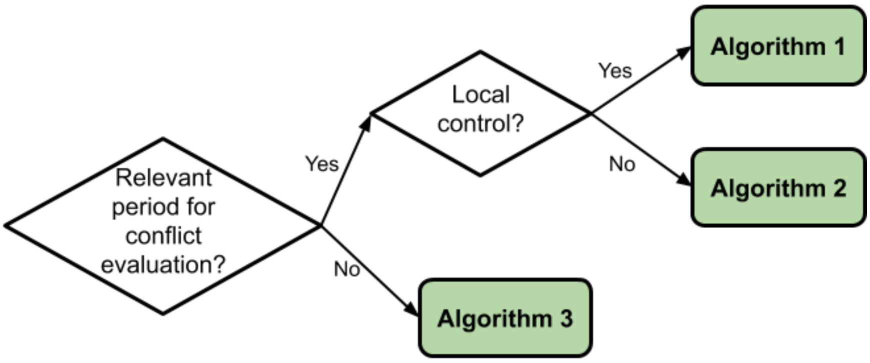


Figure 2: Determining the appropriate algorithm for estimating the acceptance threshold for a specific target.

4.1.2. Ward targets

The techniques presented in the previous section explore a confined part of the problem, namely considering only a single nurse. Some ward targets belong to other confined parts of the problem (e.g., a single day of the rostering horizon), and for those, we can develop similar techniques as for individual targets. Nonetheless, ward targets may also be broader and for those cases, we will not find accurate acceptance thresholds by only exploring a restricted part of the problem. Thus, we have to rely on manually defined acceptance thresholds when the relationship with other targets and constraints becomes complicated. The silver lining is that most nurse rostering targets are individual and can be addressed with the previously presented algorithms.

In nurse rostering, we will always see at least one ward target, namely corresponding to the coverage constraints. To set the corresponding acceptance threshold, we analyze the available resources (i.e., nurses). For each day of the rostering horizon, we assess whether the available resources are sufficient by analyzing the nurses needed for each shift and skill combination separately along with the total number of nurses needed for the day. Furthermore, we consider the accumulated number of nurses needed for a given shift and skill combination throughout the rostering horizon, to assess whether we can satisfy it without exceeding the nurses' contractual hours. For further details on the procedure, we refer the reader to Böðvarsdóttir et al. (2019c).

4.2. Lexicographic goal programming

The concepts *targets* and *acceptance thresholds* fit naturally in the framework for goal programming, as for every target, we should stay within the corresponding acceptance threshold if feasible, and if not, then we should minimize the violation of the target. Thus, the acceptance thresholds correspond to *target levels* in the goal programming framework. We introduce a lexicographic goal programming (LGP) approach for satisfying the acceptance thresholds, as shown with Algorithm 4, where the first step requires the practitioners to define and prioritize their targets.

Algorithm 4 Lexicographic goal programming with acceptance thresholds.

```
Identify the prioritized list of targets $\mathcal{T}$.
Set acceptance thresholds for all targets.
for $t \in \mathcal{T}$ (in prioritized order) do
    Apply optimization to minimize the deviation for target $t$ from its acceptance threshold
    if Objective value $> 0$ then
        Increase the acceptance threshold with the objective value
    end if
    Enforce acceptance threshold as hard constraint
end for
```

For each target, we use an algorithm to minimize the violation of the target. If the objective value is zero, then we have one or more feasible solutions satisfying the acceptance threshold and can safely introduce it as a hard constraint. If the objective value is greater, then we must adjust the acceptance threshold accordingly before introducing it as a hard constraint. If we need to adjust a threshold, the appropriate increase comes directly from the value of the objective function. When solving the goal program for the last target we obtain an optimal roster w.r.t. the priorities and the original acceptance thresholds.

With the lexicographic approach, we iteratively introduce the acceptance thresholds as hard constraints, but only after confirming their feasibility. Using distinct priorities provides a clear message when target violations occur, namely that the threshold cannot be met without sacrificing higher priority targets. Furthermore, we iteratively cut some areas away from the solution space, resulting in a gradual reduction of its size. By applying accurate thresholds, we ensure that we do not cut away promising areas, but instead, remove those areas where the solver would waste time on exploring unpromising solutions with inappropriate trade-offs.

Individual targets have multiple acceptance thresholds, one corresponding to each nurse. We do not enforce a hard constraint for each specific threshold (i.e., nurse), as the resulting restrictions to the solution space would be unnecessarily tight. Instead, we enforce hard constraints based on overall measurements of the violations that allow for some flexibility in moving violations between nurses

For each target $t \in \mathcal{T}$, we consider the sum of the violations for all nurses. We formulate the corresponding goal program with objective (1) and constraints (2), where $\mathcal{N}$ is the set of nurses and $AT_{t,n}$ is the acceptance threshold for nurse $n \in \mathcal{N}$ corresponding to target $t \in \mathcal{T}$. Furthermore, $x_{t,n} \geq 0$ is the value of the constraint corresponding to target $t \in \mathcal{T}$ for nurse $n \in \mathcal{N}$ and $\gamma_{t,n} \geq 0$ is an auxiliary variable denoting the magnitude of the target violation for nurse $n \in \mathcal{N}$.

$$\min \sum_{n \in \mathcal{N}} \gamma_{t,n} \tag{1}$$

$$\text{s.t. } x_{t,n} - \gamma_{t,n} \leq AT_{t,n} \quad \forall n \in \mathcal{N} \tag{2}$$

In addition, the manager may require potential violations to be balanced between nurses to promote fairness for some targets. For those targets, we also consider the maximum individual violation. We formulate the goal program for minimizing the maximum violation of any acceptance threshold for target $t \in \mathcal{T}$ with equations (3)-(5), where ϕ_t is the maximum violation of any threshold for the target, and the remainder of the notation is as described previously. For these targets, we enforce the resulting cut before considering the sum of all violations. Therefore, requiring a balance between the nurses may lead to a greater total violation.

$$\min \phi_t \tag{3}$$

$$\text{s.t. } x_{t,n} - \gamma_{t,n} \leq AT_{t,n} \quad \forall n \in \mathcal{N} \tag{4}$$

$$\gamma_{t,n} \leq \phi_t \quad \forall n \in \mathcal{N} \tag{5}$$

LGP can be employed to avoid any direct comparisons between different criteria. While this attribute may be valuable, it also imposes a challenge as it does not allow a small violation for a high priority target to replace a significant violation for one of a lower priority (Branke et al., 2008). However, we control how to solve conflicts between targets when estimating their acceptance thresholds and implicitly incorporate desirable trade-offs in the thresholds. Thus, we account for trade-offs between targets with different priorities before the lexicographic step. We hypothesize that using general acceptance thresholds (i.e., aiming for zero or minimal slack) in the lexicographic framework will provide substantially worse results, compared to estimated thresholds. We address this hypothesis when validating the methodology in Section 5.

Goal programming includes several variants and while the lexicographic variant was common several decades ago, its use has substantially declined today. Nonetheless, nothing indicates that this decline relates to the quality of the approach. On the contrary, Jones and Tamiz (2010) state that an investigation of the goal programming literature revealed that in many cases the "problems" that arise are not due to any flaw in the technique itself but more due to a lack of good goal programming practice. For example, one needs to show great care when setting the appropriate thresholds (or target levels), as no trade-offs are made between targets of different priorities. Therefore, employing estimated acceptance thresholds that implicitly incorporate trade-offs fits well with the lexicographic variant.

5. Case study

This section describes a case study covering two wards in two Danish hospitals. The formulation, which was developed in close collaboration with practitioners, was first introduced as a mixed integer programming (MIP) model with a weighted sum objective function (Böðvarsdóttir et al., 2019a). The formulation considers numerous constraints, both individual and for the ward as a whole. Furthermore, each nurse may have a number of specific requests, both hard and soft where the soft requests are further categorized into high or low priority. Overall, the formulation categorizes as $ASBCI|RVNO|PLXM$ using the $\alpha|\beta|\gamma$-notation by De Causmaecker and Vanden Berghe (2011) and as it includes multiple constraints on consecutive days, it is NP-hard (Smet et al., 2016).

The wards considered in this case have a few characteristics that distinguish them from the general NRP. We describe these differences as follows: First, we consider multiple shifts and define three *shift blocks* (day, evening, and night) where each block may include multiple shifts. This shift structure is more complex than in most NRP formulations.

Second, to obey the Danish legislation, the formulation of days off differs from other formulations. According to the legislation, all employees are entitled to *protected days off* (or *PF*) which are weekly days off. Compared to the general EU legislation, PF days need to satisfy tighter constraints, for example, a higher number of consecutive hours off. On average, we should assign two PF days per week and distribute them evenly, preferably not having more than six days between two contiguous PF days.

Third, some nurses (referred to as *trainees*) are undertaking an educational program. To fulfill the requirements, they should be assigned to some *chaperoning shifts* together with their *chaperone* on each roster. These shifts are solely spent on planning and evaluating the progress of the education and neither nurse participates in patient care simultaneously.

Due to the rostering culture in Denmark, the work weekends for all nurses are pre-determined a long time in advance, allowing the nurses flexibility in planning their time off. Even though the work weekends have been determined, assigning each nurse to a specific shift remains a part of the rostering task (except when they have hard requests). When doing so, we strive to meet the nurses' requests and general weekend preferences.

In the experiments, we use twelve instances for two wards, Ward A and Ward B, available from Böðvarsdóttir et al. (2019b). We consider seven instances, A01-A07, for Ward A and five instances, B01-B05, for Ward B. For each ward, the main difference between the instances lies in the real-life variability from one rostering horizon to the next, where an example could be different nurses' requests. Table 1 presents information on the instances, including the length of the rostering horizon in days and the number of nurses in the wards. Additionally, the number of binary assignment variables describes how we can assign nurses to shifts on different days, after taking hard requests into account. Finally, we present the number of hard and soft requests.

Table 1: Summary statistics for the two wards.

	Ward A			Ward B		
	min	average	max	min	average	max
Days	28	28.0	28	28	28.0	28
Nurses	45	46.0	47	30	34.4	40
Binary assignment variables	4,504	4,766.4	5,158	3,565	3,805.0	4,077
Hard requests	203	273.9	435	82	172.6	365
Soft requests	434	498.6	529	9	111.0	340

In this paper, we reformulate the problem from Böðvarsdóttir et al. (2019a) by using targets (see Section 5.1). In the weighted sum formulation, the relationship between the constraints is complicated and many support or conflict with one another. In some cases, multiple constraints act together to achieve a single target and in other cases, constraints work against each other to achieve an acceptable compromise. We present further details on the reformulation in Appendix A.

5.1. Targets

Table 2 introduces the targets considered, along with their priorities. These targets were identified in collaboration with practitioners, matching their criteria when assessing the quality of rosters.

When evaluating this quality, the nurses' requests often override the individual targets. This is incorporated by adjusting the acceptance thresholds or deactivating the corresponding target (using techniques from Section 4.1.1). For most targets, the requests *explicitly* indicate whether that target should be overridden. For example, a request for weekend work that does not match the weekend preference. The target *Isolated Workdays* is an exception, as the need to override

Table 2: Description of targets.

Target	Priority	Description
Float Nurses	1	Limit the number of float nurses.
High Priority Requests	2	Fulfill a minimum percentage of high priority requests for all nurses.
Weekend Preferences	3	We should assign nurses according to their weekend preferences, unless explicitly requested.
Distance Between PF	4	A maximum of 6 days between two contiguous PF days for all nurses, unless explicitly requested.
No Multiple Night Sequences	5	Limit work weeks (from Monday to Sunday) where the nurses work multiple sequences of night shifts (i.e., night shifts with days off in between) for most nurses, unless explicitly requested.
Overtime	6	Limit positive deviation (in hours) from contractual hours for all nurses, unless explicitly requested.
Undertime	7	Limit negative deviation (in hours) from contractual hours for all nurses, unless conflicting with hard requests. We set the threshold rather loose, as this target conflicts with multiple other targets promoting healthy work schedules.
Total Requests	8	Fulfill a minimum percentage of requests (independent of their priority) for all nurses.
Shift Blocks In Week	9	Prevent all nurses from working three shift blocks during one workweek (from Monday to Sunday), unless conflicting with hard requests.
Chaperoning Shift	10	Trainees should get a minimum of chaperoning shifts together with their chaperone. We may reduce this minimum if hard requests for the trainee and the chaperone result in few chaperoning shifts being feasible during the rostering horizon.
Balance Excess Staffing	11	Limit the maximum difference in the number of staff assigned to the same coverage constraint on different days.
Monday Off After Work Weekend	12	Restrict nurses from working on the Monday following their work weekend for all nurses, unless explicitly requested.
Isolated Workdays	13	Limit isolated workdays for all nurses, unless requested, either explicitly or implicitly in combination with other constraints.

the target may be *implicitly* implied by the requests instead of explicitly stated. For example, accommodating requests for the night shift on a Monday and the evening shift on a Wednesday requires Tuesday to be a day off (so as to ensure the legally required resting period). If the previous

Table 3: Estimation of acceptance thresholds and the incorporation of targets in the lexicographic framework.

Target	Type	General threshold	Bound	Estimation technique	Balance violations
Float Nurses	Ward	0	Upper	Section 4.1.2	-
High Priority Requests	Individual	-	Lower	Algorithm 3	Yes
Weekend Preferences	Individual	-	Lower	Algorithm 3	Yes
Distance Between PF	Individual	0	Upper	Algorithm 1	No
No Multiple Night Sequences	Individual	0	Upper	Algorithm 1	No
Overtime	Individual	6	Upper	Algorithm 2	No
Undertime	Individual	6	Upper	Algorithm 2$\star$	No
Total Requests	Individual	-	Lower	Algorithm 3	Yes
Shift Blocks In Week	Individual	0	Upper	Algorithm 1	No
Chaperoning Shift	Individual	2	Lower	Algorithm 2$\star$ (extended)	No
Balance Excess Staffing	Ward	-	Upper	Set by ward manager	-
Monday Off After Work Weekend	Individual	0	Upper	Algorithm 1	No
Isolated Workdays	Individual	0	Upper	Algorithm 2	No

weekend was a weekend off, then these two requests implicitly require Monday to be an isolated workday. Therefore, identifying a conflict between this target and the requests requires a broader view than only looking at what is explicitly requested.

Table 3 categorizes the targets into individual and ward targets and describes the corresponding acceptance thresholds. For each target, we list a *general* threshold (if applicable) along with the estimation technique applied, where a star ($\star$) indicates that the estimation excludes conflicts with soft requests, as discussed below. These general thresholds represent rules of thumb that practitioners use, e.g., aiming for no violations. We emphasize that the method is robust when faced with unrealistic general thresholds, as the estimation techniques presented in Section 4.1 adjust them as needed. For individual targets, we also present whether we try to balance the violations between the nurses when incorporating the targets into the goal programming framework.

We note that a general threshold is not applicable to all targets. As the number of requests along with assigned work weekends varies, the corresponding acceptance thresholds should also differ between the nurses. We set the threshold as a given percentage of the possible fulfillment and we investigate different levels of request fulfillment in the experiments (see Table 4 in Section 5.2). The best possible fulfillment is not necessarily equal to the number of requests, as the requests may conflict with other targets that should not be relaxed based on soft requests (e.g., *Undertime*). Thus, we start by accurately estimating all other individual thresholds (using Algorithms 1-2) before enforcing them as hard constraints and employing Algorithm 3 to estimate the best possible fulfillment.

Out of thirteen targets, only two categorize as ward targets. We believe that this distribution is similar for many nurse rostering problems, as the focus is on generating good individual rosters. For *Balance Excess Staffing*, we do not estimate an attainable acceptance threshold but instead, use a threshold set by the ward managers. The main reasons are both the scope of the target (relating to the entire rostering horizon along with all the nurses) and the complicated relationships (both supportive and conflicting) with other constraints and targets.

Not all individual targets are relevant for all nurses, e.g., the *Chaperoning Shift* only applies to trainees. As the feasibility of the corresponding threshold depends both on the trainee and the chaperone, we extend Algorithm 2 to include both the trainee and the corresponding chaperone, as opposed to only the trainee. In that manner, we measure the number of days where the shift is simultaneously feasible for both nurses and adjust the threshold relative to the infeasible days.

5.2. Experiments

To validate the methodology, we investigate the impact of implicitly incorporating trade-offs in the acceptance thresholds. We considered the following two categories of experiments:

1. An LGP approach employing general acceptance thresholds that minimize violations and do not include any slack or deactivation of constraints due to conflict.
2. An LGP approach employing instance-specific acceptance thresholds estimated with the techniques presented in Section 4.1.

In each category, we performed three experiments based on different settings for the fulfillment of requests and weekend preferences (see Table 4). These settings correspond to a different slack from the requests put forth (first category) or the estimated best possible request fulfillment (second category). While Setting A excludes these targets completely, Setting C forbids any slack for them. Excluding the targets provides a basis for comparison by indicating whether eventual violations are completely inevitable or because of too ambitious request fulfillment. However, these targets would never be excluded in practice.

Table 4: Acceptance thresholds as a percentage of the best possible fulfilment for *High Priority Requests* (HR), *Weekend Preferences* (WP) and *Total Requests* (TR).

Settings	HR [%]	WP [%]	TR [%]
A	0	0	0
B	100	100	80
C	100	100	100

In addition to minimizing the violations of the targets, all experiments include an additional step in the end for maximizing the overall fulfillment of requests (independent of priority). Due to the thresholds, the final step does not affect any experiment with Setting C and only affects low priority requests for Setting B.

The first category employs manual rules of thumb that do not consider the characteristics of individual instances. On the contrary, the second category draws upon these instance-specific characteristics when automatically setting acceptance thresholds. Comparing these two categories yields the value of setting realistic expectations when employing lexicographic solution approaches.

We have implemented the approach in Python 3.6.5 and use Gurobi 8.0 to solve the optimization problem in each step of the LGP. We ran the experiments on a 64-bit Windows 7 with 12GB RAM and an Intel Core i5-4570 CPU @3.20GHz.

6. Results

Sections 6.1-6.2 provide the results for the two different categories of experiments. For additional details, we refer the reader to Appendix B. Section 6.3 compares the results of these two categories, addressing the importance of realistic expectations, and Section 6.4 presents a general discussion.

6.1. General acceptance thresholds

This section provides the results for the first category, namely using general thresholds in the LGP framework. Tables 5-7 present information on these general acceptance thresholds and for nurse-specific targets, the numbers we present are accumulated for all nurses. Table 5 presents the number of active constraints for the targets where we would employ the deactivation technique if using the instance-specific information. Table 6 shows the number of requests put forth by the nurses along with the number of weekend preferences matching the pre-determined work weekends. Finally, Table 7 shows the acceptance thresholds for the remaining targets.

Table 5: Number of active constraints without any deactivation based on conflicts. A dash (-) indicates that the target is inactive for the specific instance.

Target	A01	A02	A03	A04	A05	A06	A07	B01	B02	B03	B04	B05
Distance Between PF	1069	1066	1115	1112	1089	1094	1101	814	774	743	926	964
No Multiple Night Sequences	168	168	180	180	180	176	168	-	-	-	-	156
Shift Blocks In Week	138	144	132	117	145	145	140	132	124	57	24	29
Monday Off After Work Weekend	53	51	53	54	57	52	51	61	58	50	62	52

As we use manually chosen acceptance thresholds, we do not implicitly incorporate any trade-offs between constraints before employing the lexicographic solution approach. Thus, higher priority

Table 6: Number of requests and preferences put forth by the nurses. These levels do not correspond to the acceptance thresholds, as the percentage of the best possible differs between Settings A-C. A dash (-) indicates that the target is inactive for the specific instance.

Target	A01	A02	A03	A04	A05	A06	A07	B01	B02	B03	B04	B05
High Priority Requests	101	165	159	138	136	168	150	7	7	8	12	6
Weekend Preferences	-	-	-	-	-	-	-	95	8	82	94	94
Total Requests	514	529	519	434	457	534	503	22	340	9	87	97

Table 7: General acceptance thresholds. A dash (-) indicates that the target is inactive for the specific instance. The individual threshold is the same across all nurses, and the difference between instances relates to a different number of nurses.

Target	A01	A02	A03	A04	A05	A06	A07	B01	B02	B03	B04	B05
Float Nurses	0	0	0	0	0	0	0	0	0	0	0	0
Overtime	270	270	282	282	276	276	276	198	186	180	228	240
Undertime	270	270	282	282	276	276	276	198	186	180	228	240
Chaperoning Shift	12	12	8	8	8	6	8	-	-	-	-	-
Balance Excess Staffing	3	3	3	3	3	3	3	5	5	5	5	5
Isolated Workdays	0	0	0	0	0	0	0	0	0	0	0	0

targets become infinitely more important than those of a lower priority, which may not lead to the appropriate trade-offs when conflicts occur.

Tables 8-9 present the target violations for Experiments 1A-1B, respectively. In addition, Appendix B.1 presents a similar table for Experiment 1C. For compactness, the tables exclude targets that do not have any violations. Thus, having only a few rows or having a large part of the table blank is an indication of good results, i.e., few target violations.

Even when excluding the targets for requests and weekend preferences (Setting A), each instance has violations for 5-7 targets, which is more than half of the active targets. Furthermore, some of these violations are quite extensive. When introducing acceptance thresholds for requests and weekend preferences (Setting B), the violations increase, especially for lower priority targets, and the same trend continues as the acceptance thresholds become tighter (Setting C).

Drawing on these results, we conclude that the general acceptance thresholds are unrealistically tight, and enforcing them as hard constraints in the lexicographic framework results in obscure violations for low priority targets.

Table 8: Target violations for Experiment 1A.

Target	A01	A02	A03	A04	A05	A06	A07	B01	B02	B03	B04	B05
Float Nurses					2			6	6	12	12	26
Distance Between PF	8	16	23	40	11	26	13	14	10	5	6	10
No Multiple Night Sequences												2
Overtime			10.86		16.46	4.85	11.11			15.35	26.57	1.01
Undertime	13.43	6.21	18.48	96.11		220.81		2.02	111.51	893.53	265.25	49.70
Chaperoning Shift	1	1	1	3	1	1	4					
Monday Off After Work Weekend	5	2	5	5	4	2	3	4	13	8	1	2
Isolated Workdays	5	4	8	35	16	6	2	2	23		3	3

Table 9: Target violations for Experiment 1B. For requests and weekend preferences, the former number presents the accumulated violation across all nurses, and the number in brackets is the maximum individual violation.

Target	A01	A02	A03	A04	A05	A06	A07	B01	B02	B03	B04	B05
Float Nurses					2			6	6	12	12	26
High Priority Requests		2 (1)		10 (4)	3 (2)	2 (2)	2 (1)	2 (1)	2 (2)			
Weekend Preferences								6 (2)	4 (2)	14 (4)	10 (4)	6 (2)
Distance Between PF	8	16	28	42	13	26	14	14	10	5	8	14
No Multiple Night Sequences						1	1					1
Overtime			10.86		24.44	4.85	11.11			15.35	26.57	1.01
Undertime	16.67	6.21	18.73	98.99		243.69	9.09	2.02	111.51	902.02	292.73	33.54
Total Requests	4 (1)	3 (1)	3 (1)	13 (2)	8 (3)	2 (1)	14 (4)	2 (1)	24 (4)		2 (1)	1 (1)
Chaperoning Shift	3	3	3	5	1	2	4					
Monday Off After Work Weekend	8	6	10	10	7	6	5	5	15	8	1	3
Isolated Workdays	18	25	22	52	53	24	22	2	37	1	5	3

6.2. Estimated acceptance thresholds

This section provides the results for the second category, namely considering estimated acceptance thresholds in the LGP framework. Tables 10-12 present information on the acceptance thresholds that we work with for each instance and each target. These thresholds vary substantially, both between nurses and between rostering horizons. We note that the structure of the tables matches Tables 5-7 for the first category of experiments.

Tables 13-15 present the target violations for Experiments 2A-2C, respectively. As before, fewer

Table 10: Number of active constraints after employing the deactivation technique from Algorithm 1. A dash (-) indicates that the target is inactive for the specific instance.

Target	A01	A02	A03	A04	A05	A06	A07	B01	B02	B03	B04	B05
Distance Between PF	1049	1023	1067	1043	1039	1049	1068	798	719	736	915	954
No Multiple Night Sequences	166	166	177	178	178	173	165	-	-	-	-	156
Shift Blocks In Week	138	144	131	117	145	145	140	132	124	57	24	29
Monday Off After Work Weekend	22	15	18	14	16	28	19	49	25	45	57	48

Table 11: Best possible request and preference fulfillment according to nurse-specific models (Algorithm 3). These levels do not correspond to the estimated acceptance thresholds, as the percentage of the best possible differs between the experiments. A dash (-) indicates that the target is inactive for the specific instance.

Target	A01	A02	A03	A04	A05	A06	A07	B01	B02	B03	B04	B05
High Priority Requests	100	163	159	126	132	162	146	5	5	8	12	6
Weekend Preferences	-	-	-	-	-	-	-	95	8	78	94	94
Total Requests	494	505	492	385	427	505	464	20	307	9	87	95

Table 12: Acceptance thresholds based on estimation techniques from Algorithms 2 and Section 4.1.2. The threshold for *Balance Excess Staffing* is set by the ward manager. A dash (-) indicates that the target is inactive for the specific instance.

Target	A01	A02	A03	A04	A05	A06	A07	B01	B02	B03	B04	B05
Float Nurses	0	0	0	0	2	0	0	6	6	12	12	26
Overtime	270	270	294	302	312	286	292	198	189	197	247	242
Undertime	341	298	366	405	289	779	276	672	604	2299	846	701
Chaperoning Shift	6	8	4	3	6	3	3	-	-	-	-	-
Balance Excess Staffing	3	3	3	3	3	3	3	5	5	5	5	5
Isolated Workdays	50	63	58	83	66	57	44	24	70	23	29	26

rows or blank spaces indicate few target violations and thus good results. Furthermore, we have excluded instances without violations, and thus fewer columns also indicate good results. For Setting A, we see from none to two target violations per instance. The magnitude of these violations is generally minimal, and analyzing the rosters used in the wards shows that the managers are willing to accept these kinds of violations.

Table 13: Target violations for Experiment 2A.

Target	A01	B01	B02	B03	B04	B05
No Multiple Night Sequences						2
Overtime					7.56	
Undertime	3.64					
Monday Off After Work Weekend		2	1	8	1	1

Table 14: Target violations for Experiment 2B. For requests and weekend preferences, the former number presents the accumulated violation across all nurses, and the number in brackets is the maximum individual violation.

Target	A01	A07	B01	B02	B03	B04	B05
High Priority Requests		1 (1)					
Weekend Preferences			4 (2)	4 (2)	10 (2)	10 (4)	6 (2)
Distance Between PF							4
No Multiple Night Sequences							1
Overtime						7.56	
Undertime	3.64						
Total Requests				6 (2)		2 (1)	1 (1)
Monday Off After Work Weekend			2	1	8	1	2

Table 15: Target violations for Experiment 2C. For requests and weekend preferences, the former number presents the accumulated violation across all nurses, and the number in brackets is the maximum individual violation.

Target	A01	A02	A03	A04	A05	A06	A07	B01	B02	B03	B04	B05
High Priority Requests							1 (1)					
Weekend Preferences								4 (2)	4 (2)	10 (2)	10 (4)	6 (2)
Distance Between PF												4
No Multiple Night Sequences												1
Overtime											7.56	
Undertime	3.64											
Total Requests	10 (1)	6 (1)	9 (2)	3 (1)	10 (2)	2 (1)	5 (1)		30 (4)		5 (3)	3 (3)
Chaperoning Shift	2	1		1	1							
Monday Off After Work Weekend	3		1					2	1	8	1	2
Isolated Workdays	29	7	8	4	24	12	14				1	

Setting B introduces acceptance thresholds for requests and weekend preferences. Although we see some violations of those targets, their inclusion rarely has a negative effect on lower priority targets. Overall, the target violations are very limited, both in the number of targets violated and in the magnitude of each violation. Nevertheless, when forbidding any slack for these acceptance thresholds (Setting C), we experience a substantial negative effect on lower priority targets, e.g., *Isolated Workdays*, indicating that some slack is necessary for the thresholds to be realistic.

Overall, the results of these experiments are very promising and display a large potential for improving lexicographic solution approaches by employing instance-specific information. In addition to reaching a far majority of the targets, the rosters also fulfill a majority of the requests, with the average being 97% of the estimated best possible fulfillment, which may be infeasible.

6.3. Comparison

The quality of the results for the two categories differs significantly. While employing general acceptance thresholds leads to substantial violations, the estimated thresholds result only in minimal violations. The difference is not only clear by the number of targets violated but also by the magnitude of each violation. This matches the hypothesis presented in Section 4.2, namely that using general thresholds would lead to worse results than estimated thresholds. We acknowledge that this type of quantitative comparison may be biased, as the estimated thresholds allow for more relaxation than the general thresholds. Nonetheless, performing a completely non-biased comparison between the two is impossible, as we need the estimation techniques to determine appropriate relaxations.

When analyzing the relation between the priority of a target and the corresponding violation, we see that for general thresholds the violations are more excessive for lower priority targets. However, the violations for estimated thresholds are less related to the priorities of the targets (except when enforcing too tight thresholds with Setting C). These results link back to Jones and Tamiz (2010)'s statement, that poor performance of lexicographic goal programming is not due to a flaw in the technique, but to a lack of good practices (in this case, working with unrealistic target levels).

In addition to a comparison based on the quality of the experiments' results, we also compared their running time, as presented in Tables 16-17. The time is divided into Setup and Solve, where Setup refers to the time spent reading in and pre-processing the data, estimating the acceptance thresholds, and building the LGP model, while Solve refers to the LGP phase.

By adding the step of estimating acceptance thresholds, the Setup time for the second category increases by 35% when compared to the first. Nonetheless, we must emphasize that even when estimating the thresholds the Setup only takes around 30 seconds on average, so the increase is insignificant. As the general thresholds do not implicitly incorporate trade-offs, the subsequent hard constraints become so tight that it is hard to generate a feasible solution. This is evident

Table 16: The running time in seconds for the first category of experiments, divided into setup (for preprocessing the data, estimating thresholds and building the LGP model) and solve (for solving the LGP model).

Instance	Experiment 1A			Experiment 1B			Experiment 1C		
	Setup	Solve	Total	Setup	Solve	Total	Setup	Solve	Total
A01	27.06	77.73	104.79	26.24	2004.44	2030.68	26.51	174.99	201.50
A02	26.68	122.57	149.25	27.11	279.84	306.95	26.52	94.96	121.48
A03	26.47	69.08	95.55	26.30	301.98	328.28	26.04	166.83	192.87
A04	26.37	72.98	99.35	26.39	243.60	269.99	26.79	62.64	89.43
A05	26.15	855.13	881.28	26.81	298.98	325.79	26.27	97.65	123.92
A06	30.95	813.34	844.29	30.60	195.89	226.49	31.35	74.03	105.38
A07	22.37	188.42	210.79	22.42	1,249.28	1,271.70	22.61	110.84	133.45
B01	19.29	30.18	49.47	19.71	21.85	41.56	19.40	23.75	43.15
B02	18.29	15.56	33.85	19.42	89.48	108.90	18.91	30.55	49.46
B03	14.26	20.93	35.19	14.80	20.11	34.91	14.64	20.52	35.16
B04	16.47	197.36	213.83	16.27	185.15	201.42	16.42	188.99	205.41
B05	16.97	40.57	57.54	16.20	67.51	83.71	16.64	50.01	66.65

Table 17: The running time in seconds for the second category of experiments, divided into setup (for preprocessing the data, estimating thresholds and building the LGP model) and solve (for solving the LGP model).

Instance	Experiment 2A			Experiment 2B			Experiment 2C		
	Setup	Solve	Total	Setup	Solve	Total	Setup	Solve	Total
A01	35.26	70.82	106.08	37.34	95.53	132.87	35.12	99.15	134.27
A02	36.15	57.44	93.59	35.80	96.30	132.10	36.31	110.06	146.37
A03	34.75	65.70	100.45	35.29	61.36	96.65	35.05	204.34	239.39
A04	35.58	32.48	68.06	36.07	30.38	66.45	35.27	36.52	71.79
A05	34.59	128.94	163.53	35.63	134.23	169.86	35.51	105.12	140.63
A06	41.41	48.82	90.23	41.26	46.91	88.17	40.00	30.79	70.79
A07	32.05	70.07	102.12	30.62	102.42	133.04	31.74	91.39	123.13
B01	24.45	20.36	44.81	24.62	20.53	45.15	24.91	19.74	44.65
B02	23.90	12.10	36.00	23.07	22.44	45.51	23.77	17.41	41.18
B03	18.68	16.31	34.99	20.23	20.76	40.99	19.65	18.69	38.34
B04	23.49	37.69	61.18	23.86	52.56	76.42	24.04	51.33	75.37
B05	24.36	31.49	55.85	24.78	54.42	79.20	23.84	53.32	77.16

by a substantial difference in the Solve time, where Experiment 2B finds results in half the time compared to Experiment 1B.

6.4. Discussion

The results display great potential for using alternative evaluation functions that promote controlled trade-offs to solve nurse rostering problems. The estimated thresholds implicitly incorporate the appropriate trade-offs for different constraints, resulting in rosters with an acceptable slack for these constraints and without (or with only minimal) violations. Furthermore, incorporating the trade-offs beforehand lowers the impact of the lexicographic ordering, and minimal violations indicate that the method is insensitive to the priorities chosen for the targets.

The estimated acceptance thresholds displayed a huge variability for different rostering horizons within the same wards, highlighting the real characteristics of the problem. By comparing estimated and general thresholds, we see a positive correlation between the number of requests and the change from the general threshold to the estimated one. These results are intuitive, as more requests mean that the nurses are clearer in communicating their wants, and thus, we can be more confident in relaxing general requirements. Apart from this general observation, we do not see any trends in the estimated acceptance thresholds for different rostering horizons that could help with foreseeing the appropriate slack for different constraints. Overall, we can safely conclude that setting these thresholds is not trivial and we cannot expect practitioners to do so manually.

Even though targets and acceptance thresholds can guide the search for solutions into promising areas, they are only beneficial if the acceptance thresholds are set accurately. Working with general thresholds, as opposed to instance-specific estimates, caused troubles when solving the problem lexicographically. Basing trade-offs solely on the priorities of different targets resulted in significant violations, especially for lower priority targets. Furthermore, we identified numerous non-desired trade-offs and a lack of fairness, for example, where one nurse received substantial violations caused by another nurse's fulfilled requests. Thus, a prerequisite for using the lexicographic method is to have estimated accurate thresholds that incorporate appropriate trade-offs.

Individual targets have nurse-specific thresholds, meaning that we need to estimate numerous different acceptance thresholds for each such target. Relating this to reference point methods (Branke et al., 2008, chap. 2), we see that a much greater level of detail can be included when automatically estimating thresholds compared to what can be expected from manually adjusted reference points. Even though the level of detail would be substantially reduced, setting and updating the reference point for each horizon would be a very challenging and time-consuming task. A large benefit of interactive methods is that the practitioners acquire a better understanding of the problem considered and can better justify trade-offs in the result. We argue that employing automatically estimated acceptance thresholds along with a lexicographic framework also gives a clear justification for the final solution, without requiring the human effort.

7. Conclusion

Real-world nurse rostering problems present a great variety of instances, where the appropriate trade-offs between constraints also differ. The majority of nurse rostering research uses a weighted sum evaluation function where the weights attempt to impose the appropriate trade-offs. Unfortunately, setting weights remains challenging due to the numerous conflicting aspects.

This paper introduced a new multi-objective approach that automatically estimates acceptance thresholds, implicitly incorporating appropriate trade-offs between nurse rostering constraints. The methodology addresses these thresholds with a lexicographic goal programming formulation and requires a MIP solver to solve each level of the formulation. These thresholds not only guide the search for solutions towards appropriate compromise solutions, but also capture and accommodate the unique needs and preferences of individual nurses. Nevertheless, the subjective appreciation of a given roster remains difficult to quantify and we acknowledge that various disregarded factors may impact the nurses' opinions.

Compared to other NRP approaches, this methodology requires minimal information from the users. This information consists of a list of targets along with their priorities and general acceptance thresholds. Both handles required to control each target (priority and acceptance thresholds) correspond directly to the handles (and units) that nurse rostering practitioners employ when manually generating rosters. We emphasize that the methodology is not sensitive to inaccuracy in the general thresholds. This property generally makes the methodology superior to many other a priori approaches, as reliable knowledge regarding the final solution is often lacking.

Automatically incorporating instance-specific trade-offs not only relieves the cognitive burden so often faced by managers, but also yields more realistic acceptance thresholds. By drawing upon these thresholds, we can confine the search for solutions to promising areas without being too restrictive. As the results indicate, this leads to better solutions within less computational time.

We acknowledge that comparing this methodology to approaches employing a weighted sum objective may be difficult. Due to the difference in the evaluation function, a comparison based solely on the quality of the result can be biased. Furthermore, the personnel rosters obtained with a weighted sum objective may depend substantially on the objective weights. Thus, the methodology should not only be evaluated based on the quality of the rosters, but also the effort needed to obtain them. Currently, conducting such a comparison requires significant involvement from the end-user in order to accurately evaluate the perceived effort needed. Developing the means to compare different methodologies based on multiple aspects is an intriguing challenge that could have immense value when introducing and validating alternative methodologies in the future.

Both the experiments conducted and user feedback confirm that the proposed methodology has potential for wider usage than the NRP, where the key lies in defining techniques for estimating acceptance thresholds. While the estimation algorithms we have presented are restricted to nurse rostering (or employee scheduling) problems, similar techniques may be generated for other problems by identifying appropriate trade-offs. The methodology could also prove beneficial for other problems where a direct comparison between different constraints is difficult, e.g., due to an intricate structure or different measurement units.

Finally, the use of estimated acceptance thresholds is not restricted to lexicographic goal programming. Indeed, the methodology for obtaining a realistic estimate of the roster structure is valuable for practitioners that have difficulty with foreseeing the final outcome. The estimated thresholds can thus be combined with alternative modeling or solution methods for the NRP.

Acknowledgements

We thank Data and Development Support at Region Zealand (DU) for providing insight, data and other support for this research. Furthermore, Elín Björk Böðvarsdóttir's PhD project has received financial support from DU. Research partially supported by Data-driven logistics (FWO-S007318N). Editorial consultation provided by Luke Connolly (KU Leuven).

References

Branke, J., Deb, K., Miettinen, K., Slowinski, R., 2008. Multiobjective Optimization. Interactive and Evolutionary Approaches. Springer-Verlag.

Burke, E. K., De Causmaecker, P., Petrovic, S., Vanden Berghe, G., 2002. A multi criteria meta-heuristic approach to nurse rostering. Proceedings of the 2002 Congress on Evolutionary Computation, Cec 2002 2, 1004413, 1197–1202.

Burke, E. K., De Causmaecker, P., Vanden Berghe, G., Van Landeghem, H., 2004. The state of the art of nurse rostering. Journal of Scheduling 7 (6), 441–449.

Burke, E. K., Li, J., Qu, R., 2012. A pareto-based search methodology for multi-objective nurse scheduling. Annals of Operations Research 196 (1), 91–109.

Böðvarsdóttir, E. B., Bagger, N.-C. F., Høffner, L. E., Stidsen, T., 2019a. A comprehensive integer programming formulation of the nurse rostering problem in Denmark. Technical report.

Böðvarsdóttir, E. B., Bagger, N.-C. F., Høffner, L. E., Stidsen, T., 2019b. Data for research on nurse rostering in Denmark [Data set]. http://doi.org/10.5281/zenodo.4004800.

Böðvarsdóttir, E. B., Smet, P., Vanden Berghe, G., Stidsen, T., 2019c. A modeling methodology to support nurse rostering practitioners. In: Proceedings of the 9th Multidisciplinary International Conference on Scheduling: Theory and Applications. pp. 141–155.

De Causmaecker, P., Vanden Berghe, G., 2011. A categorisation of nurse rostering problems. Journal of Scheduling 14 (1), 3–16.

Gärtner, J., Bohle, P., Arlinghaus, A., Schafhauser, W., Krennwallner, T., Widl, M., 2018. Scheduling matters-some potential requirements for future rostering competitions from a practitioner's view. In: PATAT 2018 - Proceedings of the 12th International Conference on the Practice and Theory of Automated Timetabling. pp. 33–42.

Jones, D., Tamiz, M., 2010. Practical goal programming. Springer.

Li, J., Burke, E. K., Curtois, T., Petrovic, S., Qu, R., 2012. The falling tide algorithm: A new multi-objective approach for complex workforce scheduling. Omega 40 (3), 283–293.

Liffiton, M. H., Previti, A., Malik, A., Marques-Silva, J., 2016. Fast, flexible MUS enumeration. Constraints 21 (2), 223–250.

Luque, M., Miettinen, K., Eskelinen, P., Ruiz, F., 2009. Incorporating preference information in interactive reference point methods for multiobjective optimization. Omega 37 (2), 450–462.

Mihaylov, M., Smet, P., Van Den Noortgate, W., Vanden Berghe, G., 2016. Facilitating the transition from manual to automated nurse rostering. Health Systems 5 (2), 120–131.

Petrovic, S., Vanden Berghe, G., 2012. A comparison of two approaches to nurse rostering problems. Annals of Operations Research 194 (1), 365–384.

Rihm, T., Baumann, P., 2018. Staff assignment with lexicographically ordered acceptance levels. Journal of Scheduling 21 (2), 167–189.

Smet, P., Brucker, P., De Causmaecker, P., Vanden Berghe, G., 2016. Polynomially solvable personnel rostering problems. European Journal of Operational Research 249 (1), 67–75.

Smet, P., Salassa, F., Vanden Berghe, G., 2017. Local and global constraint consistency in personnel rostering. International Transactions in Operational Research 24 (5), 1099–1117.

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.

Appendix A. Reformulation of Böðvarsdóttir et al. (2019a)

After a discussion with the practitioners to identify the actual targets, we modified the formulation presented by Böðvarsdóttir et al. (2019a) by removing or altering soft constraints as described in Table A.18. Hard constraints remain as described by Böðvarsdóttir et al. (2019a). We emphasize that this modification does not impact the model's ability to address real-life problems, but instead reduces some intricacies imposed by the weighted sum formulation.

Appendix B. Further results

Appendix B.1. General acceptance thresholds

In this appendix, we provide further results for the first category of experiments, i.e., employing general acceptance thresholds in the LGP framework.

Table B.19 provides the results for Experiment 1C, showing 7-9 target violations for each instance. Compared to Experiments 1A-1B, we see an increase in the magnitude of each violation, especially for lower priority targets.

Table B.20 presents the fulfilment of requests as a percentage of the best possible (from Table 11). We use the best possible rather than the number of given requests to ensure that these results are comparable with those from the second category of experiments. We present both the minimum percentage and the average percentage across all nurses who formulated requests. The average fulfillment of requests is correlated to the acceptance threshold (as defined in Table 4). Thus, Setting A generally yields the lowest fulfillment while Setting C the highest. Furthermore, Setting A shows a low minimum fulfillment compared to the other two settings, even down to 0% for two instances.

Table A.18: Modifications to the formulation presented by (Böðvarsdóttir et al., 2019a).

Modification	Affected constraints	Argumentation
Split constraint into several targets	S1	This constraint associates different (nurse, day, shift) assignments with weights depending on various factors. In the reformulation we address targets for different underlying factors (e.g., requests or chaperoning). We only consider those factors that the practitioners associate with direct targets.
Reformulate constraint	S2	Instead of assessing violations for the rostering horizon as a whole (as in Böðvarsdóttir et al. (2019a)), we assess them for specific days (i.e., moving from global view to local view).
	S11	Instead of associating float nurses to coverage constraints we associate them with shifts.
Remove constraint	S4, S5, S6, S14	These constraints are supported by other constraints and the practitioners do not have any directly related targets.
	S12	This constraint was only used for the early instances and later dropped from the formulation.
	S15	The constraint has a very low weight, and the practitioners do not link it to any target.
Reduce generality of constraints	S7, S8, S9, S10	These constraints are formulated in a general manner by Böðvarsdóttir et al. (2019a), allowing for immense flexibility. We reformulate these constraints to match the specific targets that practitioners have set related to them (e.g., instead of considering consecutive days in general we consider the work week from Monday to Sunday).

The impact of the last lexicographic step that maximizes the fulfillment of requests depends on the percentage used when setting the acceptance thresholds. For Setting C, this step does not have any impact as we have enforced hard constraints for meeting the number of requests put forth with minimum deviation as estimated in the corresponding goal programs. For Setting A, this step increases the number of fulfilled requests by 31%, whereas the increase is less than 2% for Setting

Table B.19: Target violations for Experiment 1C. For requests and weekend preferences, the former number presents the accumulated violation across all nurses, and the number in brackets is the maximum individual violation.

Target	A01	A02	A03	A04	A05	A06	A07	B01	B02	B03	B04	B05
Float Nurses					2			6	6	12	12	26
High Priority Requests		2 (1)		10 (4)	3 (2)	2 (2)	2 (1)	2 (1)	2 (2)			
Weekend Preferences								6 (2)	4 (2)	14 (4)	10 (4)	6 (2)
Distance Between PF	8	16	28	42	13	26	14	14	10	5	8	14
No Multiple Night Sequences						1	1					1
Overtime			10.86		24.44	4.85	11.11			15.35	26.57	1.01
Undertime	16.67	6.21	18.73	98.99		243.69	9.09	2.02	111.51	902.02	292.73	33.54
Total Requests	35 (4)	39 (3)	41 (3)	56 (4)	55 (6)	37 (5)	61 (8)	2 (1)	74 (6)		9 (3)	5 (3)
Chaperoning Shift	6	2	2	6	1	3	5					
Balance Excess Staffing	2											
Monday Off After Work Weekend	13	8	11	10	9	7	6	5	15	8	1	3
Isolated Workdays	53	64	48	59	74	43	37	2	37	1	5	3

B.

Appendix B.2. Estimated acceptance thresholds

In this appendix, we provide further results for the second category of experiments, i.e., employing estimated acceptance thresholds in the LGP framework.

For Experiment 2A (Table 13), two violations stand out. First, instance B04 has an *Overtime* violation, due to a nurse with significant overtime from the previous roster, along with hard work requests for a majority of the current rostering horizon. This nurse requests three days off, but we are only able to grant two of those without violating the staffing requirements. The second number that stands out is a violation of 8 for *Monday Off After Work Weekend* for instance B03. The reason for this extensive violation is the integration of a new IT system, meaning that the nurses had to participate in training courses for the new system. As these courses reduce the availability of the nurses, we have less flexibility and are forced to assign some nurses to work on Monday following their work weekend to meet the staffing requirements.

In Experiments 2B-2C, we include acceptance thresholds for weekend preferences. This target is only active in Ward B (instances B01-B05) and we do not reach the acceptance threshold for any of these instances. We must emphasize that those violations are unavoidable, as the managers have previously decided which nurses should work during each weekend. For some weekends, this allocation means that we cannot satisfy the staffing requirements without violating weekend

Table B.20: The individual request fulfillment [%] of the rosters generated in the first category of experiments.

Instance	Experiment 1A		Experiment 1B		Experiment 1C	
	Minimum	Average	Minimum	Average	Minimum	Average
A01	25.00	82.90	72.73	91.56	72.73	97.61
A02	0.00	85.34	80.00	92.78	80.00	97.36
A03	33.33	85.73	80.00	95.55	75.00	97.76
A04	0.00	86.45	66.67	94.54	66.67	97.48
A05	45.00	82.83	72.22	93.11	72.22	96.84
A06	33.33	89.20	66.67	95.46	66.67	98.60
A07	63.63	88.55	65.00	91.84	65.00	95.52
B01	100.00	100.00	100.00	100.00	100.00	100.00
B02	40.00	78.56	50.00	83.53	50.00	84.56
B03	66.67	83.33	100.00	100.00	100.00	100.00
B04	66.67	94.29	66.67	90.45	66.67	90.45
B05	100.00	100.00	75.00	98.08	75.00	98.08

preferences.

Table B.21 presents the fulfillment of requests as a percentage of the best possible for each individual nurse. The average request fulfillment is stable across all settings, but the minimum for several instances is substantially lower when using Setting A (i.e., excluding acceptance thresholds for requests and weekend preferences). Comparing with the results for general thresholds shows that the percentage of fulfilled requests is generally higher when employing estimated thresholds. Furthermore, the impact of the last step (i.e., maximizing the overall request fulfillment) is substantially increased, being 41% for Setting A and 7% for Setting B. These results indicate that the problem is not as tightly constrained after enforcing estimated thresholds for all target as it is when using general thresholds.

Table B.21: The individual request fulfillment [%] of the rosters generated in the second category of experiments.

Instance	Experiment 2A		Experiment 2B		Experiment 2C	
	Minimum	Average	Minimum	Average	Minimum	Average
A01	50.00	95.32	80.00	95.99	88.89	98.36
A02	75.00	98.22	83.33	97.60	83.33	98.88
A03	75.00	97.69	84.62	97.12	75.00	98.17
A04	83.33	99.36	83.33	98.56	87.50	99.63
A05	50.00	95.95	82.35	96.13	84.62	98.44
A06	83.33	98.76	84.62	98.52	92.86	99.72
A07	77.78	96.84	80.00	97.39	80.00	98.82
B01	100.00	100.00	100.00	100.00	100.00	100.00
B02	50.00	90.83	50.00	89.30	50.00	90.99
B03	100.00	100.00	100.00	100.00	100.00	100.00
B04	66.67	95.83	66.67	92.76	66.67	92.81
B05	100.00	100.00	75.00	98.08	75.00	98.08