Personnel Rostering in a Public Library: A Case Study

Bruno Basckeira Chinaglia[1], Douglas Nogueira Nascimento[1][0000-0001-9278-505X], Greet Vanden Berghe[2][0000-0002-0275-5568], and Franklina Maria Bragion Toledo[1][0000-0003-2823-7600]

[1] Institute of Mathematical and Computer Sciences, ICMC-USP, Brazil
[2] Department of Computer Science, KU Leuven, Belgium
brunobchinaglia@usp.br, douglasnn92@gmail.com,
greet.vanden.berghe@kuleuven.be, fran@icmc.usp.br

Abstract. This paper addresses a real-world personnel rostering problem arising in a public university library. The roster covers Monday–Saturday operations and must satisfy shift-coverage and skill-mix requirements, while handling a set of undesirable duties (Friday evenings and Saturdays) subject to rotation and adjacency rules. The main challenge is to reconcile these operational constraints with staff preferences regarding undesirable duties and to do so equitably. We introduce a max–min fairness criterion that maximises the minimum normalised *happiness* across employees, thereby protecting the least satisfied staff member with respect to the roster accommodating their preferences. We first investigate the effectiveness of a compact mixed-integer linear programming (MILP) formulation for this setting. We also propose an alternative, structure-exploiting solution workflow based on a two-stage decomposition: Stage 1 optimises assignments on the undesirable days, and Stage 2 completes weekday evening duties using the fixed weekly availability patterns implied by the Stage 1 decisions. Since weeks with holidays may induce additional imbalance in days off, we further introduce a refinement model that balances worked days within holiday weeks by minimising the range between the most- and least-loaded staff members. A real-world application shows that the proposed workflow produces equitable and transparent rosters, with high worst-case normalised satisfaction, and can be used routinely by non-technical personnel.

Keywords: Personnel rostering · Mixed-integer linear programming · Decomposition approach

1 Introduction

Personnel rostering problems are widespread in service operations and have been extensively studied in contexts such as hospitals [2, 12], schools and universities [3], and many other sectors [5, 8]. In this paper, we address a real rostering problem at the ICMC/USP Library. The responsible librarian used to prepare this roster manually, which was a very time-consuming task. Despite significant efforts, the team did not always perceive the roster as fair. The problem requires reconciling coverage and skill-mix requirements with staff availability and preferences under a variety of institutional rules. Such decisions directly affect service quality, the fair allocation of undesirable duties (weekend and evening assignments), and staff satisfaction [6].

Generating good-quality solutions for personnel rostering can benefit from advances in modelling and solution methods [5]. Survey and tutorial contributions highlight that practical approaches typically rely on carefully designed models, often within a mathematical programming framework that deals with feasibility constraints [7] and objective components, for example soft constraint violations [9]. They also emphasise that exploiting problem structure, focusing on a limited set of undesirable duties, or separating independent decisions is a key driver of scalable, implementable solution workflows [11]. Beyond feasibility and solution quality, practical adoption also depends on transparency and perceived fairness. For deployment in practice, decision makers often require solutions that are understandable and whose allocations can be justified to staff, especially when considering personal preferences [1].

We consider a personnel rostering problem where staff members, based on their skills, belong to either the Information Processing (IP) or the User Services (US) group. Apart from this division, the staff members in the ICMC/USP library also belong to one of three categories: *support staff* (4), *technicians* (4) and *librarians* (3), based on their level of education and training. Support staff handle routine tasks such as public service, and their schedules are fixed. These staff members can therefore be omitted from the rostering problem. The remaining staff must be assigned to weekday evening and Saturday shifts over a semester-long horizon.

Besides standard coverage and skill-mix requirements, the roster must maximise a fairness objective which depends on the distribution of Friday evening and Saturday assignments, which are considered undesirable. Interestingly, staff members explicitly state preferences for these undesirable shifts, and the library requires these duties to be allocated transparently and equitably. It is important to enforce a rotation of these undesirable shifts when modelling the problem. In addition, the occurrence of consecutive Friday–Saturday assignments should be restricted. Compared with personnel rostering models that primarily minimise soft constraint violations or maximise total preference satisfaction, we explicitly optimise equity through a max–min objective that maximises the minimum normalised satisfaction across employees, thereby protecting the least satisfied staff member and producing allocations that are easy to justify to practitioners.

While personnel rostering problems are often assumed to be NP-hard, recent theoretical work shows that tractability can depend strongly on the exact constraint set and modelling granularity, and that several meaningful rostering variants admit polynomial-time formulations via network flow models [10]. Motivated by these insights, we investigate which components drive difficulty in the ICMC/USP Library setting, with particular emphasis on (i) whether a compact mixed-integer linear programming (MILP) approach can solve the problem directly, and how the inclusion of an explicit max–min fairness objective affects solution structure and practical solvability; and (ii) whether exploiting the separation between undesirable duties (Friday evenings and Saturdays) and weekday completion enables a modular, implementable workflow. This perspective also motivates our decomposition and refinement steps, which isolate undesirable-day decisions and mitigate additional imbalances induced by holidays.

The remainder of this paper is organised as follows. Section 2 formalises the problem and its operational rules. Section 3 presents a compact mixed-integer linear programming formulation, while Section 4 describes an alternative two-stage decomposition approach. Section 5 introduces an improvement strategy to handle holidays. Section 6 reports the computational results and evidence from deployment. Finally, Section 7 concludes the paper and points to future work.

2 Problem description

The personnel rostering problem involves assigning employees to shifts over a finite planning horizon to meet demand coverage, while adhering to institutional and contractual rules (hard constraints). Due to the possibility of omitting regular day shifts from the problem, coverage and assignments can be expressed per day. The planning horizon follows a Monday–Saturday schedule. The problem focuses on the assignment of undesirable days (i.e., Friday evenings and Saturdays). Employee preferences for these days are treated as soft constraints and balanced using fairness-based objectives. Although simple cyclical rotations are attractive because of their transparency and low implementation effort, they would be restrictive for the librarian case study. Friday–Saturday adjacency restrictions, holiday effects, and day-specific employee preferences cannot be met with a single repeating rotation without compromising feasibility or fairness.

The rostering decisions concern assigning technical and librarian staff to weekday evening and Saturday shifts, while respecting the following requirements: (i) every weekday and Saturday must be covered by two staff members at technician or librarian level, (ii) Saturdays must be covered by one staff member from each group (IP and US), (iii) assignments on Fridays and Saturdays must follow a rotation. Throughout the paper, we use the term *rotation* to denote a policy that distributes the undesirable assignments among eligible staff over the planning horizon.

After a series of meetings, the staff members involved in the rostering process requested the incorporation of additional *hard constraints* for the Friday and Saturday rotations to avoid repeatedly assigning the same employee to undesirable days in consecutive weeks. In particular, (iv) no employee may be assigned to three consecutive Fridays, (v) no employee may be assigned to three consecutive Saturdays, and (vi) an employee assigned to a Friday must not be assigned to the immediately following Saturday.

Staff also stated their fixed *availability pattern* for weekdays to be followed throughout the planning horizon. Three patterns are allowed: (1) Monday, Tuesday and Friday; (2) Tuesday, Wednesday and Friday; and (3) Wednesday, Thursday and Friday. Each employee is associated with exactly one of these patterns for the entire rostering horizon. In any given week, each employee may be scheduled for at most two of the three days specified by their availability pattern. The librarian scheduling problem belongs to the category $SabB|2|L$ according to [4].

3 Mathematical Model

This section presents a compact MILP formulation for the library personnel rostering problem described in Section 2. Our formulation captures the assignment of technicians and librarians over the planning horizon, while enforcing the constraints described in the previous section. In addition, it incorporates individual preference scores for working on undesirable days and pursues a max–min fairness criterion, i.e., it maximises the preference satisfaction of the least satisfied staff member. Staff members provide preference ratings for each Friday evening and Saturday. These preference ratings are encoded as integer scores HF_{id} and HS_{id}, with 1 = dislike, 2 = indifferent, and 3 = like.

The proposed model, here denoted by *LRM (Library Rostering Model)*, is represented by (1)–(21). For clarity, in Table 1, we first introduce the sets, the parameters and the decision variables used throughout the formulation.

Table 1: Sets, parameters and variables for the LRM.

Sets	
ST	set of staff members ($ST = \{1, \dots, \lvert ST\rvert\}$);
IP	set of staff members in the Information Processing department where $IP \subset ST$;
US	set of staff members in the User Services department where $US \subset ST$, $IP \cup US = ST$ and $IP \cap US = \oslash$;
M	set of months in the planning horizon ($M = \{1, \dots, \lvert M\rvert\}$);
D	set of days in the planning horizon ($D = \{1, ..., \lvert D\rvert\}$);
$\overline{D}$	set of non-working days in the planning horizon;
F_m (S_m)	set of Fridays (Saturdays) in month $m \in M$;
$NF(NS)$	set of Fridays (Saturdays) in the horizon ($NF = \cup_{m\in M} F_m$ and $NS = \cup_{m\in M} S_m$);
$\overline{NF}(\overline{NS})$	set of all Fridays (Saturdays) except last two of the horizon.
Parameters	
AF_m	average number of Fridays per staff member in month $m \in M$ $\left(AF_m = \frac{2\,\lvert F_m\setminus\overline{D}\rvert}{\lvert ST\rvert}\right)$, the factor 2 reflects the coverage constraint of two employees per Friday;
$A2F_m$	average number of Fridays per staff member over the consecutive months m and $m+1$, $m \in M \setminus \{\lvert M\rvert\}$ $\left(A2F_m = AF_m + AF_{m+1}\right)$;
AHF	average number of Fridays per staff member over the horizon $\left(AHF = \frac{2\,\lvert NF\setminus\overline{D}\rvert}{\lvert ST\rvert}\right)$;
$AS_m^{IP}(AS_m^{US})$	average number of Saturdays per staff member in the IP(US) group in month $m \in M$ $\left(AS_m^{IP} = \frac{\lvert S_m\setminus\overline{D}\rvert}{\lvert IP\rvert}\right)$;
$A2S_m^{IP}(A2S_m^{US})$	average number of Saturdays per staff member in the IP (US) group over the consecutive months m and $m+1$, $m \in M \setminus \{\lvert M\rvert\}$;
AHS^{IP} (AHS^{US})	average number of Saturdays per staff member in the IP (US) group over the horizon;
HF_{id} (HS_{id})	"happiness" (preference) of staff member $i \in ST$ if working on Friday $d \in NF$ (Saturday $d \in NS$);
$H_i^{\max}$	maximum attainable happiness for staff member $i \in ST$ based on their preferences.
Variables	
$f_{\max}$	continuous variable representing the minimum normalised happiness across staff;
x_{id}	equals 1 if staff member $i \in ST$ is assigned to work on day $d \in D$, and 0 otherwise.

$$\textit{Maximise} \quad f_{\max} \tag{1}$$

subject to

$$f_{\max} \leq \frac{\sum_{d \in NF \setminus \overline{D}} HF_{id}\, x_{id} + \sum_{d \in NS \setminus \overline{D}} HS_{id}\, x_{id}}{H_i^{\max}} \qquad i \in ST \qquad (2)$$

$$\sum_{i \in ST} x_{id} = 2 \qquad d \in D \setminus \overline{D} \qquad (3)$$

$$\sum_{i \in IP} x_{id} = 1 \qquad d \in NS \setminus \overline{D} \qquad (4)$$

$$\sum_{i \in US} x_{id} = 1 \qquad d \in NS \setminus \overline{D} \qquad (5)$$

$$x_{i,d-4} + x_{i,d-3} + x_{id} \leq 2 \qquad i \in ST,\ d \in NF \qquad (6)$$

$$x_{i,d-3} + x_{i,d-2} + x_{id} \leq 2 \qquad i \in ST,\ d \in NF \qquad (7)$$

$$x_{i,d-2} + x_{i,d-1} + x_{id} \leq 2 \qquad i \in ST,\ d \in NF \qquad (8)$$

$$x_{id} + x_{i,d+1} \leq 1 \qquad i \in ST,\ d \in NF \qquad (9)$$

$$x_{id} + x_{i,d+7} + x_{i,d+14} \leq 2 \qquad i \in ST,\ d \in \overline{NF} \cup \overline{NS} \qquad (10)$$

$$\lfloor AF_m \rfloor \leq \sum_{d \in F_m \setminus \overline{D}} x_{id} \leq \lceil AF_m \rceil \qquad i \in ST,\ m \in M \qquad (11)$$

$$\lfloor A2F_m \rfloor \leq \sum_{d \in (F_m \cup F_{m+1}) \setminus \overline{D}} x_{id} \leq \lceil A2F_m \rceil \qquad i \in ST,\ m \in M \setminus \{|M|\} \qquad (12)$$

$$\lfloor AHF \rfloor \leq \sum_{d \in NF \setminus \overline{D}} x_{id} \leq \lceil AHF \rceil \qquad i \in ST \qquad (13)$$

$$\left\lfloor AS_m^{IP} \right\rfloor \leq \sum_{d \in S_m \setminus \overline{D}} x_{id} \leq \left\lceil AS_m^{IP} \right\rceil \qquad i \in IP,\ m \in M \qquad (14)$$

$$\left\lfloor A2S_m^{IP} \right\rfloor \leq \sum_{d \in (S_m \cup S_{m+1}) \setminus \overline{D}} x_{id} \leq \left\lceil A2S_m^{IP} \right\rceil \qquad i \in IP,\ m \in M \setminus \{|M|\} \qquad (15)$$

$$\left\lfloor AHS^{IP} \right\rfloor \leq \sum_{d \in NS \setminus \overline{D}} x_{id} \leq \left\lceil AHS^{IP} \right\rceil \qquad i \in IP \qquad (16)$$

$$\lfloor AS_m^{US} \rfloor \leq \sum_{d \in S_m \setminus \overline{D}} x_{id} \leq \lceil AS_m^{US} \rceil \qquad i \in US,\ m \in M \qquad (17)$$

$$\left\lfloor A2S_m^{US} \right\rfloor \leq \sum_{d \in (S_m \cup S_{m+1}) \setminus \overline{D}} x_{id} \leq \left\lceil A2S_m^{US} \right\rceil \qquad i \in US,\ m \in M \setminus \{|M|\} \qquad (18)$$

$$\lfloor AHS^{US} \rfloor \leq \sum_{d \in NS \setminus \overline{D}} x_{id} \leq \lceil AHS^{US} \rceil \qquad i \in US \qquad (19)$$

$$f_{\max} \geq 0 \qquad (20)$$

$$x_{id} \in \{0, 1\} \qquad i \in ST,\ d \in D \setminus \overline{D} \qquad (21)$$

The objective function (1) of LRM aims to maximise the happiness level of the least satisfied staff member, as defined by Constraints (2). We distinguish Friday and Saturday preference scores (HF_{id} and HS_{id}) because these refer to different undesirable duties and are collected separately for $d \in NF$ and $d \in NS$. For each staff member $i \in ST$, the numerator of (2) represents the happiness due to undesirable duties assigned to i on Fridays and Saturdays, weighted by the corresponding preference scores. This happiness level is computed proportionally, taking into account each staff member's maximum attainable happiness (parameter $H_i^{\max}$) based on their stated happiness ratings. The denominator $H_i^{\max}$ represents an upper bound on the happiness that employee i can attain under ideal assignments. This value is computed by assuming that employee i works the minimum required number of Friday and Saturday duties. Among these required duties, the model assumes employee i is assigned to the $\lfloor AHF \rfloor$ (which is equal to $\lfloor AHS^{IP} \rfloor$ or $\lfloor AHS^{US} \rfloor$) days with the highest preference scores, sorted in non-increasing order. Finally, each unassigned undesirable day receives a score of 4. Consequently, for staff members in the IP group, days off on Fridays and Saturdays contribute $4 * \left(|NF \setminus \overline{D}| - \lfloor AHF \rfloor + |NS \setminus \overline{D}| - \lfloor AHS^{IP} \rfloor\right)$ to $H_i^{\max}$, and analogously for the US group by replacing AHS^{IP} with AHS^{US}.

Constraints (3) ensure that exactly two staff members are assigned to each shift, while constraints (4)–(5) guarantee that one staff member from each group (IP and US) is assigned to work on Saturdays. Constraints (6)–(8) guarantee the weekday evening-shift availability rule. Specifically, for each Friday $d \in NF$, they limit the number of evening duties assigned within the corresponding three-day availability pattern: Monday–Tuesday–Friday (6), Tuesday–Wednesday–Friday (7), and Wednesday–Thursday–Friday (8). Constraints (9) prevent a staff member from being assigned to consecutive Friday and Saturday duties. Constraints (10) prevent three consecutive Friday/Saturday duties for any employee.

The balancing constraints (11)–(13) control the Friday rotation by ensuring that each staff member works between a minimum and maximum number of Friday evenings per month (Constraint (11)), over any two consecutive months (Constraint (12)), and over the entire planning horizon (Constraint (13)). Analogously, Constraints (14)–(19) balance Saturday duties across the IP and US groups. The constraints are stated separately because the groups have different sizes and each group must have a staff member working on Saturdays. In these constraints, expressions such as $F_m \setminus \overline{D}$ (and, analogously, $(F_m \cup F_{m+1}) \setminus \overline{D}$, $S_m \setminus \overline{D}$, and $NS \setminus \overline{D}$) denote the corresponding sets of *working* Fridays/Saturdays, obtained by excluding non-working days in $\overline{D}$. That is, the summations count only assignments on days when the library is open. Finally, Constraints (20)–(21) define the variables' domains.

LRM provides a detailed and comprehensive formulation that simultaneously incorporates operational requirements, individual preferences, and equity criteria in the allocation of duties. However, this richness of representation comes at the cost of a large number of variables and constraints in the formulation. In the real-world setting considered, the model is feasible because the staffing level and the library rules are mutually consistent: there are seven eligible staff members, each of whom may work at most two duties per week, yielding a weekly capacity of 14 assignments, whereas the library requires only 12 assignments per week. In addition, the staff is distributed across the fixed availability patterns and the two functional groups in a way that satisfies the weekday and Saturday coverage requirements.

A critical review of the model indicates opportunities to investigate alternative approaches that preserve the equity and preference-satisfaction criteria while simplifying the formulation. These considerations motivated the development of a new solution approach described in Section 4.

4 Decomposition Approach

Since each staff member can work at most two night shifts per week, once a staff member is assigned to Friday in a given week, the second working day is determined by that staff member's fixed evening-shift pattern (e.g. Monday–Tuesday, Tuesday–Wednesday or Wednesday–Thursday). Therefore, assignments for the remaining weekdays can be derived from the Friday decision. This feature enables handling the undesirable days (Friday and Saturday) as a subproblem. Given an optimal solution for this subproblem, all that remains is to complete the schedule with the other coverage constraints.

In regular weeks without holidays, the Friday assignments largely determine the remaining weekday evening duties based on employees' fixed availability patterns and the restriction to at most two duties per week. Moreover, since the fairness objective depends only on Friday and Saturday assignments, the completion step in regular weeks does not alter the objective function's value; it only restores feasibility for the remaining weekdays. In this regard, the proposed decomposition is strongly supported by the problem's structure. In holiday weeks, however, the direct completion may induce additional imbalance in the number of worked days, which motivates the refinement step we introduce in Section 5. Therefore, the decomposition should be understood as a problem-specific structure-exploiting workflow rather than as a formally exact Benders-type decomposition.

Accordingly, we propose a two-stage decomposition approach to solve the problem. In *Stage 1*, we exclusively determine the assignments associated with the undesirable days. Once these rotations are fixed, *Stage 2* completes the schedule for the remaining weekdays, enforcing labour constraints and individual preferences. This approach enables the subproblem to account for the distribution of days off induced by holidays, yielding fairer schedules.

4.1 *Stage 1*: Friday and Saturday rotations

Stage 1, here denoted as *LRM-S*, is a simplified version of *LRM*, with the following modifications:

- Decision variables x_{id} are defined only for Fridays and Saturdays ($x_{id} \in \{0,1\},\ i \in ST,\ d \in NF \cup NS$).
- Constraints (3) are enforced for all $d \in (NF \cup NS) \setminus (\overline{D} \cup \overline{D})$.
- Constraints (6)–(8) are removed.

4.2 *Stage 2*: Weekday evening assignments

Once the Friday and Saturday rotation is defined, the roster for the remaining weekdays must be completed. This assignment is straightforward for weeks without holidays, since each staff member works exactly two days per week. Figure 1 illustrates how the second stage can be completed for a fixed Friday rotation.

Fig. 1: Example of weekday evening-shift completion after fixing the Friday rotation.

Staff	**Mo**	**Tu**	**We**	**Th**	**Fr**
1	✓	✓			✓
2	✓	✓			✓
3		✓	✓		✓
4			✓	✓	✓
5			✓	✓	✓

(a) Availability pattern for each staff member.

Staff	**Mo**	**Tu**	**We**	**Th**	**Fr**
1					
2					✓
3					
4					
5					✓

(b) Friday assignments fixed in Stage 1.

Staff	**Mo**	**Tu**	**We**	**Th**	**Fr**
1	✓	✓			
2					✓
3		✓	✓		
4			✓	✓	
5					✓

(c) Automatic assignment of staff not selected for Friday.

Staff	**Mo**	**Tu**	**We**	**Th**	**Fr**
1	✓	✓			
2	✓				✓
3		✓	✓		
4			✓	✓	
5				✓	✓

(d) Completion of the remaining assignments in Stage 2.

Figure 1a presents the weekly availability patterns of the five staff members. In this example, staff members 1 and 2 follow pattern FN_1 and can work on Mondays (Mo), Tuesdays (Tu) and Fridays (Fr); staff member 3 follows pattern FN_2 and can work on Tuesdays, Wednesdays (We) and Fridays; and staff members 4 and 5 follow FN_3 and can work on Wednesdays, Thursdays (Th) and Fridays. Figure 1b shows the Friday assignments determined in Stage 1, highlighted in blue for staff members 2 and 5. Given this Friday decision, the remaining weekday assignments can be inferred from the availability pattern. Figure 1c indicates that staff members who are not assigned on Friday must work on the other two days of their pattern; therefore, staff members 1 and 3 are assigned to Tuesday, while staff member 4 is assigned to Wednesday and Thursday. Finally, Figure 1d shows the completion of the schedule: the staff members assigned to Friday are allocated to the only remaining day in their pattern that still requires coverage. Staff member 2 is assigned to Monday, and staff member 5 to Thursday, yielding the final roster.

The main advantage of this approach lies in the clarity and flexibility of the resulting model. By explicitly identifying the undesirable days, the formulation can accommodate computational adjustments—such as including or excluding holidays—in a direct manner, without requiring a major reformulation of the constraints. Moreover, separating the definition of undesirable-day assignments from the completion of the remaining weekdays significantly reduces the computational time and improves scalability to longer planning horizons. Hence, the proposed approach preserves the key principles of the original model, particularly the pursuit of equity and preference satisfaction, while offering a leaner, modular formulation better suited to practical applications and future analyses.

5 Holiday-week refinement

While solving real-world instances, we observed that weeks containing holidays introduce additional imbalance in the workload distribution, particularly regarding days off induced by non-working days. To address this issue, we developed a complementary model that balances the distribution of working days across weeks with holidays, accounting for different feasible weekly patterns and respecting the maximum number of evening-shifts per week. The formulation seeks to minimise the gap between the maximum and minimum number of worked days across staff members over the planning horizon. In this improvement model, Monday and Thursday in holiday weeks are parameters (Mo_s, Th_s) in addition to Friday assignments (Fr_{is}). The decisions focus on distributing Tuesday/Wednesday assignments so that holiday-related time off is spread as evenly as possible across staff. The refinement model for weeks with holidays, which aims to reduce disparities in the number of worked days (i.e., to minimise the range between the most- and least-loaded staff members), is called *(LRM-H)* and defined by (22)–(36). The corresponding sets, parameters, and decision variables are given in Table 2.

Table 2: Sets, parameters and variables of the model *(LRM-H)*.

Sets	
FN_1	set of staff members with working pattern is Mon., Tue. and Fri.;
FN_2	set of staff members with working pattern is Tue., Wed. and Fri.;
FN_3	set of staff members with working pattern is Wed., Thu. and Fri.;
NSF	set of weeks containing at least one holiday;
Parameters	
Mo_s (Tu_s, We_s, Th_s)	equals 1 if Monday (Tuesday, Wednesday, Thursday) in week $s \in NSF$ is a working day (not a holiday), and 0 otherwise;
Fr_{is}	equals 1 if staff member $i \in ST$ was assigned on Friday of week $s \in NSF$ (and 0 otherwise); if Friday is a holiday, all $Fr_{is} = 0$.
Variables	
t_{is}	is equal to 1 if staff member $i \in FN_1 \cup FN_2$ works on Tuesday in week $s \in NSF$, and 0 otherwise;
q_{is}	is equal to 1 if staff member $i \in FN_2 \cup FN_3$ works on Wednesday in week $s \in NSF$, and 0 otherwise;
$Z_{\min}$ ($Z_{\max}$)	minimum (maximum) number of Monday-Friday assignments across staff members over holiday weeks.

$$\textit{Minimise} \quad Z_{\max} - Z_{\min} \tag{22}$$

subject to

$$Z_{\max} \geq \sum_{s \in NSF} (Mo_s + t_{is} + Fr_{is}) \qquad i \in FN_1 \tag{23}$$

$$Z_{\max} \geq \sum_{s \in NSF} (t_{is} + q_{is} + Fr_{is}) \qquad i \in FN_2 \tag{24}$$

$$Z_{\max} \geq \sum_{s \in NSF} (q_{is} + Th_s + Fr_{is}) \qquad i \in FN_3 \tag{25}$$

$$Z_{\min} \leq \sum_{s \in NSF} (Mo_s + t_{is} + Fr_{is}) \qquad i \in FN_1 \tag{26}$$

$$Z_{\min} \leq \sum_{s \in NSF} (t_{is} + q_{is} + Fr_{is}) \qquad i \in FN_2 \tag{27}$$

$$Z_{\min} \leq \sum_{s \in NSF} (q_{is} + Th_s + Fr_{is}) \qquad i \in FN_3 \tag{28}$$

$$Mo_s + t_{is} + Fr_{is} \leq 2 \qquad i \in FN_1,\ s \in NSF \tag{29}$$

$$t_{is} + q_{is} + Fr_{is} \leq 2 \qquad i \in FN_2,\ s \in NSF \tag{30}$$

$$q_{is} + Th_s + Fr_{is} \leq 2 \qquad i \in FN_3,\ s \in NSF \tag{31}$$

$$\sum_{i \in FN_1 \cup FN_2} t_{is} = 2\ Tu_s \qquad s \in NSF \tag{32}$$

$$\sum_{i \in FN_2 \cup FN_3} q_{is} = 2\ We_s \qquad s \in NSF \tag{33}$$

$$Z_{\min}, Z_{\max} \geq 0 \tag{34}$$

$$t_{is} \in \{0, 1\} \qquad i \in FN_1 \cup FN_2,\ s \in NSF \tag{35}$$

$$q_{is} \in \{0, 1\} \qquad i \in FN_2 \cup FN_3,\ s \in NSF \tag{36}$$

Objective (22) minimises the range of evening assignments across staff members in holiday weeks, promoting equity. Constraints (23)–(25) and (26)–(28) define upper and lower bounds on each staff member's cumulative number of weekday assignments over the weeks with holidays. For a given staff member, this cumulative workload consists of the weekday assignments that are fixed once the holiday calendar is known (e.g. the parameter Mo_s is equal to zero on holidays), together with the Tuesday/Wednesday assignments determined by the refinement model (t_{is} and q_{is}) and the Friday assignment Fr_{is} fixed in Stage 1. Since this expression depends on the employee's weekly availability pattern, three pattern-specific pairs of constraints are required. In this way, $Z_{\min}$ and $Z_{\max}$ represent, respectively, the smallest and largest cumulative workloads among all staff members in holiday weeks. Analogously, Constraints (29)–(31) ensure that the total number of assignments received by a staff member in a holiday week does not exceed two. Finally, Constraints (32)–(33) enforce two assignees on Tuesday/Wednesday when these are not holidays. The variable domains are defined by Constraints (34)–(36).

6 Computational results

This section reports results from applying the developed decomposition approach to real planning instances from the ICMC/USP Library. More precisely, the semester rosters were first generated by applying Stage 1 to determine Friday and Saturday assignments and Stage 2 to complete the remaining weekday evening duties. The holiday-week refinement model serves only to evaluate its effect on workload equity in weeks affected by holidays.

We applied the two-stage decomposition approach to four semester-long rostering instances (one instance per semester), covering the second semester of 2024 through the first semester of 2026. Each semester was solved separately. The instances were solved using the SCIP Optimization Suite (10.0.1) on a computer with an Intel(R) Core(TM) i5-10300H processor, 16 GB of RAM, and the Windows 11 operating system. Across the four semesters, the Stage 1 models comprised, on average, 172 decision variables and 558 constraints, while the Stage 2 models comprised 110 decision variables and 143 constraints. The computational time required to generate the schedule for each semester was less than one second. Across these four instances, the minimum normalised happiness achieved was at least 0.95, ensuring that even the least satisfied staff member attained at least 95% of their maximum attainable happiness. These results show consistently high and balanced preference satisfaction regarding undesirable duties, despite the seasonal variability in the number and distribution of holidays and in the number and timing of undesirable-day assignments that had to be covered.

Figure 2 reports the effect of the holiday-week refinement on workload equity. The bars labelled “Without Holidays” correspond to the rosters obtained solely from the two-stage decomposition. In contrast, the bars labelled “With Holidays” correspond to the rosters after applying the holiday-week refinement model. The results indicate that incorporating this refinement yields a more equitable distribution of days off, thereby reducing the difference over the planning period and improving the perceived fairness of the resulting roster.

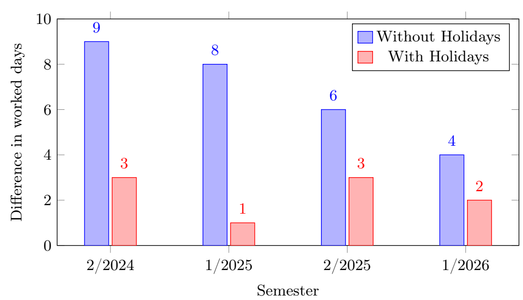

Fig. 2: Difference between the most- and least-loaded staff members in terms of worked days over the semester.

7 Final remarks

This paper presents a practical, optimization-based approach to generating staff schedules for a public university library. The proposed method combines a compact MILP formulation, a two-stage decomposition focused on undesirable duties, and a refinement step for weeks with holidays. In contrast to the manual, time-consuming roster construction previously performed by library staff, the developed approach generates high-quality schedules rapidly and transparently, facilitating its routine use in practice. Across four consecutive semesters, the minimum normalised satisfaction among staff members remained high (95%), indicating consistently robust preference satisfaction despite the variability in holiday configurations and operational conditions. In conclusion, the generated schedules were presented to the library team and have been successfully integrated into their routine since the second semester of 2025, contributing to greater perceived fairness and transparency in the rostering process.

Future research may explore the performance of the developed models and solution framework for larger instances, as well as the incorporation of additional operational constraints common to library settings. It would be interesting to investigate whether the personnel scheduling problem addressed in this paper can be solved with straightforward algorithms without using MILP solvers.

Acknowledgements This work was funded by the São Paulo Research Foundation (FAPESP), Brazil, grants #2025/01587-2 and #2023/16405-1. The authors also acknowledge financial support from the National Council for Scientific and Technological Development (CNPq) grant #309161/2022-3, and the Research Foundation – Flanders (FWO grant K804824N).

References

1. Ásgeirsson, E.I.: Bridging the gap between self schedules and feasible schedules in staff scheduling. Annals of Operations Research **218**, 51–69 (2014)
2. Burke, E.K., De Causmaecker, P., Vanden Berghe, G., Van Landeghem, H.: The state of the art of nurse rostering. Journal of Scheduling **7**, 441–499 (2004)
3. Ceschia, S., Di Gaspero, L., Schaerf, A.: Educational timetabling: Problems, benchmarks, and state-of-the-art results. European Journal of Operational Research **308**(1), 1–18 (2023)
4. De Causmaecker, P., Vanden Berghe, G.: A categorisation of nurse rostering problems. Journal of Scheduling **14**(1), 3–16 (2011)
5. Ernst, A., Jiang, H., Krishnamoorthy, M., Sier, D.: Staff scheduling and rostering: A review of applications, methods and models. European Journal of Operational Research **153**(1), 3–27 (2004)
6. Fuchs, G., Schimmelpfeng, K., Brunner, J.O.: A roadmap for integrating fairness in personnel planning and scheduling in hospitals. Operations Research, Data Analytics and Logistics **45**, 200479 (2025)
7. Legrain, A., Omer, J.: A dedicated pricing algorithm to solve a large family of nurse scheduling problems with branch-and-price. INFORMS Journal on Computing **36**(4), 1108–1128 (2024)
8. Rocha, M., Oliveira, J.F., Carravilla, M.A.: Cyclic staff scheduling: optimization models for some real-life problems. Journal of Scheduling **16**, 231–242 (2013)
9. Santos, G.H., Toffolo, A.T., Gomes, A.R.: Integer programming techniques for the nurse rostering problem. Annals of Operations Research **239**, 225–251 (2016)
10. Smet, P., Brucker, P., De Causmaecker, P., Vanden Berghe, G.: Polynomially solvable personnel rostering problems. European Journal of Operational Research **249**(1), 67–75 (2016)
11. Valouxis, C., Gogos, C., Goulas, G., Alefragis, P., Housos, E.: A systematic two phase approach for the nurse rostering problem. European Journal of Operational Research **219**(2), 425–433 (2012)
12. Wickert, T.I., Kummer Neto, A.F., Boniatti, M.M., Buriol, L.S.: An integer programming approach for the physician rostering problem. Annals of Operations Research **302**, 363–390 (2021)