Università degli studi di Pavia
Dipartimento di Matematica
Corso di Laurea Magistrale in Matematica

Fair Scheduling Models for Security Operations Centers

Tesi di Laurea Magistrale in Matematica

Relatore (Supervisor):
Prof. Stefano Gualandi
Correlatore (Co-Supervisor):
Dr. Valentina Morandi

Tesi di Laurea di:
Vittoria Crotti
Matricola 544809

Anno Accademico 2025-2026

Contents

Chapter 1

Introduction

Scheduling is a classic problem that is well known and studied in Operations Research. It consists of assigning activities to a limited number of available resources. Many scheduling problems have been discussed in the literature such as timetable scheduling, personnel scheduling, production scheduling and many others. The personnel scheduling problem consists of assigning tasks and/or work hours to employees in an optimized way. Typically, the objective is to maximize productivity, minimize labor costs or similar. Of course, many constraints may be involved. First of all, the number of employees scheduled must be sufficient to deliver the required service. For example, in a nurse rostering problem, a proper number of nurses must be assigned to each hospital department. Naturally, the closer the problem is to real-world conditions, the greater the number of constraints involved, making it very difficult to solve manually. Typically, personnel scheduling is classified as an *NP-hard* problem. This issue is very important for companies because producing a high-quality schedule requires a lot of managerial experience and a significant amount of time. Furthermore, employee requests for days off and legal regulations must also be taken into account. Therefore, solving a personnel scheduling problem in an optimized way can provide substantial benefits.

In particular, shift scheduling is one of the most difficult personnel scheduling problems. The problem consists of assigning shifts to a pool of operators in order to ensure an adequate level of service. Shift work is common in industries and sectors where operations cannot be interrupted. Many shift-based activities provide services 24 hours a day, such as hospitals, emergency services, and certain manufacturing sectors. According to *ISTAT* (the Italian National Institute of Statistics), in 2024 more than 3.2 million workers were employed in shift-work, consisting in around 13-14% of the Italian workforce. Therefore, studying this type of problem is highly relevant. Moreover, several additional constraints arise in shift scheduling. Examples include minimum rest periods between consecutive shifts, limits on the maximum number of consecutive working days, and other regulations imposed by labor laws. This thesis refers specifically to *CCNL* (Italian collective labor agreements). These constraints significantly reduce the flexibility of the scheduling model and are among the most restrictive requirements to satisfy.

Another important aspect to consider is that multiple tasks need to be performed, but not all employees possess the skills required for every task. For example, in a nurse rostering problem, some nurses may be qualified only for ward duties, while others may be qualified for both ward and operating room assignments. Therefore, the scheduling process must take employee qualifications into account. Once legal constraints, requests for days-off, and task requirements have been considered, the schedule can select workers only from a relatively limited pool of eligible employees.

An important aspect, although often underestimated, is the need of being precisely aware of the company's requirements in terms of coverage, intended as number of working employees. Usually, the schedule relies on the manager's perception of these needs. Even if is often sufficient, there may be patterns that can only be detected through an operational data analysis. In this thesis, a data-based approached is proposed and applied to a use case, see Chapter 3 for more details. The goal of the analysis is to obtained an hourly workload profile. Of course, the precise calculation varies from use case to use case. The workload profile can be fitted using different time-slot durations: the greater the granularity, the more precise the result. In this thesis, one hour time-slots were chosen. Once the profile is fitted, the number of required operators is established per hour, with the support of the manager, and will represent the entry demand for the model. The aim of the optimization model is to assign shifts, which are typically are 8 hours long. However, the demand for operators could vary a lot throughout the shift. The approach proposed in this thesis considers 1 hour time-slots, allowing operator demand to be represented more precisely. An appropriate number of operators is ensured for each time-slot by assigning them to shifts and identifying which shifts cover each time-slot. This is a more general approach. Although operator demand could be fitted over the entire shift, this would probably result in an overly flat representation. Moreover, this approach makes it easier to compare different types of shift schemes. Once the demand per hour is fixed, the set of shifts can be easily changed. Furthermore, in this thesis, the demand is computed for each day of the week. It is reasonable to assume that weekends and weekdays have different workloads. In some types of work, seasonality could play a role, but it was not considered in this thesis. In order to ensure the requested coverage, a penalty for lack of coverage is defined. It is represented as an increasing function with the percentage of lack of coverage. The model minimizes this penalty.

It is reasonable to assume that shift-based work heavily affects operators' quality of life, since it alters their sleep-wake cycle. In recent years, both companies and employees have become increasingly concerned about work-life balance. The goal of this work is to balance the company's needs in terms of operators with operators' needs and preferences by introducing the concept of *fairness*. Fairness can be computed in several ways and can prioritize different aspects, as further explained in Section 1.1. However, in general, it represents how positively a schedule is perceived by the operator. Of course, since fairness is a perception, it is not easy to define from a mathematical point of view. In this thesis, an approach based on employees' preferences is proposed. The idea is to make the employees rank the shifts from the worst to the best. Once the preferences of all employees are collected, they are used as input for the model. The underlying idea is to follow employees' preferences while guaranteeing the company's coverage needs. To do this, the model minimizes the *unfairness* throughout the pool of operators. Unfairness is measured through a penalty based on the difference between the highest preference score and the score of the assigned shift. Of course, preferences are followed only when possible. This could lead to high oscillations in perceived fairness. Therefore, the cumulative unfairness accumulated throughout the rolling horizon is also computed. The penalty is defined as a combination of *current* unfairness and *cumulative* unfairness. To ensure that no operator is penalized with respect to the others, a maximum unfairness threshold is set. In this way, implicit equality between the operators is guaranteed. The results presented in Chapter 4 show that introducing fairness aspects does not penalize coverage. Moreover, using an optimization model guarantees similar or even better coverage while significantly increasing fairness. Therefore, similar productive performance can be achieved with much happier employees. To summarize, coverage and fairness are the two fundamental aspects on which this work is based. In fact, the proposed optimization model is called *Fair Demand-Responsive Scheduling* model. The model was implemented in Python, using spreadsheets as data input.

This work was tested and developed on a use case. In fact, it arose from the specific needs of the company involved. Multiprotexion s.r.l. is a high-level security company located near Pavia. At Multiprotexion s.r.l., 25 control room operators work in shift 24/7, and their work is comparable to that of emergency call-center responders. Since they are required to provide a high-quality service, scheduling represents an important aspect. This collaboration was encouraged and made possible by the fact that the author of this thesis has been a Multiprotexion employee since 2023, while the co-supervisor is a consultant for the company. Since scheduling was becoming an issue, the idea was to develop a dedicated model. The workload curve was computed using real operational data, with the year 2024 used as a benchmark. The model was then tested on this particular use case. It should be noted that, although the model was developed in response to a company need, it can be applied to any shift-based work environment and is also able to schedule different tasks. The results obtained are presented in Chapter 4. Different scenarios, corresponding to different levels of importance assigned to fairness, were tested. Moreover, different shift schemes were analyzed, allowing the role of scheme *flexibility* to be investigated. The model provided satisfactory results, both in terms of coverage and fairness. Specific metrics were computed, as described in Section 4.1. The company has now introduced the model and is currently in a testing phase. In conclusion, this work was also the subject of an article submitted in collaboration with the co-supervisor Dr. Morandi and Prof. Speranza (University of Brescia).

In Chapter 2, the mathematical formulation of the problem is presented. The *FDS* model is introduced in Section 2.3, while its integration with legal constraints is discussed in Section 2.3.1. In Chapter 3, the data collection process and the case study are presented. The results, obtained under different scenarios and hypotheses, are reported in Chapter 4. Finally, conclusions and future developments are discussed in Chapter 5.

1.1 Literature review

In this section , a review of the scientific literature used for this thesis is given. A wide literature for personnel scheduling is available, it is a well known problem, studied even in universities optimization courses. In recent years, the literature on personnel scheduling and call center workforce management shows a progressive shift from purely cost-oriented optimization toward more human-centered planning. The traditional literature on call centers was focused on forecasting, queueing, and staff scheduling. The main issue is that the demand is very fluctuating during the day and uncertain. Therefore, the manager must schedule enough operators to handle the demand while controlling the labor cost. Call centers are especially labor-intensive environments, where agent staffing typically represents a major share of operating costs, and where operational decisions directly affect both customer waiting times and perceived service. Aksin et al. (2007) highlights that modern call centers are no longer only operational systems designed to answer calls efficiently. Technological developments, such as automatic callbacks, and the integration of sales activities have increased the work complexity. At the same time, call centers have become strategic in customer retention and satisfaction. Moreover, the authors highlight the importance of behavioral factors since the interaction between the operator and the customer can affect both service quality and system performance. Van den Bergh et al. (2012) provide a broad literature review of personnel scheduling problems. The article shows that scheduling research has expanded across many sectors and technical approaches. The review also classifies studies based on problem settings and techniques used. The main highlight is has become increasingly complex because companies consider employee needs, preferences, part-time contracts and flexible working arrangements, rather than treating labor only as a cost-minimization input. A central theme emerging in the later published articles is *fairness* in personnel scheduling. Wolbeck (2019) argues that shift work has a heavy impact on employee's private lives and a high dissatisfaction can lead to a reduction of job performance and absenteeism. As explained in this article, fairness is difficult to define because it depends on the perspective and the aspect that is more considered. For example, individual or group fairness, short-term or long-term fairness, preference satisfaction, equal distribution of the workload and many more. Therefore, a unique definition doesn't exist but the author suggest that fair scheduling should generally combine a consideration of individual preferences with an even distribution of workload. As mentioned before, in this thesis only the operator's preferences were considered. However, fairness was guaranteed both in short-time and long-time and a group equity also by upper bounding the maximum unfairness allowed. The balanced workload was not considered but actually the company is studying an algorithm to fairly assign workload. Uhde et al. (2020) provide a research on how employee define fairness in healthcare environments. In the article, interviews are reported. They ask employee how they would define fairness, from a human point of view.

In particular, two aspects are often set against: *equality* and *equity*. *Equality* refers to an even distribution of shifts, benefits, or burdens among employees, whereas *equity* implies that individual differences, previous contributions, or specific circumstances should be considered when allocating shifts. Therefore, while equality aims at treating employees in the same way, equity aims at producing a fair outcome by accounting for relevant differences between them. The provided interviews show that equality is often seen as an abstract idea, not perceivable while fairness is more perceived when individual needs are considered and negotiated. The study also shows that collaborative decision-making is perceived as fairer than decisions made autonomously by a system. This suggests that fair scheduling systems should not only optimize outcomes but also support transparency, negotiation, and team-based decision processes. Şimşek and Kundakcioglu (2022) apply fairness to contact center agent scheduling. They developed a mathematical model that minimizes operating costs related to labor, transportation, and lost customers while also incorporating agents' shift preferences as a measure of fairness. Their study explicitly evaluates the trade-off between employee satisfaction and total operating cost in a real contact center context.They show that fairer schedule can be produced by setting satisfaction bounds. Of course, the satisfaction bound depends on the chosen concept of fairness. They also point to limitations such as sudden demand changes.

The main contribution of this body of literature is the evolution from traditional workforce scheduling as a cost and coverage problem toward a broader decision-making framework that includes service quality, employee satisfaction, fairness, and organizational sustainability. The key point to highlight is that fairness is not a defined concept, but it depends on the aspects that are privileged in the specific use case. The literature suggests that future scheduling systems should combine mathematical optimization with human-centered design supporting collaborative decision-making. This is particularly important in service environments such as call centers, healthcare, and transport, where employee satisfaction and customer service quality are closely connected.

Chapter 2

Model formulation

2.1 Problem description and fundamental metrics

The problem we consider is a scheduling problem. The idea is to optimize the shift scheduling in order to guarantee a proper coverage level while improving operators' satisfaction, taking into account fairness. Both coverage and fairness need to be defined in a mathematical way. This is also necessary in order to evaluate the resulting schedule.

2.1.1 Definition of coverage demand

In 24/7 working environments, it is necessary to ensure a proper level of working operators in order to be able to provide reliable services. For example, in hospital-related context, a certain number of doctors is requested to be able to deal with emergencies. In this context, efficient shift scheduling is needed.

Often, the company knows how many operators are needed in each shift for each task. This number could also vary depending on the day of the week. In many cases it is lower during weekends and national holidays. Actually, a better estimation is based on time slots. Time slots are shorter than typical shifts. Therefore, an estimation based on time slots better reflects the actual demand for operators. Often, the workload, and consequently the demand for operators, follows recurring patterns throughout the day.

The most common shift duration is 8 hours. However, the workload is not always evenly distributed within a shift. For instance, it may be higher during the first hours and lower toward the end. In a 24/7 fast-food restaurant, for example, it is reasonable to expect peak workload during meal times. Since shifts have a fixed duration, when scheduling them, managers must ensure that the number of operators is sufficient to handle peak demand without leading to overscheduling. Therefore, staffing decisions should balance peak requirements with overall efficiency rather than rely on a simple average.

An approach based on time slots is more accurate and allows for the design of shifts tailored to the specific workload of the organization. This concept will be further explored in Chapter 3. See Figure 2.1 for an example of a comparison between the estimated operators' demand based on shifts and on time-slots. However, let's suppose we know

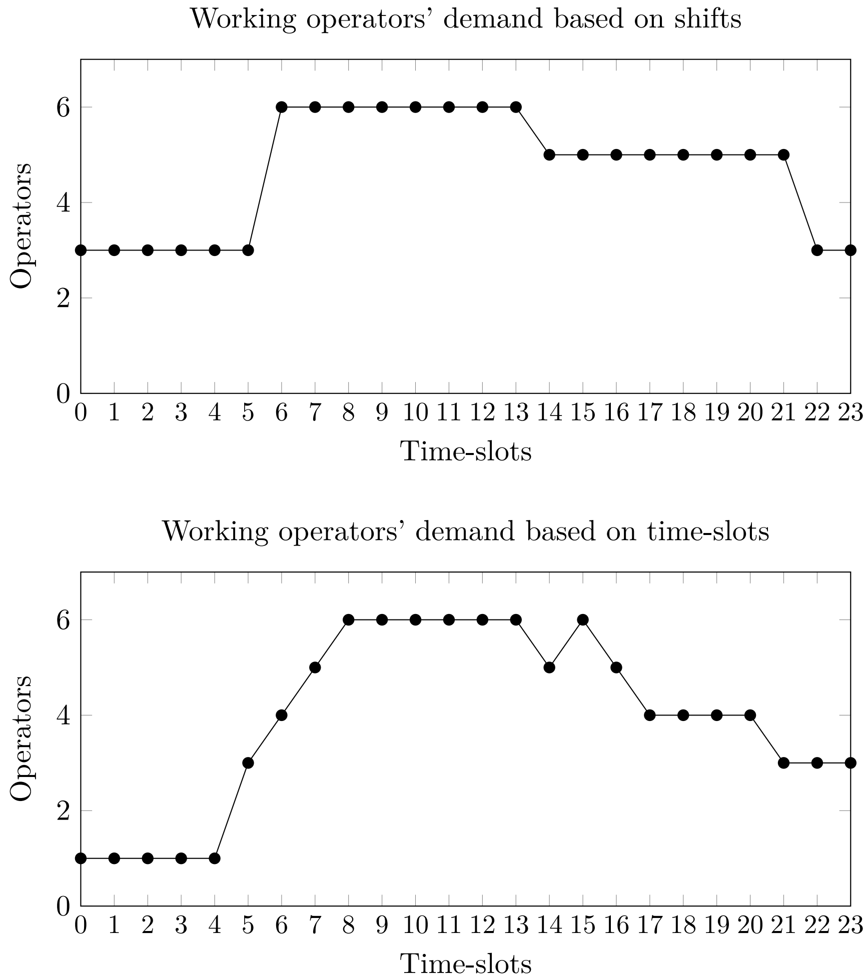


Figure 2.1: Operators' demand based on shifts and on time-slots

exactly the demand of working operators for each task and for each time slot of each day of the time horizon, that will be referred as *coverage demand*, is exactly known.
The model will try to maximize the coverage in each time slot of each day. Actually, it will try to minimize the *lack of coverage*. Therefore, the idea is to assign a penalty for the lack of coverage. Every time the agency demand is not satisfied, a penalty is computed.

A well designed calculation of the penalty is fundamental. The first thing to avoid is an unbalanced situation. In real life context, it is not always possible to completely satisfy the coverage demand. When is known in advance that is impossible to completely

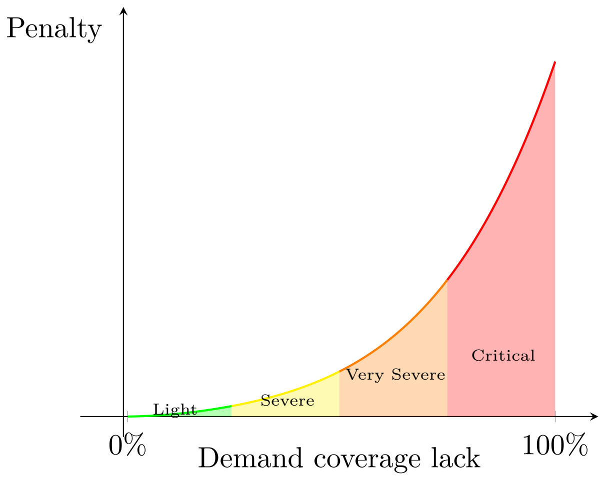


Figure 2.2: Penalties for the demand coverage lack

fulfill the demand, is better to distribute the lack of operators among multiple shifts. In this way, the company is more likely to provide a good service, and also the workload is distributed among the shifts. The underlying idea is to avoid borderline situation that tends to become critical. For example, if the minimum requirement for a time slot is 5 operators, is very risky to assign only 3. If a disruption occurs, for example an employee calling in sick, the company may be unable to provide the service. Moreover, such an arrangement is not efficient, as it creates an unbalanced distribution of resources. For example, let's consider time slot A and time slot B, both requiring 5 operators. An unbalanced schedule would assign 5 operators to time slot A and 3 operators to time slot B, resulting in uneven workload coverage. A more balanced solution would be to assign 4 operators to each time slot, reducing the risk of undercoverage and improving overall efficiency. If a linear calculation of the penalty would be taken into account, those two situations would be penalized in the same way, even if the first one is worse than the second. This is the reason why a non-linear penalty is needed. This also allows to heavily penalize critical situations. The proposed solution is to model the penalty as an increasing convex function of the lack of coverage, as represented in Figure 2.2. To approximate the increasing convex function, a piecewise linear function is used, as pictured in Figure 2.3. When the agency coverage demand is completely fullfilled the penalty is 0% and when no operator is assigned the penalty is 100%. More details of the mathematical calculation are explained further.

2.1.2 Definition of fairness

Since working on shifts affects heavily the operators' private life, the fairness measurement need to be strictly related to the shift assignment. The model will assign shifts minimizing the lack of coverage, while taking into account fairness. The fairness will

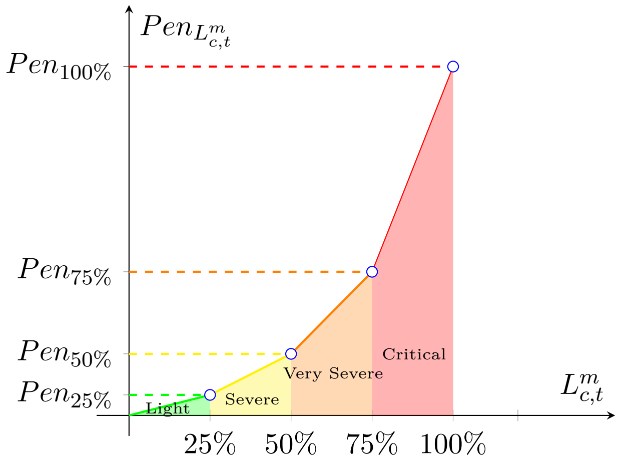


Figure 2.3: Penalty function structure

not be directly considered in the objective function but into constraints in order to force the model to produce a schedule that is fair enough. The unfairness cannot exceed a fixed threshold, therefore equity between operators is automatically ensured. The idea is to avoid unbalanced situations when operator A is heavily penalized and operator B is rewarded.

As a fairness metric, the company proposed to ask operators for their preferences. Those preferences are based not only on the shifts, but also on the day of the time horizon and also the task. This is made in order to best represent preferences of the operators in a realistic way. Compute fairness just based on the shift is too flat and semplicistic. Finding an assignment that balances such preferences allows the operators to manage their private life better and so to avoid unexpected leaves.

A good mathematical representation of those preferences is fundamental. One way could be to just assign score to the shifts for each task. This way could be misleading because the operators could assign highest score to the ones they like and very low scores to the ones they dislike, probably the medium-liked range would not be well represented. Hence it could be that fairness has very high fluctuations and is not representative of the perceived fairness. The proposed solution, is to rank shifts for each task. This will allow to better fit the real perceived fairness and a better modelization. Based on this ranking, an unfairness penalty is calculated as the difference between the highest in the rank and the assigned one. An example of raking is given in Table 2.1.

Another important aspect that must be taken into account is the perceived fairness during time. For example, if operator A was unsatisfied on the January scheduling it makes sense to reward him with a scheduling that suits his preferences on February. In order to be able to do this, as mentioned before, a concept of *past unfairness* must be introduced. The model will also take into account of the previous unfairness, in order to balance it with the current one. More details will be discussed further, in Section 2.2. However, the model will consider a combination between the previous unfairness and the current one.

Monday-task A	**Saturday-task B**	**Christmas-task C**
Afternoon	Morning	Night
Morning	Afternoon	Morning
Night	Night	Afternoon

Table 2.1: Shift scheduling example

2.2 Mathematical formulation of the problem

In this section, the quantities and variables of the model are presented.

First of all, a *time horizon* is needed. The first day is indexed as day zero. Let $T = \{0, \ldots, \mathcal{T}\}$ be the time horizon. As anticipated, the model is based on time slots, not on shifts. Let C denote the set of time-slots in which the time horizon T is partitioned. Typically, the considered time-slots are one hour long. This quantity is essential to know in which time slots which shifts are active. If time slots were perfectly aligned with shifts, the model would effectively be shift-based.An approach based on time slots is more flexible and general. As discussed in Chapter 4, it allows for better modeling and, consequently, a better fit to the coverage demand. For example, it allows to consider overlapping shifts that in general lead to better solutions. A set of shifts S is defined. This set represents all the possible shifts that can be assigned within the company. To know which shifts are are in which time slots, the boolean parameter C_s^c is introduced, where s is a shift and c is a time-slot. M is the set of tasks an operator can be assigned. It is assumed that an operator does not change tasks during a single shift.
I denotes the set of operators included in the schedule. The goal is to assign decision variables $x_{s,t}^{i,m} \in \{0, 1\}$, that takes value 1 if the operator i is assigned on the shift s of the day t attending the task m. First, the model needs to know which operators are available on which day t of the time horizon. To this end, the decision variable $w_{i,t} \in \{0, 1\}$ is introduced, taking value 1 if the day t is a rest day for operator i. The underlying idea is to schedule rest days in a convenient way. For instance, if Monday is understaffed, is convenient to assign Sunday as a rest day to an operator, so that they can work on Monday. It is also necessary to introduce parameters that capture specific characteristics and constraints of individual operators. For example, some operators might be unable

to work night shift for medical reasons and so on.. Those parameters are fundamental to make the model applicable in real-life contexts. The boolean parameter $Allow_i^{s,t}$ takes value 1 if the operator i can be assigned to the shift s on the day t. This parameter also ensures that pre-assigned days off are respected. Furthermore, in practice, not all operators are qualified to perform every task. The boolean parameter $Able_i^m$ takes value 1 if the operator i can perform the task m. In order to account for fairness, operators' preferences must be considered. Operators are asked to assign a score to each combination of shift, day and task. The aim is to accommodate their personal needs as much as possible. The day of the time horizon plays a key role because preferences could be different throughout the week due to personal commitments. The parameter $Score_{s,t}^{i,m}$ represents the score that the operator i assigned to the task m in the shift s on the day t. $Score_{s,t}^{i,m} \in [1, Score_{s,t,m}^{MAX}]$, where $Score_{s,t,m}^{MAX}$ is the maximum achievable score for that combination. Operators are asked to rank option from worst to best. Based on those scores, fairness is computed.

In the model, unfairness is minimized, therefore a penalty measure must be defined. The idea is that the unfairness is not only evaluated for the current schedule, but also incorporates the unfairness accumulated in previous schedules. The goal is to create a sort of equity among operators over time. The boolean parameter γ_i represents, for each operator, how much the previous unfairness is important to him with respect to the current unfairness. The current unfairness, according to Wolbeck (2019), is modeled as follows:

$$f_{i,t} = \sum_{m\in M}\sum_{s\in S}\left(\frac{Score_{s,t,m}^{MAX} - Score_{s,t}^{i,m}}{Score_{s,t,m}^{MAX}}\right)x_{s,t}^{i,m} \tag{2.1}$$

where $x_{s,t}^{i,m}$, as explained further, is the decision variable indicating if the operator i is assigned to the task m of the shift s on the day t. Essentially, the current unfairness is computed as the difference between the maximal score and the actually assigned one. The unfairness is computed as the current unfairness $f_{i,t}$ and the unfairness of the previous day $u_{i,t-1}$. The unfairness of the operator i on the day t is computed as follows:

$$u_{i,t-1} = \gamma_i f_{i,t} + (1-\gamma_i)u_{i,t-1} \tag{2.2}$$

Therefore, is a convex combination between the current unfairness and the previous one. Both $f_{i,t}$ and $u_{i,t}$ are decision variables of the model, representing the *total* unfairness and the *current* unfairness, given by the schedule released. To ensure a control of the unfairness, the parameter $\alpha_i \in [0,1]$ is defined for each operator. It represents the maximum threshold that unfairness can reach for each operator. In fact, if the operator i is very near the unfairness threshold α_1 he will not be penalized in the following day, probably the shift he doesn't like will be assigned to another operator.

In order to produce a schedule that is actually usable by an agency is necessary to be aware of the agency' needs. Two quantities are defined for this purpose. $OptOper_{c,t}^m > 0$ is the optimal number of operators needed during the time slot c of the day t to perform the task m. $MaxOper_{c,t}^m$ is the maximum number of operators that can be assigned to the task m during the time slot c of the day t. This is important, because we don't want to over-populate a shift. In real life context, with all of the uncertainty and law boundaries, is very difficult to reach an over-assignment but still can happen. A minimum number of operators required is not defined, since the objective function prevents the not satisfying the minimum number. Three decision variables regarding demand coverage are defined: $L^m_{c,t} \geq 0$, $PenL^m_{c,t} \geq 0$ and $\lambda^m_{c,t} \geq 0$. Lc, t^m represents the lack of coverage in the time-slot c of the day t regarding the task m. By convention, $L^m_{c,t} = 0\%$ when the number of operators scheduled is equal or greater to $OptOper^m_{c,t}$ and is $L^m_{c,t} = 100\%$ when no operator is assigned. This decision variable represents the percentage of not satisfied demand that is tolerated on a specific time-slot for a task m. In order to minimize the lack of coverage, the model tries to minimize a penalty regarding that. The decision variable $PenL^m_{c,t}$ is defined as the penalty gained on time-slot c of the day t for the task m. As explained before, the penalty is discretized in a piecewise function that is mimicking the convex function (see Figure 2.2 and Figure 2.3). The idea is to penalize in different ways different levels of lack of coverage. $\lambda^h_{m,c,t}$ is a continuous variable representing the amount of demand coverage lack assigned to the interval $[pen^{h-1}_{m,c,t}, pen^h_{m,c,t}]$.

See Table 2.2 for a summary of the quantities, parameters and decision variables presented above.

Table 2.2: Summary of quantities, parameters, and decision variables.

Category	**Quantities / Parameters**	**Decision variables**
Time discretization	$T = \{0, \ldots, T\}$ C M C^c_s	–
Operators	$Allow^{s,t}_i$ $Able^m_i$ I	$x^{i,m}_{s,t}$ $w_{i,t}$
Preferences and fairness	$Score^{i,m}_{s,t}$ $Score^{\max}_{s,t,m}$ α_i γ_i	$f_{i,t}$ $u_{i,t}$
Coverage demand	$OptOper^m_{c,t}$ $MaxOper^m_{c,t}$	$L^m_{c,t}$ $PenL^m_{c,t}$ $\lambda^h_{m,c,t}$

2.3 Fair Demand-Responsive Scheduling model formulation

In this section, the formulation of the model will be detailed. The model will be presented in two slightly different formulations. The first one, presented in this section, is the most general one. In the second version, some considerations about legal boundaries are added, see Subsection 2.3.1. In this work, the more general model will be referred as *Fair Demand-responsive Scheduling Model (FDS).*

First of all, the objective function is defined as the minimization of the penalties due to the lack of coverage demand.

$$\min \sum_{t\in T}\sum_{m\in M}\sum_{c\in C} PenL^m_{c,t} \tag{2.3}$$

$$PenL^m_{c,t} = \sum_{h=1}^{H} \frac{Pen^h_{m,c,t} - Pen^{h-1}_{m,c,t}}{L^h_{m,c,t} - L^{h-1}_{m,c,t}} \lambda^h_{m,c,t} \qquad c\in C, m\in M, t\in T \tag{2.4}$$

$$\sum_{h=1}^{n} \lambda^h_{m,c,t} = L^m_{c,t} \qquad c\in C, m\in M, t\in T \tag{2.5}$$

$$L^m_{c,t} \geq \frac{OptOper^m_{c,t} - \sum_{i\in I}\sum_{s\in S} Slots^c_s x^{i,m}_{s,t}}{OptOper^m_{c,t}} \qquad c\in C, m\in M, t\in T \tag{2.6}$$

$$\sum_{i\in I}\sum_{s\in S} Slots^c_s x^{i,m}_{s,t} \leq MaxOper^m_{c,t} \qquad c\in C, m\in M, t\in T \tag{2.7}$$

$$x^{i,m}_{s,t} \leq Able^m_i Allow^{s,t}_i \qquad s\in S, m\in M, i\in I, t\in T \tag{2.8}$$

$$\sum_{m\in M}\sum_{s\in S} x^{i,m}_{s,t} + w_{i,t} = 1 \qquad i\in I, t\in T \tag{2.9}$$

$$f_{i,t} = \sum_{m\in M}\sum_{s\in S} \left(\frac{Score^{MAX}_{s,t,m} - Score^{i,m}_{s,t}}{Score^{MAX}_{s,t,m}} \right) x^{i,m}_{s,t} \qquad i\in I, t\in T \tag{2.10}$$

$$u_{i,t} = \gamma_i f_{i,t} + (1-\gamma_i) u_{i,t-1} \qquad i\in I, t\in T \tag{2.11}$$

$$u_{i,t} \leq \alpha_i \qquad i\in I, t\in T \tag{2.12}$$

$$0 \leq \lambda^h_{m,c,t} \leq L^h_{m,c,t} - L^{h-1}_{m,c,t} \qquad c\in C, m\in M, t\in T,\ h\in H \tag{2.13}$$

$$x^{i,m}_{s,t} \in \{0,1\} \qquad s\in S, m\in M, i\in I, t\in T \tag{2.14}$$

$$w_{i,t} \in \{0,1\} \qquad i\in I, t\in T \tag{2.15}$$

$$L^m_{c,t} \geq 0 \qquad m\in M, c\in C, t\in T. \tag{2.16}$$

The objective function (2.3), to be minimized, is the sum over the time horizon, the set of tasks $m \in M$ and the time slots $c \in C$ the penalty gained due to the lack of coverage.

The penalty $PenL^m_{c,t}$, minimized in (2.3) is computed in the equation (2.4). $PenL^m_{c,t}$ define the value taken by the piecewise linear function for the time slot c, task m of the day t. Since the penalty function is discretized, it becomes necessary to ensure that is well defined. Constraint (2.13) makes sure that $PenL^m_{c,t}$ is covered by $\lambda^h_{m,c,t}$ variables. The constraints (2.4) and (2.13), along with the convexity of the objective function (2.3), ensures that the model assigns the decision variables $x^{i,m}_{s,t}$ in a way that covers the lower h-index interval first. In other words, the model assigns first the variables to cover the more critical time slots, the most uncovered ones. Constraint (2.6) computes the lack of coverage $L^m_{c,t}$ as the normalized difference between the agency request $OptOper^m_{c,t}$ and the actual number of operators assigned.
Constraint (2.8) ensures that operator i can be assigned to the shift s of the day t to attend the task m. Constraint (2.9) ensures that no operator is assigned if he is on a rest day. As explained before, constraints (2.10) and (2.11) compute the actual unfairness and the cumulative one. Moreover, contsraint (2.12) ensures that the unfairness is not above the fixed treshold α_i. Constraints (2.14), (2.15) and (2.16) define the domain of the decision variables.

2.3.1 FDS model integrated with legal requirements

In real-life use cases, the scheduling of operators needs to satisfy legal boundaries. For example, some of those requirements are about mandatory rest hours and rest days. Therefore, those requirements need to be incorporated in the model by introducing additional quantities, decision variables and constraints. In this way, the scheduling produced can actually take place.

The most common laws regarding working in shifts are about rest hours and rest days. A minimum number of rest hours are mandatory between a shift and the subsequent one. For example, in Italy, according to *CCNL* (Italian work law), 11 rest hours are required within subsequent shifts. To integrate this feature in the model, the quantity $Comp_{s,s'}$ is defined. $Comp_{s,s'}$ takes value 1 if the shifts $s, s' \in S$ are compatible in subsequent days, 0 otherwise. Examples of the matrix $Comp$ will be given in the use case presentation. Regarding rest days, the most common type of law states that an operator must work a maximum number of consecutive days and then rest for a minimum number of rest days. In other words, the operator follows a working days sequence followed by a rest day sequence, and then starts again. Those type of fixed sequences are very common in nurse rostering and similar contexts. To integrate this feature in the model two new parameters are required. Parameter n_i is the maximum number of consecutive working days for the operator i. k_i is the minimum number of consecutive rest days for operator i. Those quantities need to be defined for each operator because often operators are hired with different type of contract. Moreover, there is not a loss of generality because even work not based on shifts can be represented in this way. We can think to a common Monday-Friday from 9 to 5 as a sequence of 5 working days with a daily shift followed by 2 consecutive rest days. This allows to use this scheduling model even in contexts where different types of contracts are mixed.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F
W	W	W	W	W	W	R	R	W	W	W	W	W	W	R	R	W	W	W	W	W	W	R	R	W	W	W	W	W	W	R

Figure 2.4: Example of a 6 work day sequence alternated with 2 rest day sequence

By considering the working time as a sequence of consecutive work days and rest days, scheduling the shifts for an operator $i$means to determine in the first place when he is going to start the rotation. This is due to the fact that when he starts the rotation he needs to be scheduled for at maximum n_i days and then rest for at least k_i days. Therefore, decision variables about when he is going to start the work sequence and the rest sequence are needed. Decision variable $q_{i,t} \in \{0,1\}$ takes value 1 if operator i starts a work day sequence on day t. In the same way decision variable $z_{i,t} \in \{0,1\}$ takes value 1 if the operator i starts a rest day sequence on day t. To summarize, to satisfy legal requirements the model not only assign operators to shifts but first of all needs to decide when an operator is working or resting, meaning when he is starting a work sequence or a rest sequence.

To summarize, here are listed the quantity and decision variables to meet legal requirements.

- $Comp_{s,s'} \in \{0,1\}$
- $n_i \geq 0$
- $k_i \geq 0$
- $q_{i,t} \in \{0,1\}$
- $z_{i,t} \in \{0,1\}$

The *FDS* model integrated with legal requirements here is stated.

$$\min \Phi \equiv \sum_{t\in T}\sum_{m\in M}\sum_{c\in C} PenL^{m}_{c,t} + \delta \tag{2.17}$$

$$(2.4) - (2.16)$$

$$\sum_{m\in M} x^{i,m}_{s,t} \leq w_{i,t-1} + \sum_{m\in M}\sum_{s'\in S} Comp_{s',s} x^{i,m}_{s',t-1} \qquad i \in I, t \geq 1, s \in S \tag{2.18}$$

$$z_{i,t} \geq w_{i,t} - w_{i,t-1} \qquad t \in T, i \in I \tag{2.19}$$

$$\sum_{l=0}^{\min\{k_i-1,T-t\}} w_{i,t+l} \geq \min\{k_i, T-t\} z_{i,t} \qquad -k_i \leq t \leq T, i \in I \tag{2.20}$$

$$q_{i,t} \geq 1 - w_{i,t} - (1 - w_{i,t-1}) \qquad t \in T, i \in I \tag{2.21}$$

$$\sum_{l=0}^{\min\{n_i,T-t\}+1} w_{i,t+l} \geq q_{i,t} \qquad -n_i \leq t \leq T, i \in I \tag{2.22}$$

$$\delta = \epsilon(\sum_{i\in I}\sum_{t\in T} z_{i,t} + q_{i,t} + 2 * w_{i,t}) \tag{2.23}$$

$$z_{i,t} \in \{0,1\} \qquad i \in I, t \in T \tag{2.24}$$

$$q_{i,t} \in \{0,1\} \qquad i \in I, t \in T. \tag{2.25}$$

Constraint (2.18) ensures that the operator i is assigned to a shift s on day t that is compatible with the shift s' he was assigned on day s'. Moreover, it also checks if on the day $t-1$ the operator was on rest day, through the quantity $w_{i,t-1}$, then there is no necessity to check the compatibility. Constraint 2.19 ensures that the variable $z_{i,t}$ is equal to 1 if $w_{i,t} = 1$ and $w_{i,t-1} = 0$. Therefore, minimizing in the objective function the variable δ ensures that a rest sequence can only start when a rest day is assigned after a work day. Constraint 2.19 along with 2.20 ensures that when an operator is assigned to a rest day, he also rests for the remaining consecutive $k_i - 1$ days. Constraint 2.21 ensures that a work sequence starts only when a rest sequence is finished. Constraint 2.21 along with constraint 2.22 ensures that when the operator i starts the work sequence he will work for at the most the next consecutive $n_i - 1$ days. Parameter ϵ is a small positive weight ensuring feasibility of rest-sequence variable in constraint 2.23. The domains of the additional decision variables are defined in constraints 2.24 and 2.25.

Chapter 3

Case study and data collection

Multiprotexion s.r.l. is an italian company specialized in high-level security services, both for supply chain transportation and high-valued buildings such as logistics ones. The alarms generated by attempted robbery are sent to the *Alarm Receiving Center (ARC)*, where a group of 25 operators manage the alarms and eventually alert the emergency services. The *ARC* must provide continuous assistance, therefore, is working 24/7. The operators work in shifts. At the moment, three shifts take place in Multiprotexion: *Morning* from 6 a.m. to 2 p.m., *Afternoon* from 2 p.m. to 10 p.m. and *Night* from 10 p.m. to 6 a.m., as shown in Figure 3.1. In the further, this shifts' scheme will be referred at as *classical* scheme. At the moment, 25 operators are employed at the

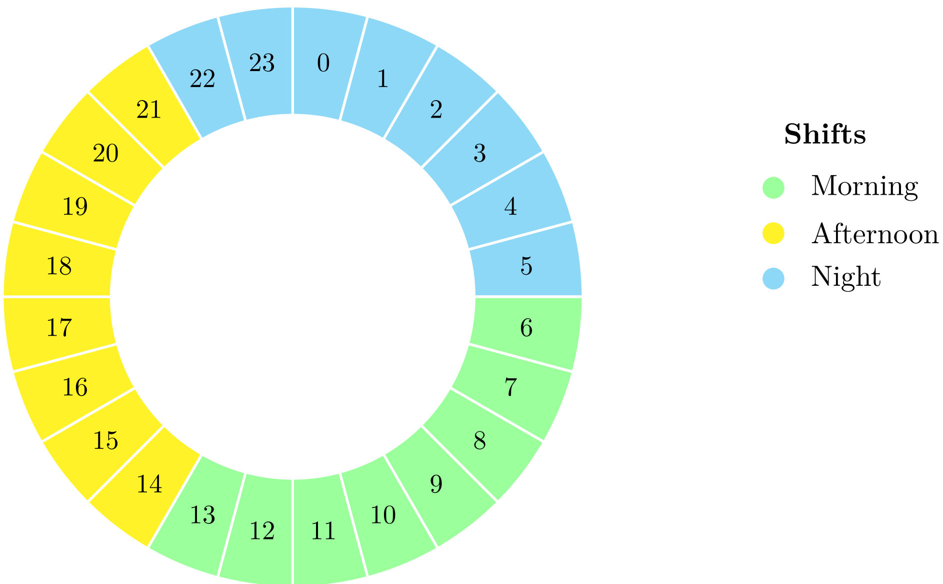


Figure 3.1: Shifts' distribution in Multiprotexion s.r.l.

ARC. According to their contracts, operators work for 6 consecutive days followed by 2 consecutive rest days. With this type of contract, operators can work any day of the week, including holidays. As explained further, operators are divided into junior and senior operators.

3.1 Data collection

First of all, a precise understanding of the company operators' demand is fundamental for developing a feasible schedule. This might seem obvious, but it is not. Typically, in small and medium-sized companies, knowledge of the company's needs is based not on data, but solely on managers' experience. Even though this experience is often quite accurate, data analysis can sometimes reveal patterns that are not immediately obvious. Therefore, the first question to address was: how can data be collected in order to predict *ARC* company's operator demand?

The first thing to do is understand how the business works. In Multiprotexion s.r.l. the work is the same for everyone, but junior operators cannot handle high-level security services. Therefore, two tasks are considered: *High-Level Security (HSL)* services and *Telematics* services. If an operator is allowed to handle *HSL* events, then the operator is also allowed to handle *Telematics* events, but not vice versa. 21 operators are able to handle both types while 4 can handle just *Telematics*, see Table 3.1 for more details. The company requires that at maximum one junior operator is assigned per shift, since

Senior Operators (*HSL*)	**Junior Operators (*Telematics*)**
#1 #2 #3 #4 #5 #6 #7 #7 #8 #9 #10 #11 #12 #13 #14 #15 #16 #17 #18 #19 #20 #21	#22 #23 #24 #25

Table 3.1: Division of operators according to their tasks

he needs to be supervised. On the other hand, at maximum 12 senior operators can work on the same shift, since is the number of seats available.

The *Telematics* workload is a very small percentage of the total workload. Therefore, in the following, the *HSL* workload is presented and referred as workload. In Multiprotexion the *ARC* work is divided in three categories: *managing alarms, incoming calls* and *outgoing calls*. Those actions can sometimes take place simultaneously and can be related, therefore they're not considered as three separated sub-tasks. The *incoming calls* are calls received from customers with various requests. These might be seem random and unpredictable, since they depend on customer behavior, but the data shown they still follow a pattern. The *outgoing calls* are calls made by operators to contact customers, and most of the times they help to manage an alarm. The *managing alarms* are alarms received by the *ARC* that operators handle by following certain actions dictated by the protocol. Approximately 99,9% are false alarms and follow precise daily patterns. This is because alarms are generated by devices installed on trucks. Therefore, they reflect the time distribution of the goods transportation. During the hours of the day when the transportation activity is higher, a lot significantly larger number of alarms is received. Furthermore, the type of goods being transported also creates specific patterns. For example, trucks transporting fresh goods to supermarkets usually make deliveries during the early hours of the morning, so it is rare for them to generate alarms in the evening. Moreover, the day of the week must also be considered. Not only do Saturdays and Sundays have workloads that differ from weekdays, but each weekday may also exhibit a distinct pattern. For example, many international deliveries start during the early hours of Tuesday. Since a lot of variables and different patterns take place, the data analysis must be as granular as possible, considering the hour and even the day of the week, public holidays and public holiday eves. Of course, those measurements need to be repeated several times during the year, the workload can increase or decrease throughout the year due to the company behavior. This is an additional reason why is more correct to base the shift assignment on time-slots and not on shifts. The time slots are more accurate and they tend to better fit the real situation. For this analysis, 1 hour time slot has been used.

As explained before, operators perform three types of work: *incoming calls, outgoing calls*, and *alarm management*. By extracting queries from the company database, the average number of alarms, incoming calls, and outgoing calls handled by the *ARC* for each day of the week and each hour of the day was calculated. By summing these quantities, the average workload was obtained. See Figure 3.2 for an example of the average workload.

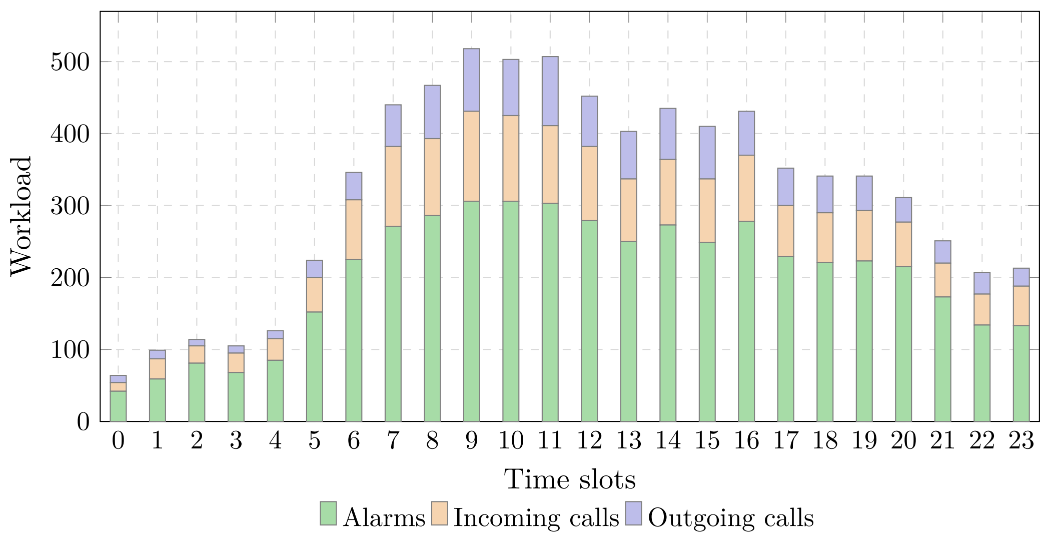


Figure 3.2: Example of average workload.

Once the average workload is computed, it is divided it in percentiles: 0%-25%, 25%-50%, 50%-75% and 75%-100%. This is done to rescale the data and be able to confront different months. It perfectly shows on which time slots there is the workload peak. See Figure 3.3 for an example.

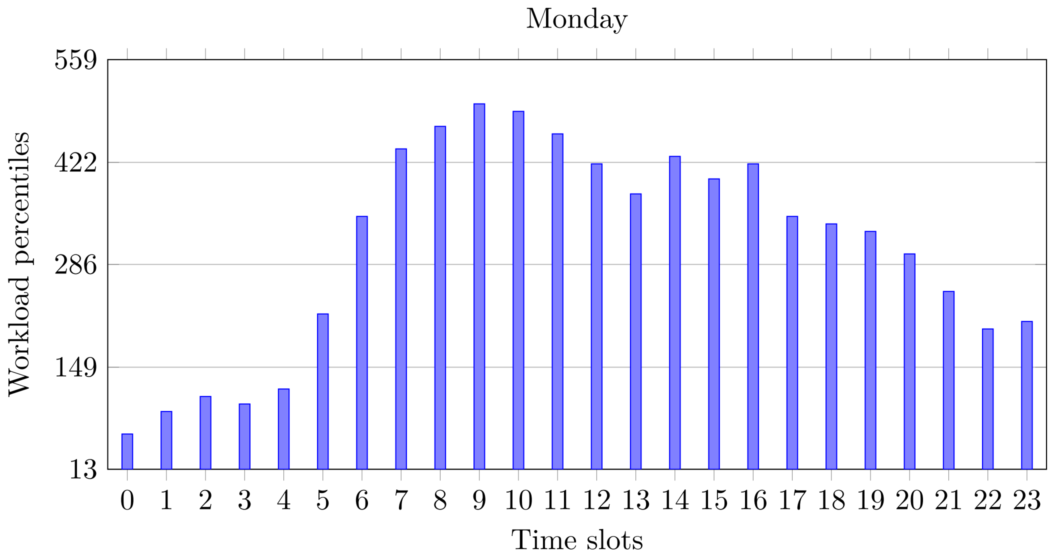


Figure 3.3: Average workload in time slots

The number of operators demanded must be sufficient to handle the workload, even considering breaks. In the following pictures, the estimanted operators'demand is presented, see Figure 3.4. Of course, those demands were approved by the company.

Once known the company needs in terms of working operators'demand, operators' preferences must be collected in order to consider fairness. Of course, if an operator cannot attend one shift, that shift will be the worst in the rank. It will not affect the penalties because it will never be assigned to that shift due to the quantity $Allow_i^{s,t}$. Operators'prefrences were collected via Google Form. As an example, preferences for *HSL* on Monday, are shown in Table 3.2

3.1.1 Overlapping shifts

By analyzing the results, particularly the workload profiles, it becomes clear that the 8-hour shifts currently used in Multiprotexion do not fit the agency demand very well. Although the shift length must legally be fixed at 8 hours, these shifts are too rigid and not flexible. In particular, the demand expressed in terms of shifts cannot accurately approximate the demand observed across individual time slots. As a result, some time slots become overstaffed, while others remain understaffed during the same shift. See Figure 3.5 for an example. Moreover, with these *classical* shifts, handling disruptions such as sick leave replacements becomes very difficult. In the Multiprotexion use case, an operator must rest for at least 11 hours between consecutive shifts. As a consequence,

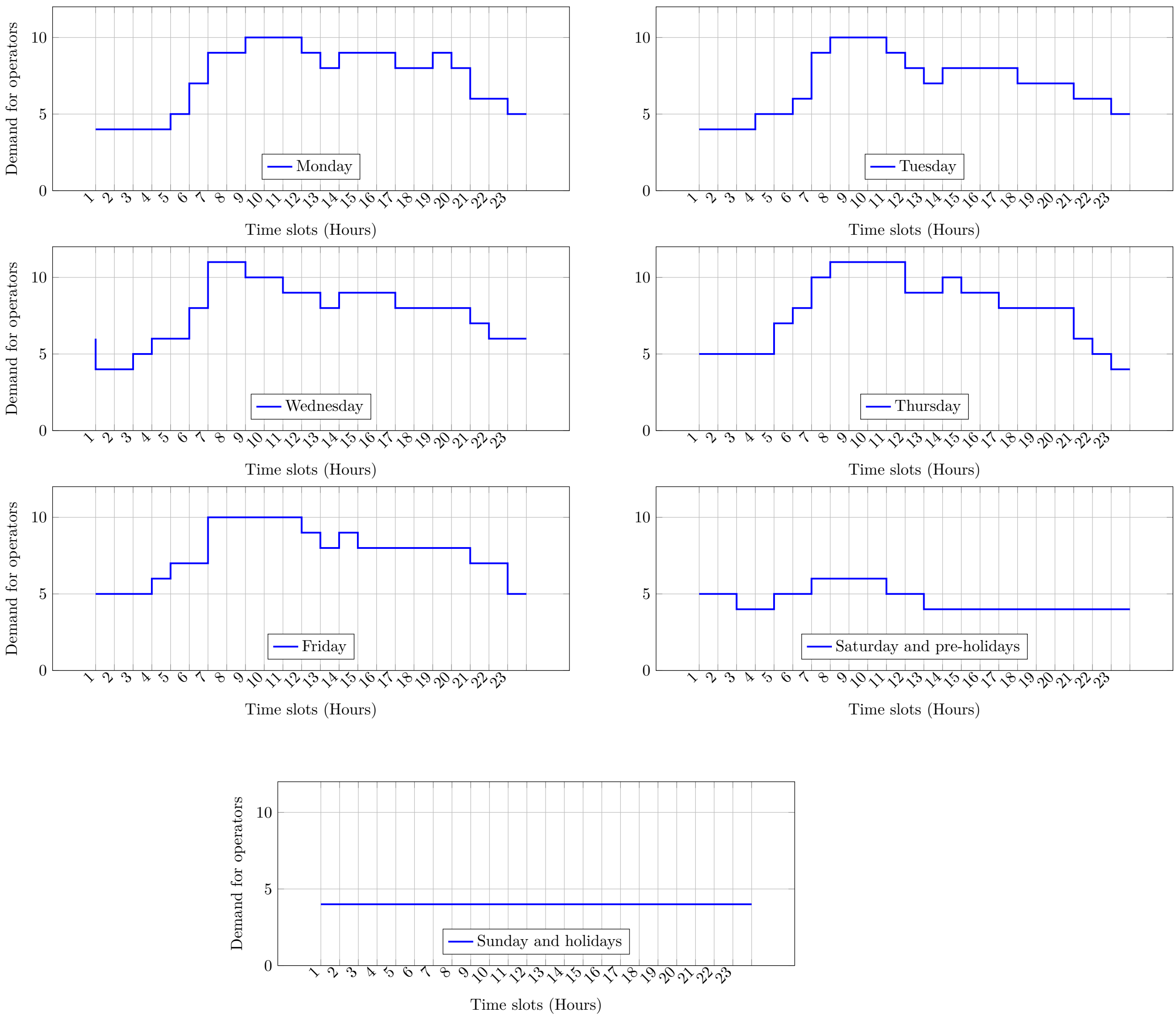


Figure 3.4: Daily demand for operators

Operator #1	Operator #2	Operator #3	Operator #4	Operator #5
A	N	M	A	M
N	A	A	N	A
M	M	N	M	N

Operator #6	Operator #7	Operator #8	Operator #9	Operator #10
N	M	A	N	M
M	N	M	M	A
A	A	N	A	N

Operator #11	Operator #12	Operator #13	Operator #14	Operator #15
A	N	A	A	A
N	A	M	N	M
M	M	N	M	N

Operator #16	Operator #17	Operator #18	Operator #19	Operator #20
A	A	A	A	M
M	M	M	M	A
N	N	N	N	N

Operator #21	Operator #22	Operator #23	Operator #24	Operator #25
N	N	N	M	N
M	A	A	N	M
A	M	M	A	A

Table 3.2: Operators' preferences for *HSL* task on Monday

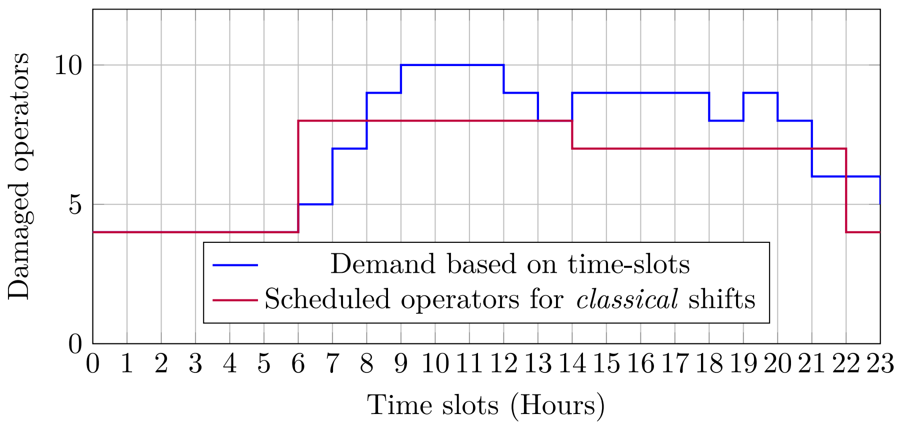


Figure 3.5: Comparison between the demand for operators on Monday and operators actually scheduled with *classical* shifts

many shifts are not mutually compatible. For example, if an operator assigned to the morning shift is absent, they can only be replaced by an operator who either rested on the previous day or also worked the morning shift on the previous day. The resulting compatibility matrix, denoted by $Comp_{s,s'}$, is reported in Table 3.3. In this table, the shift s worked on the previous day is represented by the rows, while the shift s' assigned on the following day is represented by the columns.

	M	A	N
M	1	1	1
A	0	1	1
N	0	0	1

Table 3.3: Representation of shifts' compatibility in subsequent days

Therefore, the idea is to introduce further shifts that can better match demand. These additional shifts must overlap with the *classical* ones in order to achieve finer control over the number of operators working in each time slot. Three auxiliary shifts are considered to evaluate the extent to which the model's results improve. The auxiliary shifts are: *Late Morning*, from 7 a.m. to 3 p.m.; *Late Afternoon*, from 3 p.m. to 11 p.m.; and *Daily*, from 9 a.m. to 5 p.m. In the following, the shift configuration composed of both the classical and the auxiliary shifts is referred to as the *additional* scheme. Figure 3.6 provides a representation of the *additional* scheme. In this setting the compatibility

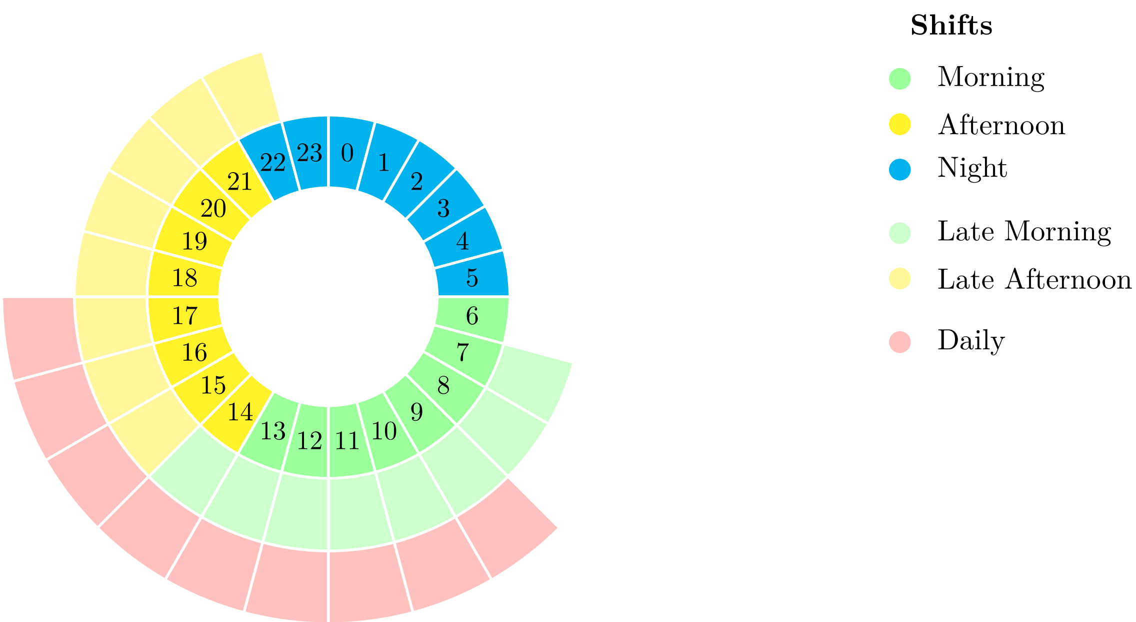


Figure 3.6: *Additional* scheme

matrix is shown in Table 3.4.

	M	A	N	LM	D	LA
M	1	1	1	1	1	1
A	0	1	1	1	1	1
N	0	0	1	0	0	0
LM	1	1	1	1	1	1
D	1	1	1	1	1	1
LA	0	1	1	0	0	1

Table 3.4: Representation of *additional* shifts' compatibility in subsequent days

Therefore, the operators'preferences were collected. They were asked to rank the *classical* shifts, as before, and then to do a separate ranking for the *additional* ones. They weren't asked to perform a singular ranking because the *additional* shifts are just a shifting of the *classical* ones of not so many hours. Therefore, a singular ranking would produce too much penalty that does not reflect the actual situation. For example, *Late Morning* shift is just a 1 hour shift of the *Morning* shift. See Table 3.5 for an example. Of course, in Chapter 4, the computational results will first be presented using only the *classical* shifts and then using the *additional* ones. This approach also allows us to evaluate the benefits of introducing auxiliary shifts.

The first advantage is the availability of a more flexible and diversified scheduling scheme that can better match the operators' demand. Moreover, overlapping shifts make it possible to better cover shift-change periods, which are always critical moments.

Since the first results were promising, the idea of introducing different shifts appeared to be effective. Therefore, to further investigate this concept and evaluate the actual benefits of increasing the number of shift types, three additional shifts are considered: the *Early Afternoon* shift from 12 p.m. to 8 p.m., the *Early Night* shift from 8 p.m. to 4 a.m., and the *Split Daily* shift from 8 a.m. to 12 p.m. and then from 4 p.m. to 8 p.m. While the previous three auxiliary shifts represented a forward shift with respect to the *classical* ones, these new shifts represent a backward shift. In the following, the shift configuration composed of both the *additional* shifts and the auxiliary shifts is referred to as *flexible* scheme. See Figure 3.7 for a representation of the *flexible* scheme. The idea of introducing the split shift arises from the presence of workload peaks during the early morning hours and the late afternoon hours. By assigning an operator to this shift, it becomes possible to cover both peaks with a single operator, whereas under the classical scheme two different operators working separate shifts would be required.

Moreover, this type of shift may be particularly attractive to operators, as it allows for a long lunch break. Another advantage is that it simplifies the management of shift substitutions. In this case, the compatibility matrix is reported in Table 3.6.

The results obtained by adding those shifts will be presented in Chapter 4. In the same way as before, operators were asked to to separately rank those new three shifts, see Table 3.7.

To summarize, we will use those names to indicate the group of shifts involved in the scheduling:

Operator #1	Operator #2	Operator #3	Operator #4	Operator #5
A	A	A	M	N
M	M	M	N	A
N	N	N	A	M
D	D	D	LA	D
LA	LM	LA	LM	LM
LM	LA	LM	D	LA

Operator #6	Operator #7	Operator #8	Operator #9	Operator #10
A	M	A	M	N
M	N	M	N	M
N	A	N	A	A
LM	LA	LM	LM	LM
D	LM	LA	LA	LA
LA	D	D	D	D

Operator #11	Operator #12	Operator #13	Operator #14	Operator #15
A	N	M	N	N
M	A	A	A	A
N	M	N	M	M
D	LA	LM	LA	LM
LA	LM	D	D	D
LM	D	LA	LM	LA

Operator #16	Operator #17	Operator #18	Operator #19	Operator #20
M	M	N	N	N
N	N	A	M	A
A	A	M	A	M
D	LA	LM	LA	D
LA	D	LA	D	LM
LM	LM	D	LM	LA

Operator #21	Operator #22	Operator #23	Operator #24	Operator #25
M	M	M	N	A
N	A	A	M	N
A	N	N	A	M
LA	LA	LA	LA	LA
LM	D	D	D	D
D	LM	LM	LM	LM

Table 3.5: Operators' preferences for *HSL* task on Monday

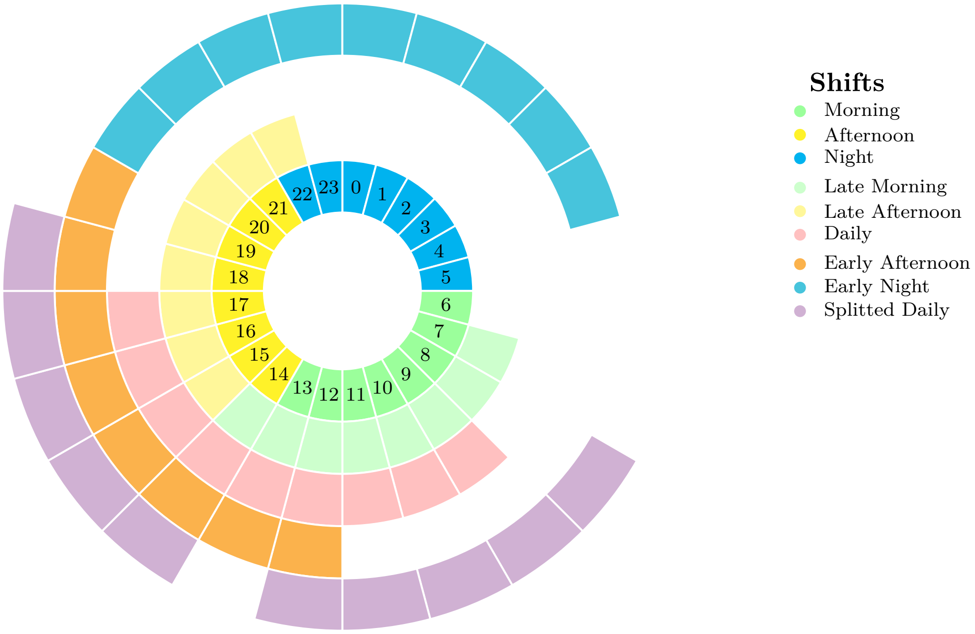
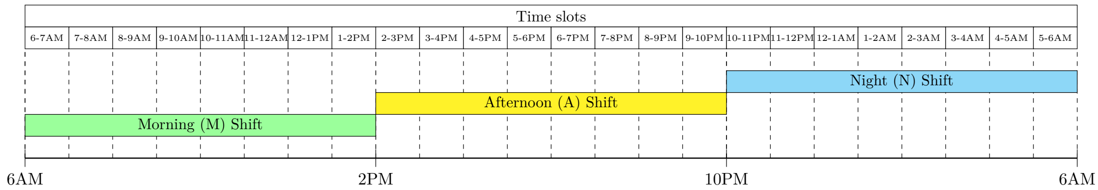


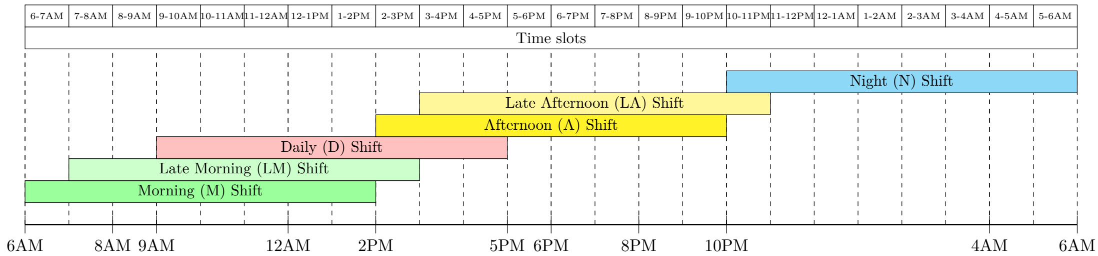
Figure 3.7: *Flexible* scheme

- Classical: *Morning, Afternoon, Night*
- Additional: *Morning, Afternoon, Night, Late Morning, Late Afternoon, Daily*
- Flexible: *Morning, Afternoon, Night, Late Morning, Late Afternoon, Daily, Early Afternoon, Early Night, Splitted daily*

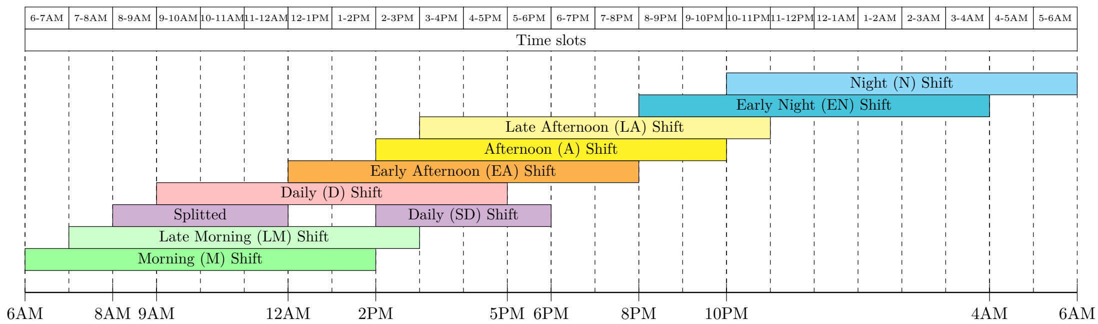
To better understand the difference between the three asset of shifts see Figure 3.8



(a) The *classical* scheme



(b) The *additional* scheme



(c) The *flexible* scheme

Figure 3.8: Comparison of the three shift scheduling schemes.

	M	A	N	LM	D	LA	SD	EA	EN
M	1	1	1	1	1	1	1	1	1
A	0	1	1	1	1	1	0	1	1
N	0	0	1	0	0	0	0	0	1
LM	1	1	1	1	1	1	1	1	1
D	1	1	1	1	1	1	1	1	1
LA	0	1	1	0	0	1	0	1	1
SD	1	1	1	1	1	1	1	1	1
EA	0	1	1	1	1	1	1	1	1
EN	0	0	1	0	0	1	0	0	1

Table 3.6: Representation of *flexible* shifts' compatibility in subsequent days

Operator #1	Operator #2	Operator #3	Operator #4	Operator #5
A	A	A	M	N
M	M	M	N	A
N	N	N	A	M
D	D	D	LA	D
LA	LM	LA	LM	LM
LM	LA	LM	D	LA
EN	EN	EA	SD	EN
EA	SD	SD	EN	EA
SD	EA	EN	EA	SD

Operator #6	Operator #7	Operator #8	Operator #9	Operator #10
A	M	A	M	N
M	N	M	N	M
N	A	N	A	A
LM	LA	LM	LM	LM
D	LM	LA	LA	LA
LA	D	D	D	D
EA	EN	SD	SD	EA
SD	SD	EA	EN	EN
EN	EA	EN	EA	SD

Operator #11	Operator #12	Operator #13	Operator #14	Operator #15
A	N	M	N	N
M	A	A	A	A
N	M	N	M	M
D	LA	LM	LA	LM
LA	LM	D	D	D
LM	D	LA	LM	LA
SD	EA	SD	EN	SD
EA	SD	EA	EA	EA
EN	SD	EN	SD	EN

Operator #16	Operator #17	Operator #18	Operator #19	Operator #20
M	M	N	N	N
N	N	A	M	A
A	A	M	A	M
D	LA	LM	LA	D
LA	D	LA	D	LM
LM	LM	D	LM	LA
EA	EA	EA	SD	EN
EN	SD	SD	EA	EA
SD	EN	EN	EN	SD

Operator #21	Operator #22	Operator #23	Operator #24	Operator #25
M	M	M	N	A
N	A	A	M	N
A	N	N	A	M
LA	LA	LA	LA	LA
LM	D	D	D	D
D	LM	LM	LM	LM
EN	EA	EN	SD	SD
SD	EN	EA	EA	EA
EA	SD	SD	EN	EN

Table 3.7: Operators' preferences for *HSL* task on Monday

Chapter 4

Computational results

n this chapter, the computational results are presented. Section 4.1 introduces the metrics used in the analysis. Then, Section 4.2 presents the results obtained by applying the model to the *classical* scheme. In this way, the benefits of introducing the model into the state of the art can be evaluated. Finally, Section 4.3 shows the results obtained by applying the model to the *additional* and *flexible* schemes. The results refer to a rolling horizon covering one year, from January 2024 to December 2024. The benchmark used for comparison is the actual scheduling adopted by Multiprotexion s.r.l. during the same period. This schedule is used to compute the benchmark values.

4.1 Metrics used

In this section, the metrics used to evaluate the model are presented. First, the value of the objective function is computed, both on a daily basis, Φ_t, and as an average over the entire rolling horizon, Φ.

Then, two aspects need to be evaluated: demand coverage lack and unfairness. To assess the demand coverage lack, all metrics are computed as averages. This approach is appropriate because summing the shortages would not allow us to distinguish between a situation in which the shortage is evenly distributed and one in which it is highly unbalanced. Three metrics are proposed: $Ave.L_c$, the average demand coverage lack for each time slot $c \in C$; $Ave.L_t$, the average demand coverage lack for each day $t \in T$; and $Max.L_t$, the maximum demand coverage lack for each day $t \in T$. By introducing $Max.L_t$, it is possible to capture the difference between the average situation and the daily situation, thereby ensuring that there are no excessively critical days.
To evaluate unfairness, four statistics are proposed: the average experienced daily unfairness $Ave.F_t$, the average experienced cumulative unfairness $Ave.U_t$, the maximum experienced daily unfairness $Max.F_t$ and the maximum experienced cumulative unfairness $Max.U_t$. Daily and cumulative unfairness are analyzed separately rather than combined through a convex combination, as done in the model. This separation makes it possible to evaluate the model's performance with respect to both daily and cumula-

tive unfairness individually. Otherwise, situations in which one measure performs poorly while the other performs well could be masked by an overall acceptable combined value. Moreover, this approach allows us to assess how the model improves each measure independently. As before, average measurements are presented together with maximum metrics in order to capture both the general behavior of the model and the worst-case situations, ensuring that there are no excessively penalized days. A summary of the metrics used can be found in the bullet-point list below.

- $\Phi_t = \frac{1}{|M||C|} \sum_{m \in M} \sum_{c \in C} PenL^m_{c,t}$: the daily objective function value
- $\Phi = \frac{1}{|M||C||T|} \sum_{t \in T} \sum_{m \in M} \sum_{c \in C} PenL^m_{c,t}$: the average objective function value
- $Ave.L_c = \frac{1}{|T|} \sum_{m \in M} L^m_{c,t}$: the average demand coverage lack in the time slot $c \in C$
- $Ave.L_t = \frac{1}{|C|} \sum_{m \in M} L^m_{c,t}$: thae daily average demand coverage lack
- $Max.L_t = \max_{|C|} \max_{|M|} L^m_{c,t}$: the daily maximum demand coverage lack
- $Ave.F_t = \frac{1}{|I|} f_{i,t}$: the average experienced daily unfairness
- $Ave.U_t = \frac{1}{|I|} u_{i,t}$: the average experienced cumulative unfairness
- $Max.F_t = \max_{|I|} f_{i,t}$: the maximum experienced daily unfairness
- $Max.U_t = \max_{|I|} u_{i,t}$: the maximum experienced cumulative unfairness

4.2 Results with *classical* scheme

In this section, the results obtained by applying the model to the *classical* scheme are presented to evaluate the benefits of introducing the model into the state of the art. Regardless of the results presented further, a first clear advantage is the time saving. Currently, scheduling is performed manually by the manager and requires approximately one working day. Moreover, the results produced by the model are perfectly usable. The schedules obtained for the first two weeks of January 2024 are presented in the tables shown in Appendix .1. In Appendix .1 are also presented examples of shift scheduling produced by the *FDS* model considering the three schemes.

The model was applied to the 14-day planning horizon using the *FDS* model on the *classical* scheme and compared against the Benchmark schedule adopted by the company. Figures 4.1-4.4 show the corresponding performance statistics. Since the maximum cumulative unfairness is an important parameter of the model, different values of this parameter have been tested to compare different scenarios. Since the optimization on the first two weeks of January 2024 is shown as an example to better understand the behavior of the model, results presented in Figures 4.1-4.4 have been produced under the assumption of a *Fair* level of maximum cumulative unfairness. The different scenarios will be discussed further. Of course, results obtained on all 2024 year will be shown further, in order to test the robustness of the optimization framework under realistic long-term dynamics. This setting captures seasonal demand fluctuations, varying operator availability, and the accumulation of fairness effects over consecutive scheduling rounds. First of all, the value of the daily objective function Φ_t has been evaluated. Figure 4.1 shows the results obtained for the first two weeks of January 2024. In most cases, the results obtained with the *FDS* model are better than those of the benchmark, as the value of the objective function is generally lower. Moreover, the objective function shows a more stable behavior, suggesting that the schedule produced by the *FDS* model is more balanced across the days. Figure 4.2 presents the average lack of coverage across the time slots $Ave.L_c$. As shown, the values of$Ave.L_c$ obtained with the *FDS* model are lower than, or at worst equal to, those of the benchmark. Therefore, the *FDS* model provides better results across all time-slots. This indicates that the model performs in a balanced way throughout the entire day. It should be noted that the maximum lack of coverage reached by the the benchmark is around 12%, which is already quite low. Therefore, since the benchmark already provides good results, a large percentage improvement should not be expected. The *FDS* model reaches the feasible optimum while satisfying all the imposed constraints. In fact, in the time slots where the benchmark reaches its maximum lack of coverage, the *FDS* model obtains the same value, suggesting that these choices are imposed by the constraints and cannot be further improved. Figure 4.3 shows the average daily unfairness, denoted by $Ave.F_t$, measured during the first two weeks of January 2024. The daily unfairness computed by the *FDS* model is consistently lower than the one obtained with the benchmark. Moreover, the improvement is significant. The comparison also highlights an interesting pattern. When the benchmark daily unfairness remains relatively stable, the unfairness produced by the *FDS* model follows a similar trend, but at a lower level. However, when the benchmark presents peaks in daily unfairness, the *FDS* model tends to smooth them out, producing much lower values. This suggests that the *FDS* model is more effective in balancing the distribution of unfairness across operators and days. Figure 4.4 shows the average cumulative unfairness, denoted by $Ave.U_t$, measured during the first two weeks of January 2024. The unfairness obtained with the *FDS* model is significantly lower than that of the benchmark. In general, the average cumulative unfairness of the *FDS* model follows the same pattern as the benchmark, but at a significantly lower level.

As mentioned above, the results obtained using a rolling horizon over the entire year 2024 are now presented. In order to better understand the behavior of the model, different scenarios are compared. The maximum level of cumulative unfairness is an important constraint, as it can significantly affect the resulting schedule. Therefore, different scenarios are presented: *Equitable*, *Fair* and *Lightly Fair*, corresponding to different values of the parameters α_i. The values of the α parameters are reported below:

- *Equitable*: $\alpha = 0\%$

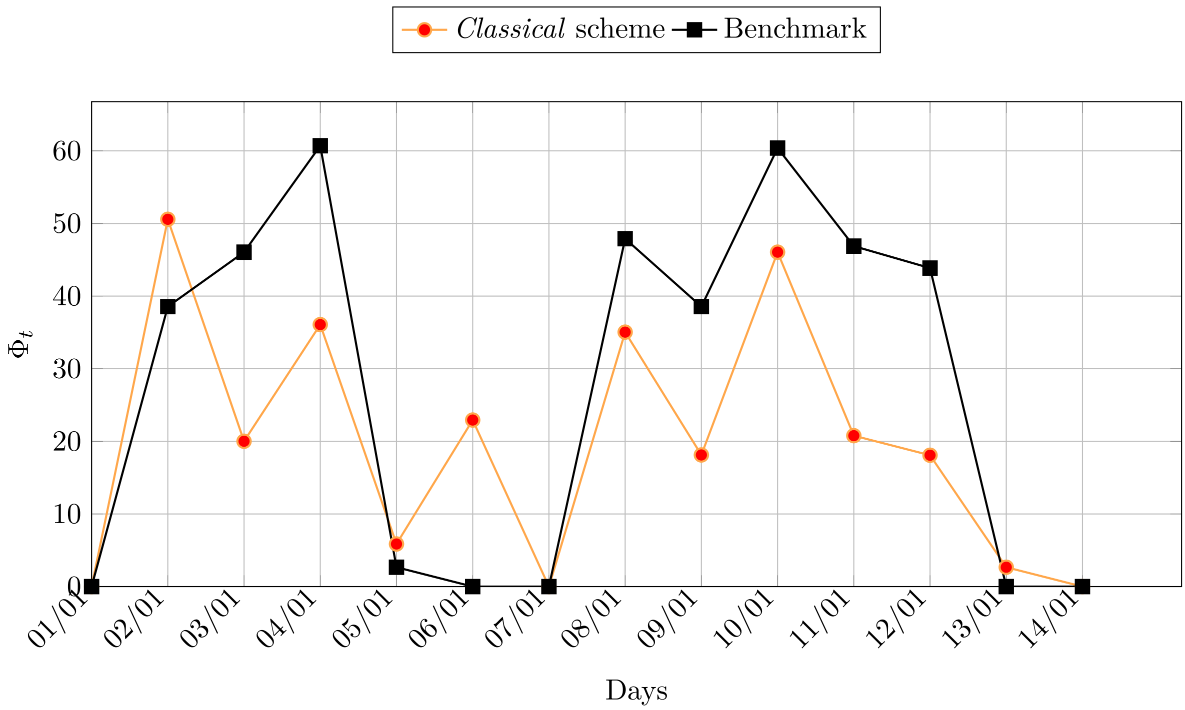


Figure 4.1: Objective function value Φ_t under *Fair* level of maximum cumulative unfairness

- *Fair*: $\alpha = 40\%$
- *LightlyFair*: $\alpha = 20\%$

In Figure 4.5 the horizontal black line labeled as B represents the value of the corresponding parameter computed on the benchmark. In Figure 4.5, the values of the average objective function Φ_t, the average lack of coverage $.L_t$ and the average cumulative unfairness $Ave.U_t$ are presented. As shown in Figure 4.5a, the objective function value is significantly higher in the *Equitable* scenario. This result is expected, since requiring cumulative unfairness to be zero forces the model to assign operators to the shifts they prefer, rather than optimizing the overall solution. The objective function value is around three times higher than the values obtained in the other scenarios. Therefore, the *Equitable* is not convenient. Conversely, only a small difference can be observed between the objective function values of the *Fair* and *LightlyFair* scenarios. Regarding the average lack of coverage $Ave.L_t$, shown in Figure 4.5b, the value obtained in the *Equitable* scenario is twice as high as those obtained in the *Fair* and *LightlyFair* scenarios. Again, from this point of view, there is no substantial difference between *Fair* and *LightlyFair*. The results for average cumulative unfairness are shown in Figure 4.5c. As expected, the average cumulative unfairness is 0% for the *Equitable* scenario. In this case, the difference between *Fair* and *LightlyFair* is more evident: $Ave.U_t$ is half as high in the *Fair* scenario as in the *LightlyFair* scenario. This is fully consistent with the values chosen for

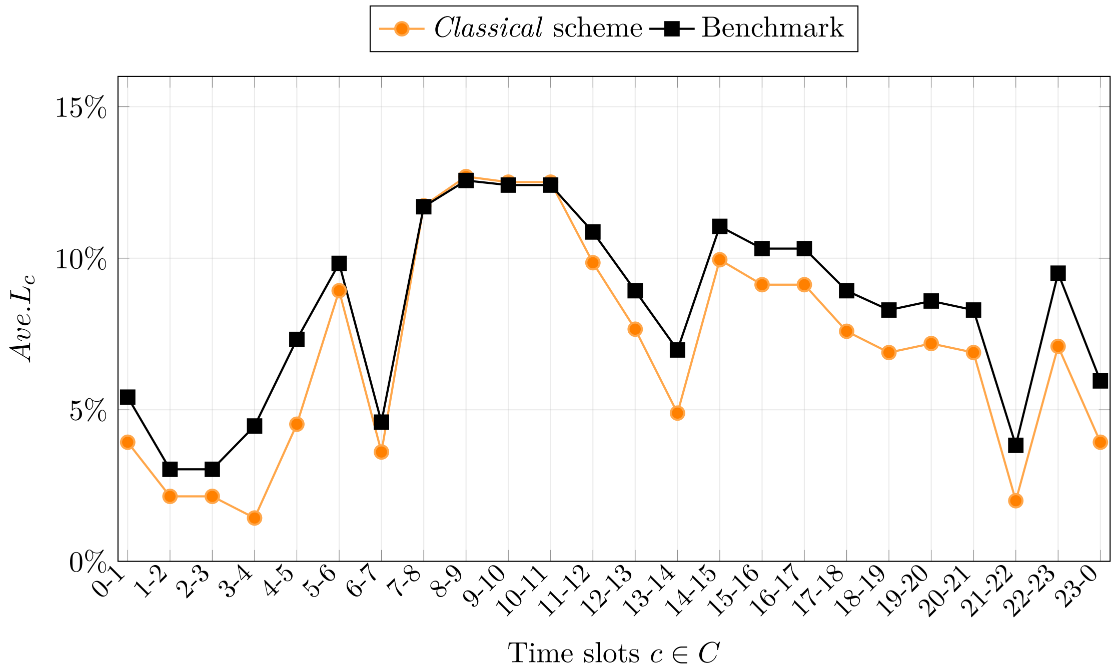


Figure 4.2: Average lack of coverage demand for time slot $Ave.L_c$ (%).

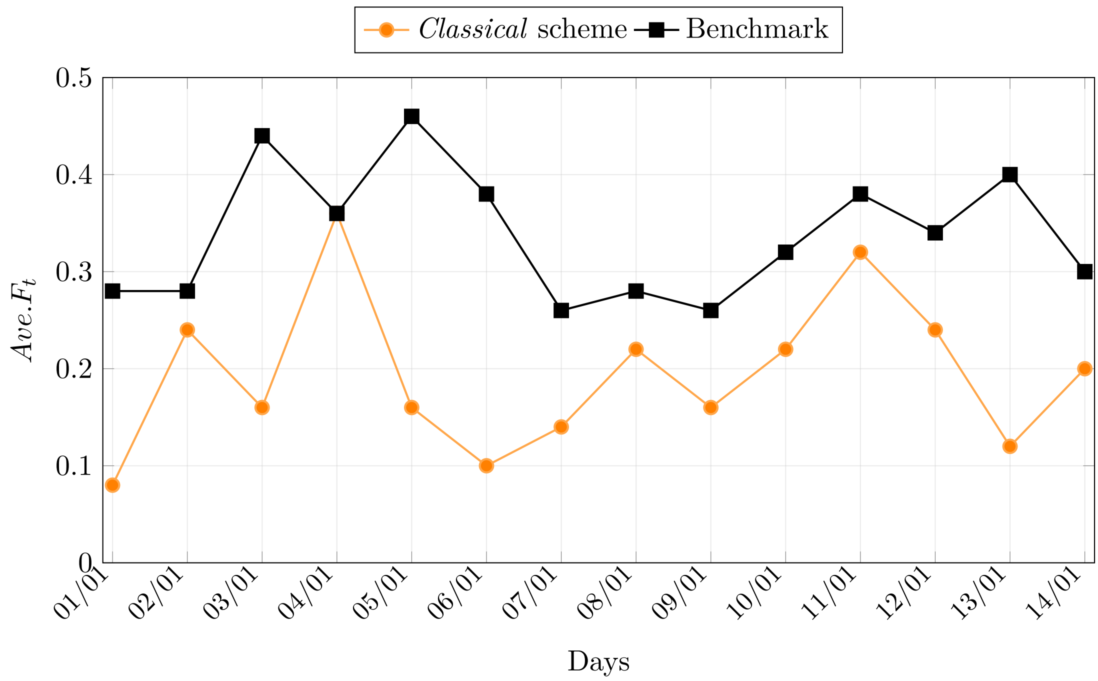


Figure 4.3: Average daily unfairness $Ave.F_t$.

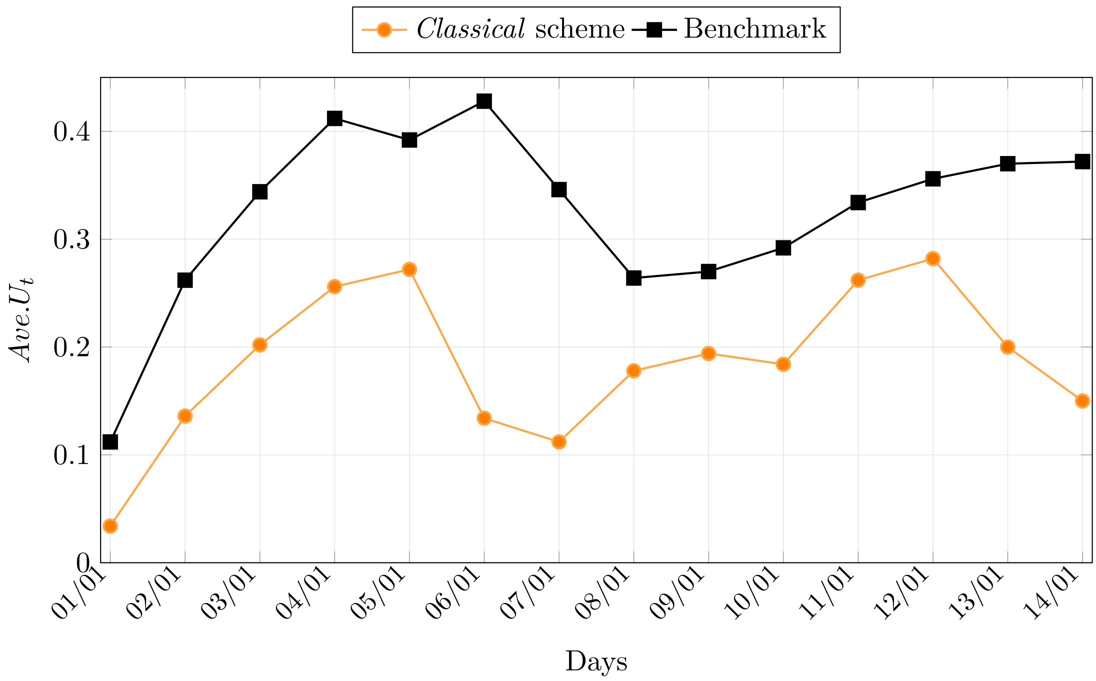
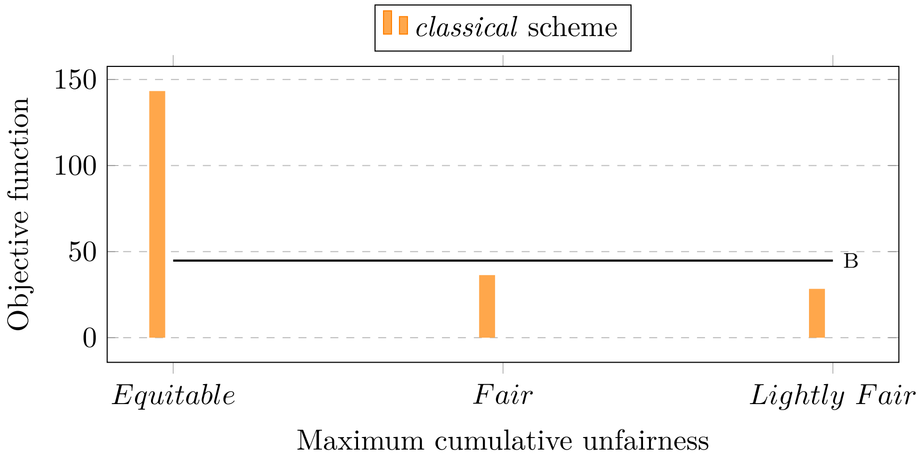


Figure 4.4: Average cumulative unfairness $Ave.U_t$.

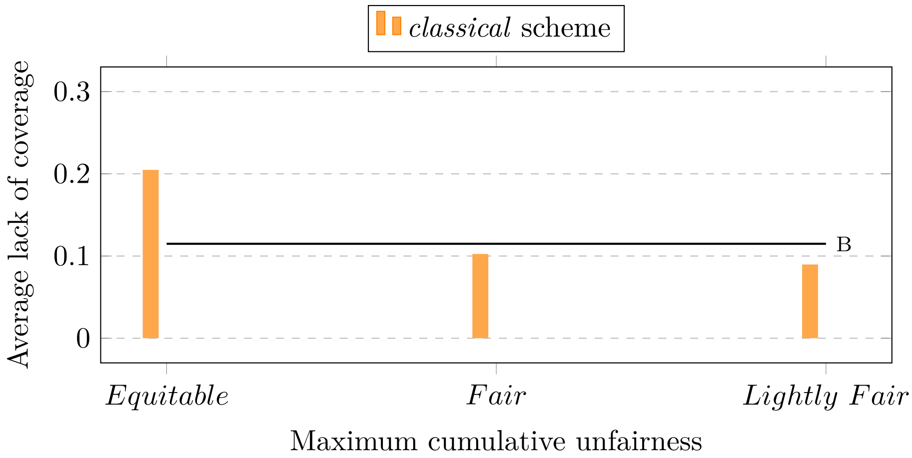
α, it was expected. For this reason, the *Fair* scenario appears to be the most convenient one, since it provides results similar to those of *LightlyFair* in terms of objective function value and lack of coverage, while achieving lower cumulative unfairness. Nevertheless, both the *Fair* and *LightlyFair* scenarios improve with respect to the benchmark. By contrast, the *Equitable* scenario is not convenient, as it performs significantly worse than the benchmark for all parameters. Although it is a simple system to apply, it is not optimized and does not properly account for fairness.

4.3 Results with *flexible* schemes

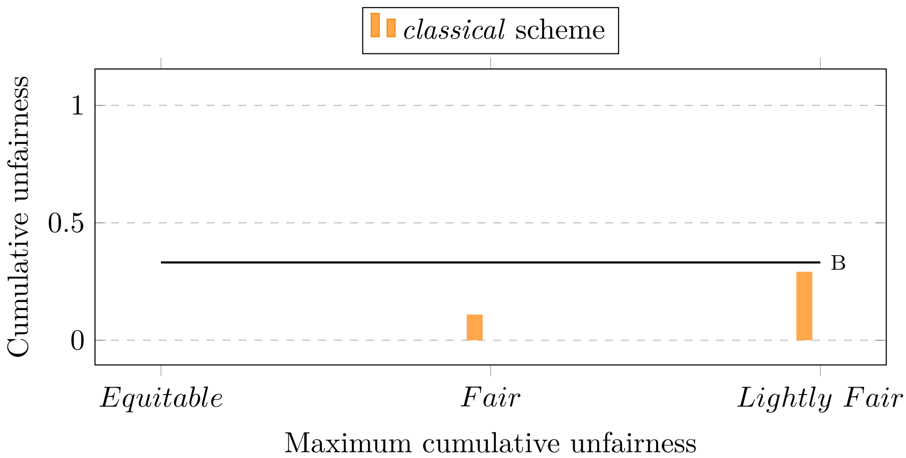
Note: mettere esempio turni ottenuti tabella+sistemare legende In this section, the results obtained with the introduction of additional shifts are presented. The benchmark is compared with the *classical*, *additional* and *flexible* schemes. The metrics used are the same as those already introduced in Section 4.2. The goal is to understand whether and how flexibility can lead to better results. In Section 4.2, it became clear that an optimized approach can lead to an improvement with respect to the benchmark. The objective now is to understand which level of flexibility makes the application of the algorithm more convenient. Of course, many different schemes could be tested, not only the three considered here. However, the aim is to assess whether flexibility is a key driver of improvement. As before, the results for the first two weeks of January 2024 are presented as an example. Then, the results obtained under *Equitable*, *Fair* and



(a) Objective function value Φ for increasing maximum cumulative unfairness values.



(b) Average lack of coverage $Ave.L_t\%$ with increasing maximum cumulative unfairness values.



(c) Average cumulative unfairness $Ave.U_t$ with increasing maximum cumulative unfairness values.

Figure 4.5: Performance indicators for increasing maximum cumulative unfairness values.

LightlyFair scenarios over the entire year are shown. This makes it possible to identify the best trade-off between the amount of unfairness allowed and the level of flexibility. The corresponding values of α are the same as before. The results for the first two weeks of January 2024, under the *Fair* scenario, are presented in Figures 4.6-4.10. In Figure 4.6 the objective function value Φ_t is presented. As already said, in most cases, the results obtained with the *FDS* model are better than those of the benchmark, as the value of the objective function is generally lower. Moreover, the objective function shows a more stable behavior, suggesting that the schedule produced by the *FDS* model is more balanced across the days. The objective function values obtained with the *classical*, *additional*, and *flexible* schemes follow the same general pattern, although the values obtained with the *additional* and *flexible* schemes are lower than those obtained with the *classical* one. However, there is no significant difference between the *additional* and *flexible* schemes, even though the *flexible* scheme performs slightly better.

The average lack of coverage, $Ave.L_c$, is shown in Figure 4.7. The values obtained with the *FDS* model are always lower than the benchmark. Among the three schemes, the *classical* scheme has the highest lack of coverage, while the *additional* and *flexible* schemes perform better. The *flexible* scheme appears to perform slightly better than the *additional* one, although the difference is small. As mentioned before, the coverage achieved by the benchmark is already good, therefore, large improvements are not expected. Moreover, the $Ave.L_c$ obtained with the *classical* model was already low, so it is reasonable that there are no major differences among the three schemes, since there is limited room for improvement. The main improvement is observed when a time slot is critical, possibly due to lower operator availability. In these cases, the *FDS* model follows the same pattern as the benchmark, meaning that the critical time slot remains the most uncovered, but the lack of coverage is significantly lower. For example, from 8 a.m. to 12 p.m., the benchmark has an average lack of coverage of approximately 45%, whereas the three schemes have values around 30% or lower. Naturally, the more *flexible* the scheme, the more variable the lack-of-coverage pattern may appear. Nevertheless, these fluctuations are small in percentage terms, and the values are always lower than the benchmark. This represents the main advantage of flexibility: it provides more possibilities to address critical time slots. For instance, the time slot from 5 p.m. to 6 p.m. has a lack of coverage of approximately 30% for the benchmark, the *classical* scheme, and the *additional* scheme, while the *flexible* scheme achieves 0%. All schemes except the *flexible* one are forced to maintain a flat lack of coverage of about 30% from 5 p.m. to 9 p.m., whereas the *flexible* scheme fluctuates between 0% and 30%. This is due to the rigidity of the other schemes. The *flexible* scheme can probably better adapt to the different operator shifts, allowing it to reduce lack of coverage in specific critical time-slots.

The average daily unfairness statistics are presented in Figure 4.3. Except for one instance, the daily unfairness of the benchmark is always higher. Surprisingly, the *classical* and *flexible* scheme follow the same pattern and there is no significant difference between them. The *additional* scheme is the more stable and performs better in general. This makes sense, because *classical* and *flexible* are two opposite schemes in terms of flexibility. The *classical* scheme is probably less fair and unstable because, on critical days, the model is forced to assign operators to shifts they don't like to ensure a proper coverage. Since there is no flexibility at all, there is no room to combine different kinds of shifts to cover critical time-slots. On the contrary, the *flexible* scheme probably combines many different types of shifts to achieve coverage, and this results in unfairness fluctuations. Therefore, a compromise between no flexibility and too much flexibility seems to produce more stable fairness across the days. A more stable fairness level is preferable because, otherwise, operators may be satisfied one day and heavily dissatisfied the next, leading to a biased perception of fairness. Of course, to confirm this hypothesis, it is essential to analyze different scenarios. However, since the *Fair* scenario is the average one, it should represent the behavior of the model and the relationship between coverage and fairness. The average cumulative unfairness $Ave.U_t$ is shown in Figure 4.9. The improvement in the cumulative unfairness is significant, there is a 30% or more gap between the benchmark and the results obtained by the *FDS* model. This is natural, because in the benchmark fairness is not considered at all. The cumulative unfairness of the three schemes is always lower than 20%. Therefore, all of the three schemes can lead to significant reduction of the unfairness. However, the *additional* scheme is more stable one. Therefore, both from a daily and a cumulative unfairness point of view is the preferable one. In Figure 4.9 the dashed line represents the α value for the *Fair* scenario. As explained before, $\alpha \in [0, 1]$ represents the maximum level of cumulative unfairness accepted. In this scenario, there is at worst a 20% gap between α and the actual cumulative unfairness, for all of the three schemes. The maximum cumulative unfairness is represented in Figure 4.10. For each day, it represents the maximum cumulative unfairness experienced between the operators, therefore it represents the worst case. Again there is a huge gap between the benchmark and the three schemes. The benchmark is always significantly higher than the maximum feasible value α, represented by the dashed line. There is not a significant difference between the three schemes, but they never cross the upper boundary α. Even in the worst instance, they produce an acceptable level of unfairness. In conclusion, from this analysis of the results obtained for the first two weeks of January 2024, it is evident that the most significant improvement achieved through the adoption of the *FDS* model concerns fairness, both daily and cumulative. Of course, there is also an improvement in terms of coverage; however, as explained before, since the benchmark is already good, this improvement cannot be large in percentage terms. Nevertheless, these preliminary results show that slightly better coverage can be achieved while ensuring a much higher level of fairness, which represents a considerable improvement in terms of operator satisfaction. It is important that the improvement concerns both daily and cumulative unfairness, because this ensures not only that operator satisfaction is higher throughout the year, but also that it remains stable on a daily basis, leading to a perceptible increase in satisfaction.

In Figure 4.11 the results obtained with different scenarios are shown. The horizontal black line marked as B represents the value of the corresponding statistic computed for the benchmark. In terms of the objective function value, the results are presented in Figure 4.11a. As expected, *Equitable* has, for each scheme, the highest value of the objective function. Still, it's interesting to notice that, while *classical* is far above the benchmark, *additional* and *flexible* are significantly lower and around the benchmark. Therefore, in the *Equitable* scenario there is a significant difference between *classical* and the other two schemes. Both *Fair* and *LightlyFair* are under the benchmark for all of the three schemes. While the objective function value remains similar between *Fair* and *LightlyFair*, *classical* has a more significant improvement from *Fair* to *LightlyFair*. The same exact behavior take place in terms of average lack of coverage, as shown in Figure 4.11b. In terms of cumulative unfairness the pattern is different, as shown in 4.11b. As expected, the average cumulative unfair is zero for the *Equitable* scenario. For all of the three schemes, the cumulative unfairness is three times higher than the *Fair* scenario, around 0.4. However, in both scenarios the performance is more or less equal for all of the three schemes. This means, that in average, the three schemes performs in the same way in terms of cumulative unfairness throughout all the year. Still, even if the average is the same throughout the year, the *additional* scheme is the most stable one, as explained before, so it's the most convenient one.

In conclusion, the *Fair* scenario provides a lower unfairness while mantaining a convenient level both in terms of objective function value and average lack of coverage. The significant improvement with respect to the benchmark is in terms of unfairness. Between the three schemes, the most convenient is the *additional* one, providing good results both in terms of coverage and unfairness and being stable throughout the year. To resume, a compromise between too flexibility and no flexibility at all is the best setting. Moreover, a medium rate for the maximum cumulative unfairness allowed α, is the most performing parameter choice.

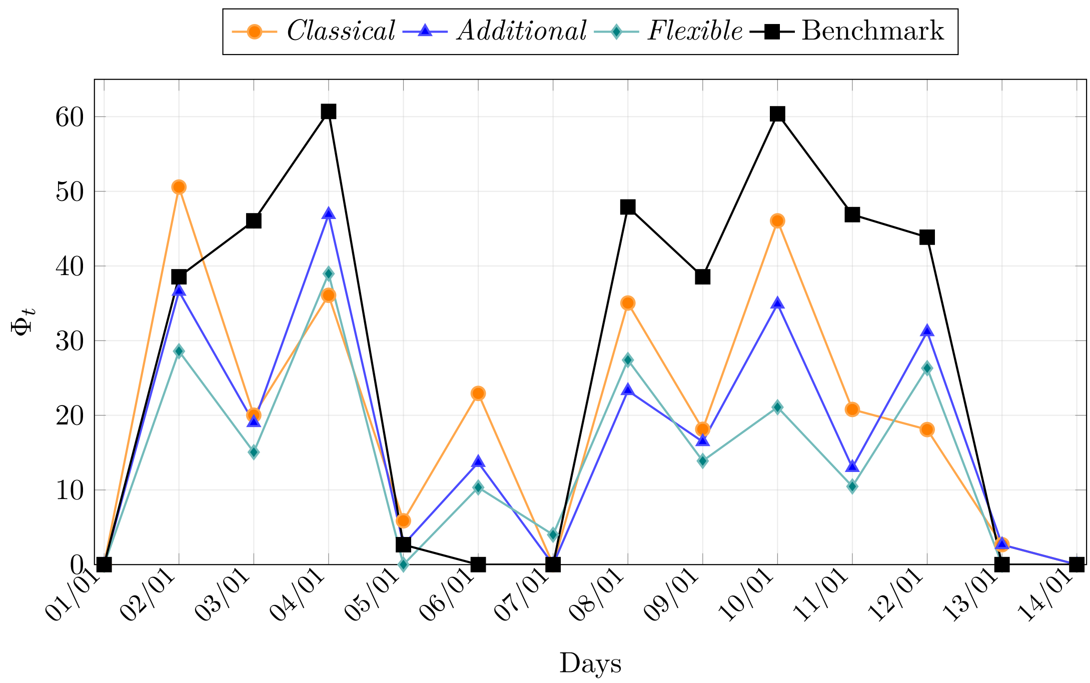


Figure 4.6: Objective function value Φ_t in the single optimization round and under *Fair* level of maximum cumulative unfairness.

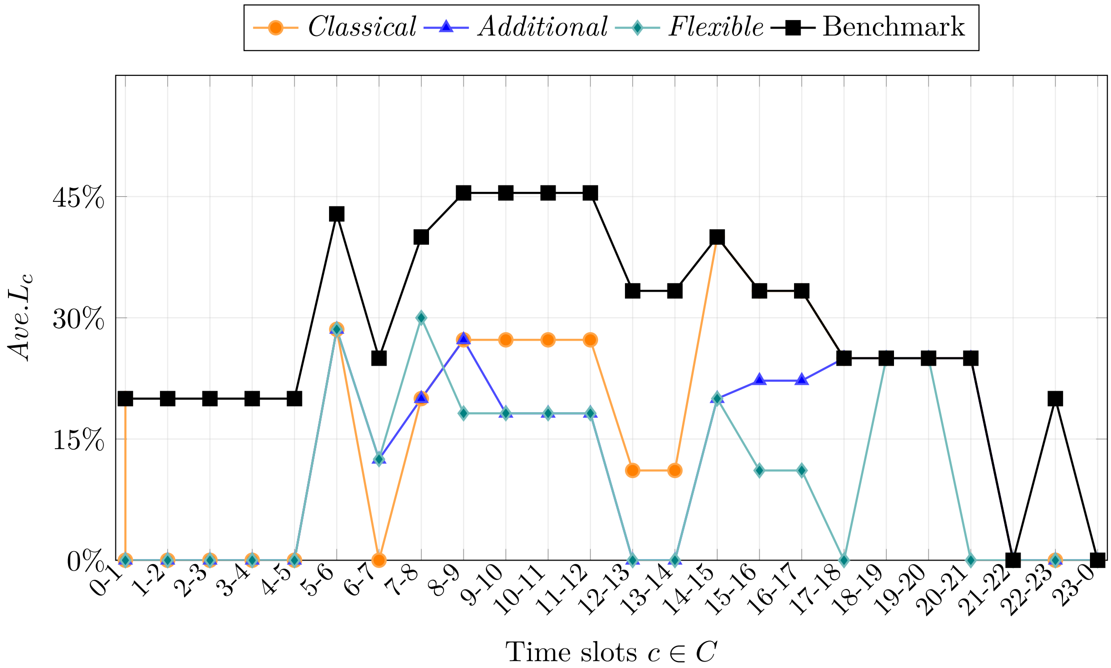


Figure 4.7: Average lack of coverage $Ave.L_c$ (%) for time slot on a single day and under *Fair* level of maximum cumulative unfairness.

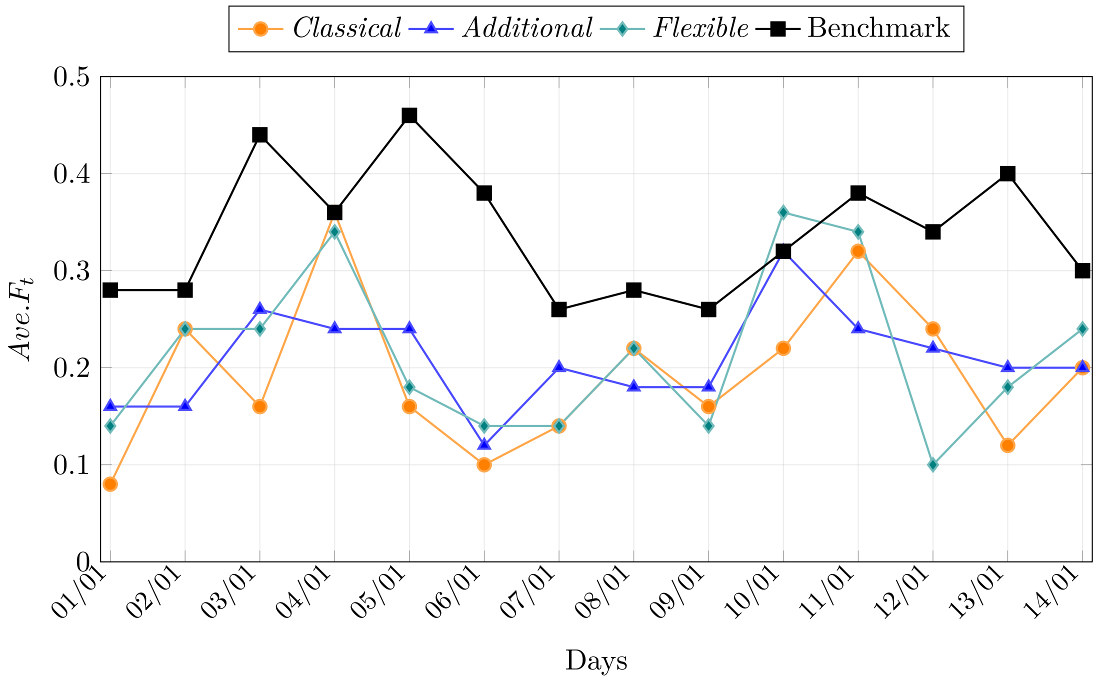


Figure 4.8: Average daily unfairness $Ave.F_t$.

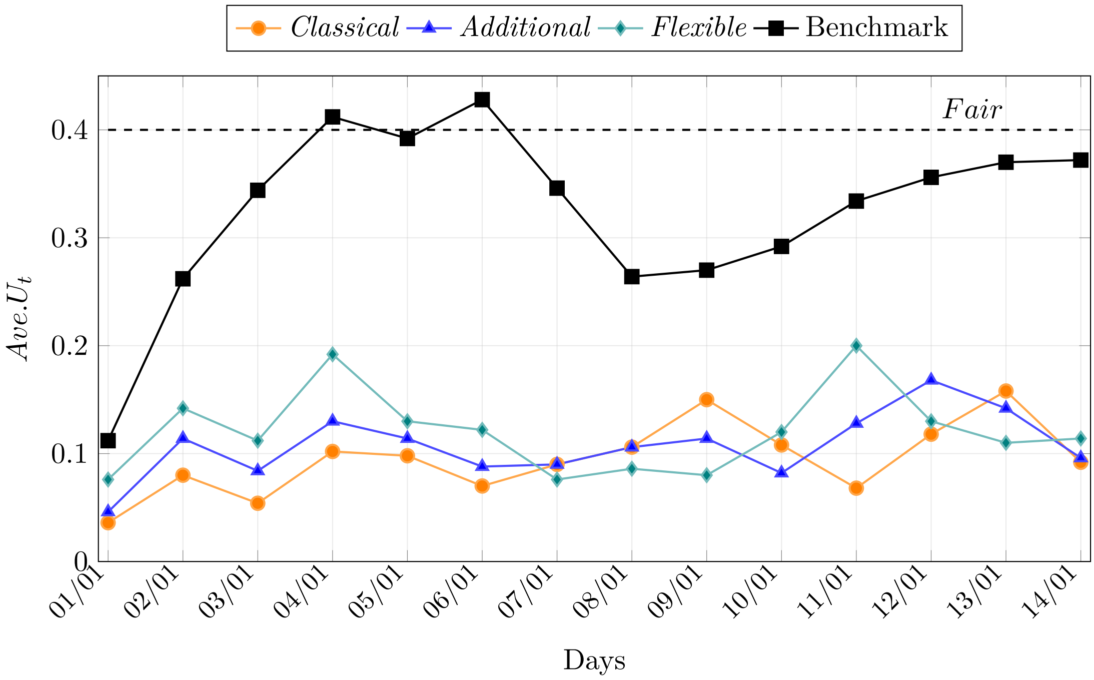


Figure 4.9: Average cumulative unfairness $Ave.U_t$ under *Fair* level of maximum cumulative unfairness.

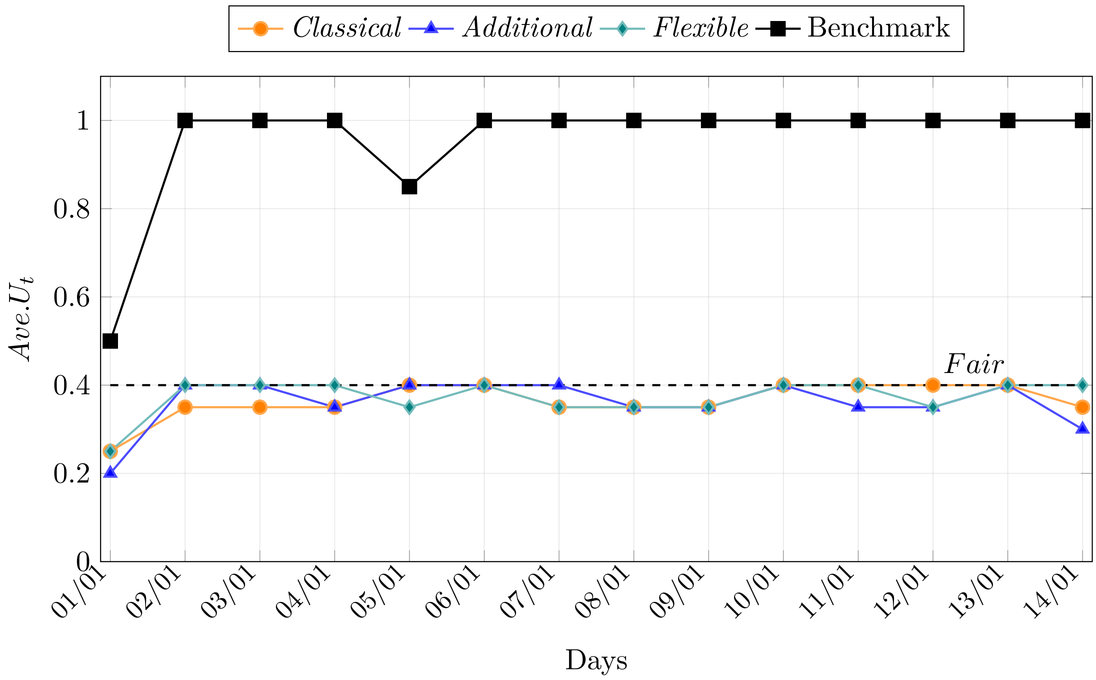
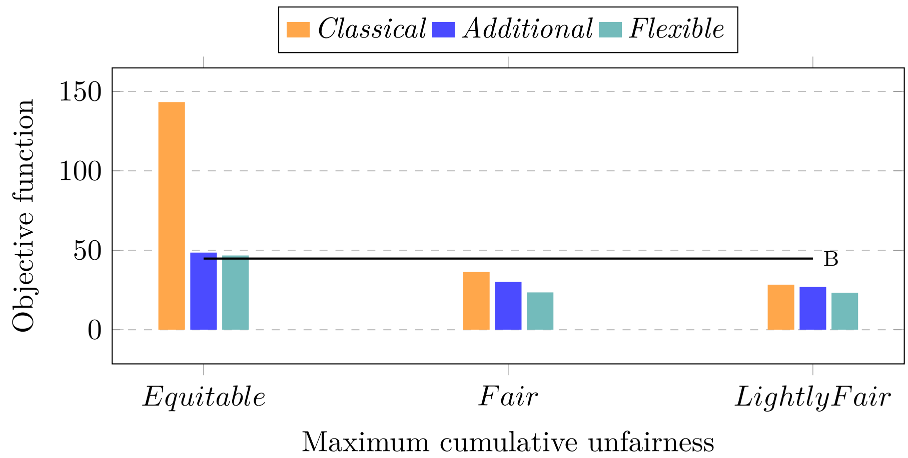
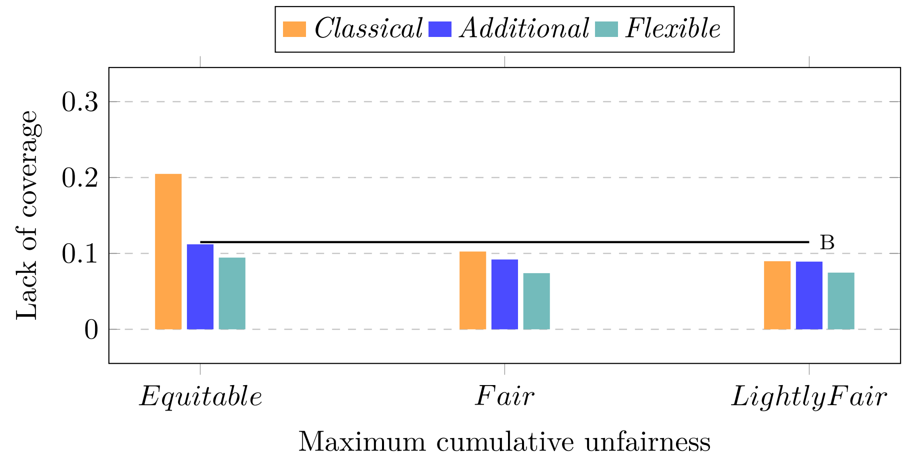


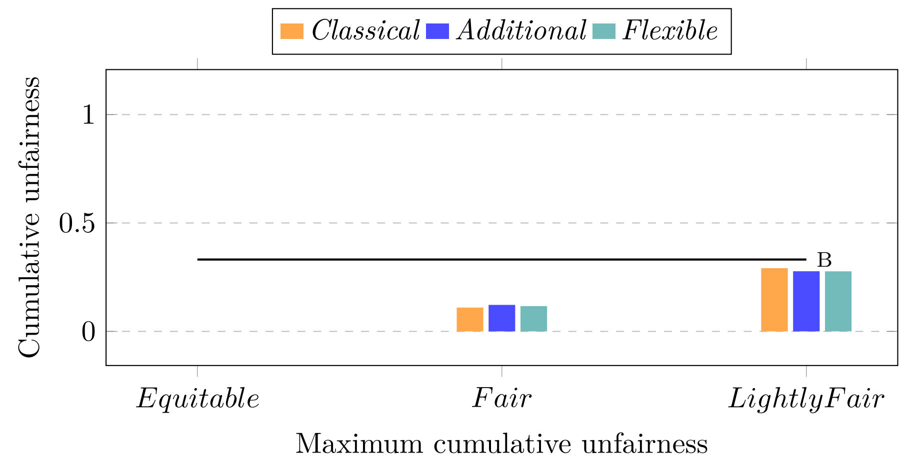
Figure 4.10: Maximum cumulative unfairness $Max.U_t$ under *Fair* level of maximum cumulative unfairness.



(a) Objective function value Φ for increasing maximum cumulative unfairness values.



(b) Average lack of coverage $Ave.L_t\%$ with increasing maximum cumulative unfairness values.



(c) Average cumulative unfairness $Ave.U_t$ with increasing maximum cumulative unfairness values.

Figure 4.11: Performance indicators for increasing maximum cumulative unfairness values under different schemes.

Chapter 5

Conclusions and future developments

The purpose of this thesis was to analyze and propose a solution to the scheduling problem for a type of work organized into shifts, comparable to that of a call-center. Work organized in shifts is common in emergency services, hospitals, security services and many other sectors. An adequate number of working operators is necessary to ensure the required level of service. On the other hand, work organized in shifts can heavily affect operators' quality of life. Therefore, considering *fairness* aspects is fundamental. Nowadays, fairness has become increasingly important, as both companies and workers pay more attention to this issue. To resume, the goal was to optimize workforce coverage while ensuring a proper level of fairness.

In this thesis, a model was developed to reach this goal: the *Fair Demand-Responsive Scheduling (FDS)* model. To ensure a proper coverage, a lack of coverage penalty is computed for each task and for each time slot of every day within the planning horizon. The objective function minimizes this penalty. There are different definition of *fairness*, each prioritizing different aspects. The approach proposed aims to follow the operators' preferences in terms of shifts. Therefore, a penalty based on those preferences is computed and referred to as *unfairness*. In particular, two aspects of unfairness are considered: *daily* unfairness and *cumulative* unfairness. The overall unfairness is computed as a convex combination between the two of them. Unfairness is controlled throughout constraints that ensure it does not exceed a predefined upper bound. See Section 2.3 for the mathematical formulation. Some additional constraints have been added to make the model usable in real-life context and compliant with legal requirements. See Section 2.3.1 for details. This thesis was developed in collaboration with Multiprotexion S.r.l., a leading Italian security company, and the model was tested on this real-world use case. Real operational data were collected. First of all, the company needs in terms of working operators must be determined. To this end, an hourly workload profile was computed, and the number of required operators was subsequently estimated and validated by the manager. Then, the preferences of 25 operators currently employed by Multiprotexion were collected. See Chapter 3 for details.

The monthly schedules produced by the *FDS* model for 2024 were discussed with the manager. In addition, a set of KPIs was defined to evaluate both coverage and fairness. The schedules that actually took place in 2024 were used as a benchmark. The model was tested under different shift schemes , characterized by varying levels of flexibility, and different maximum levels of unfairness. The results obtained are presented in Chapter 4. The first advantage of adopting an optimization-based approach to the scheduling problem is a smarter and more efficient management of schedule generation, resulting in a considerable reduction in the manager's workload. In terms of coverage, the model achieved an improvement, although the benchmark schedules already provided a satisfactory level of coverage. The best results were obtained with a medium level of flexibility and a medium level of maximum unfairness allowed. However, the main improvement and added value of the model is in terms of fairness. The model is able to ensure the same level of coverage, or even higher one in most cases, while achieving a significantly higher level of fairness. This means that operational performance can be improved while simultaneously increasing operator satisfaction. Once again, the better results were obtained with a medium level of flexibility and a medium level of maximum cumulative unfairness. These configurations also proved to be more stable in terms of fairness over the long term.

In conclusion, this thesis proposes an optimization-based approach to shift scheduling that incorporates fairness considerations. Moreover, the *FDS* model is usable in real-world contexts, as it integrates all relevant legal constraints, such as minimum rest periods between shifts, and can accommodate operator requests that have already been approved, such as holidays and days off. Furthermore, the model can be easily extended to include additional workforce characteristics, such as part-time employees and shift supervisors. Therefore, the proposed approach provides a practical and flexible tool for addressing workforce scheduling problems while balancing operational requirements and employee fairness.

From the company's point of view, the work has been widely appreciated. The model is currently undergoing a testing phase and is already being used by the Operations Center manager for shift scheduling activities. Furthermore, to ensure a smart and user-friendly experience, a dedicated web platform has been developed by the *Data Analytics and Artificial Intelligence* team, in particular by Dr. Angjelo Zojzi with the collaboration of Dr. Valentina Morandi (co-supervisor). This platform facilitates the practical adoption of the *FDS* model and supports its integration into the company's operational processes. The dashboard is shown in Figure 5.1. Moreover, several additional functionalities have been implemented. First of all, the user is provided with a drag-and-drop interface. In practice, shift schedules sometimes require only minor modifications. For example, when one operator requests a shift change, it is not convenient to reschedule the entire month, as this could potentially alter the schedules of all operators. Thanks to this functionality, the manager can modify only the shift of the operator concerned and perform quick simulations while obtaining an immediate visualization of the resulting schedule. Furthermore, several workforce categories have been incorporated into the model, including part-time employees, shift supervisors, and night-only operators. These additions make the model fully capable of handling company-specific requirements and exceptional cases. Another interesting development was the introduction of some *jolly* operators, such as managers, who can be assigned to a shift only when no other feasible solution is available. This feature is particularly useful in emergency situations or during critical periods, such as seasonal flu outbreaks, when a significant number of employees may be unavailable. On-call operators were also included in the model. In addition, a heuristic procedure was developed. However, finding the optimal solution requires only about 30 seconds. Considering that schedules are generated on a monthly basis, computational time does not represent a significant issue. For this reason, exact optimization remains a practical and effective approach. Overall, the results obtained and the positive feedback received from the company suggest that the system will likely be officially adopted in the coming months.

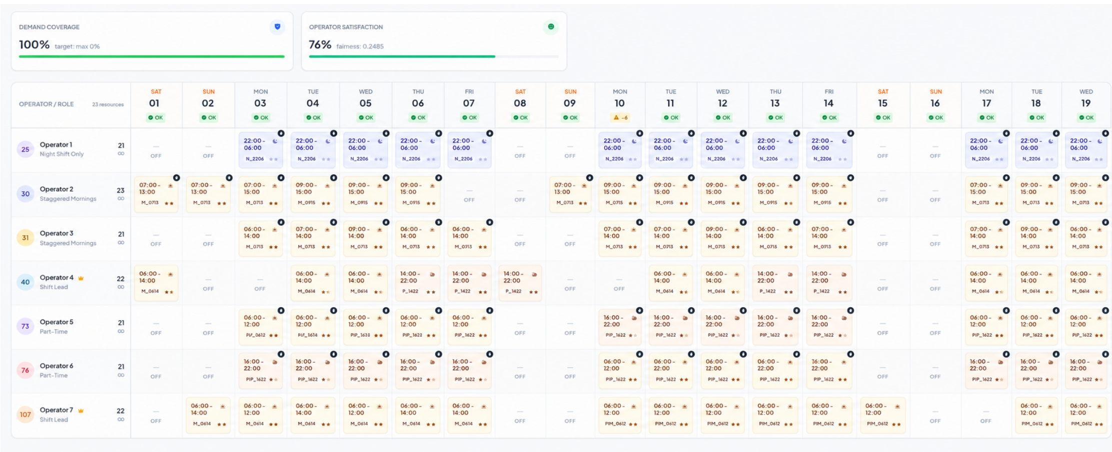


Figure 5.1: Operator scheduling dashboard

5.1 Future developments

The main improvement achieved in this thesis is in terms of fairness. Regarding coverage, companies are generally able to ensure a satisfactory level of coverage in order to deliver the services. However, producing a high-quality schedule still requires significant managerial experience and a considerable amount of time. Indeed, many variables must be taken into account, including the maximum number of consecutive working days, minimum rest periods, employee requests, days off, specific contractual conditions, and many others. In fact, workforce scheduling is an NP-hard problem. Nevertheless, companies are usually able to produce a good shift schedule in most cases. An even more important and challenging issue is *rescheduling* which becomes necessary when disruptions occur, such as employee sick leave. The main challenge of rescheduling in shift-based work environments is related to the minimum number of resting hours requirements between consecutive shifts. As mentioned before, in Italy is at least 11 consecutive hours. As an example, consider the *classical* shift scheme. If an operator assigned to the *Morning* shift is absent, they can only be replaced by operators who either worked the *Morning* shift or were off duty on the previous day. Similarly, if an operator assigned to the *Afternoon* shift is absent, he can be replaced by operators who worked either the *Morning* or the *Afternoon* shift, or who were off duty on the previous day. Finally, if an operator assigned to the *Night* shift is absent, they can be replaced by operators who worked any shift on the previous day or who were off duty. The compatibility relationships are summarized in Table 5.1. The rows represent the shift that must be covered, while the columns represent the shift worked by the replacement operator on the previous day. For example, when a *Morning* shift must be covered, the manager can only choose a replacement from a limited pool of operators. The selection must also consider additional constraints, such as days off, contractual limitations, and individual scheduling requirements. There is another important aspect to consider. Suppose that a *Night* shift has an absent operator and a replacement operator is reassigned to cover that shift. Because of minimum rest-time requirements and the structure of the shift rotation, the replacement operator may be required to continue following the *Night* shift cycle for the remaining days of the work rotation. Consequently, a single substitution may affect not only the current day but also the schedules of subsequent days, potentially generating the need for further adjustments and additional substitutions. If the reassignment occurs near the end of the rotation cycle, the impact may be limited. However, if it occurs near the beginning, the consequences can be significant. Adopting a more flexible shift scheme can help mitigate these issues. For example, the *Daily* shift included in the *additional* scheme can often be used to cover unexpected absences without creating conflicts with minimum rest-time requirements. Similar compatibility analyses should also be provided for the other shift schemes considered in this work and are presented in Table 5.1. Despite the availability of these solutions, rescheduling remains a critical issue for companies. Moreover, shift changes can have a significant impact on operators' quality of life, as well as on their sleep–wake cycles. Excessively frequent schedule modifications are therefore a major source of employee dissatisfaction. Furthermore, accepting a schedule change is typically not mandatory, and operators may refuse a proposed reassignment. For all these reasons, rescheduling represents a particularly delicate and complex problem that must be handled carefully in real-world workforce management contexts.

A possible future development, which has been already discussed with the company, is the development of a rescheduling model. The goal of this model would be to manage unexpected disruptions while restoring the required level of coverage with the as low as possible impact on the existing schedule. The key idea is to fix the schedule by modifying only the shifts that are strictly necessary. To achieve this goal, a penalty could be associated with each schedule modification. The objective function would then minimize the total number of changes, thus preserving the original schedule as much as possible. However, minimizing schedule changes alone is not sufficient. Fairness should also be taken into account. In particular, the burden of covering unexpected absences should not systematically fall on the same group of operators. Therefore, the fairness measure should be extended to include the distribution of replacement assignments among employees, ensuring that additional workload is shared equitably over time. Another important aspect concerns operator preferences. When a shift is modified, the new assignment may differ from the employee's preferred schedule. Consequently, the rescheduling model should also minimize the degradation of preferences, preserving, whenever possible, the operators' original requests and desired shift patterns. It is worth noting that such a model would be activated only when necessary. In many situations, a single absence does not significantly compromise service coverage, and no corrective action is required. In other cases, minor adjustments can be handled directly by the manager without the support of an optimization model. Therefore, the rescheduling procedure would be particularly valuable in situations involving multiple absences, critical periods, or disruptions that cannot be efficiently managed through manual interventions. In conclusion, a rescheduling model would represent a natural extension of the *FDS* model. While the current model focuses on generating fair and efficient monthly schedules, the rescheduling model would provide decision support for handling unforeseen events, balancing coverage requirements, fairness, schedule stability, and employee satisfaction.

	Morning	Afternoon	Night	Rest
Morning	✓	✗	✗	✓
Afternoon	✓	✓	✗	✓
Night	✓	✓	✓	✓

(a) *Classical* scheme

	Morning	Late Morning	Afternoon	Late Afternoon	Daily	Night	Rest
Morning	✓	✓	✗	✗	✓	✗	✓
Late Morning	✓	✓	✗	✗	✓	✗	✓
Afternoon	✓	✓	✓	✓	✓	✗	✓
Late Afternoon	✓	✓	✓	✓	✓	✗	✓
Daily	✓	✓	✓	✗	✓	✗	✓
Night	✓	✓	✓	✓	✓	✓	✓

(b) *Additional* scheme

	Morning	Late Morning	Early Afternoon	Afternoon	Late Afternoon	Splitted Daily	Daily	Early Night	Night	Rest
Morning	✓	✓	✗	✗	✗	✓	✓	✗	✗	✓
Late Morning	✓	✓	✓	✗	✗	✓	✓	✗	✗	✓
Early Afternoon	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓
Afternoon	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓
Late Afternoon	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓
Splitted Daily	✓	✓	✓	✗	✗	✓	✓	✗	✗	✓
Daily	✓	✓	✓	✓	✗	✓	✓	✗	✗	✓
Early Night	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Night	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

(c) *Flexible* scheme

Table 5.1: Compatibility matrices between the shift to be covered and the shift worked on the previous day by the replacing operator

Bibliography

Z. Aksin, M. Armony, V. Mehrotra, The modern call center: a multi-disciplinary perspective on Operations Managment Research, *Production and Operations Management*, **16**, 2007

J. Van den Bergh, J. Beliën, P. De Bruecker, E. Demeulemeester, L. De Boeck, *European Journal of Operational Research*, 2012

A. Uhde, N. Schlicker, D. Wallach, M. Hassenzahl, Fairness and Decision-making in collaborative shift scheduling systems,*Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020

L. Wolbeck, Fairness aspects in personnel scheduling, 2019

O. Şimşek, O. Kundakcioglu, Cost of fairness in agent scheduling for contact centers, *Journal of Industrial & Management Optimization*, 2022

.1 Examples of shift scheduling produced by the *FDS* model

In this appendix are shown the schedules obtained for the first two weeks of January 2024 with the three schemes. All of them was produced under the *Fair* assumption.

	MORNING	AFTERNOON	NIGHT	REST
01/01/2024	#6 #15 #19 #23 #25	#4 #17 #20 #21 #22	#1 #7 #14 #16 #24	#2 #3 #8 #9 #10 #11 #12 #13 #18
01/02/2024	#2 #6 #15 #19 #23 #25	#8 #9 #17 #20 #21	#1 #4 #16 #24	#3 #6 #7 #10 #11 #12 #13 #14 #18 #22
01/03/2024	#2 #3 #10 #11 #13 #18 #23 #25	#5 #6 #8 #12 #21 #25	#1 #9 #16 #17 #24	#4 #7 #14 #15 #19 #20 #22
01/04/2024	#2 #7 #13 #14 #18 #22	#5 #8 #10 #11 #12 #23	#3 #6 #9 #21 #25	#1 #4 #15 #16 #17 #19 #20 #24
01/05/2024	#13 #15 #19 #20 #22	#4 #7 #10 #11 #12	#2 #3 #8 #18 #25	#1 #5 #6 #9 #14 #16 #17 #21 #23 #24
01/06/2024	#15 #19 #20 #22 #24	#4 #7 #11 #12 #17	#1 #2 #10 #16 #18	#3 #5 #6 #8 #9 #13 #14 #21 #23 #25
01/07/2024	#6 #20 #21 #23 #24	#4 #7 #15 #17 #22	#1 #5 #11 #16	#2 #3 #8 #9 #10 #12 #13 #14 #18 #19 #25
01/08/2024	#8 #9 #13 #21 #23 #24	#3 #6 #14 #20 #22 #25	#5 #7 #15 #16 #17	#1 #2 #4 #10 #11 #12 #18 #19
01/09/2024	#2 #10 #12 #13 #18 #19 #21	#3 #6 #8 #9 #14 #20 #25	#15 #16 #17 #22 #23	#1 #4 #5 #7 #11 #24
01/10/2024	#2 #4 #10 #11 #13 #18 #19	#3 #6 #8 #9 #12 #20 #21	#14 #16 #17 #23 #25	#1 #5 #7 #15 #22 #24
01/11/2024	#1 #2 #5 #7 #13 #18	#4 #8 #9 #10 #12	#3 #6 #11 #14 #16	#15 #17 #19 #20 #21 #22 #23 #24 #25
01/12/2024	#17 #21 #22 #23 #24	#1 #2 #4 #5 #8 #9 #25	#3 #6 #10 #11 #12 #13 #14	#7 #15 #16 #18 #19 #20
01/13/2024	#17 #18 #19 #20 #21 #22	#1 #4 #5 #23 #24 #25	#2 #3 #8 #9 #10 #11 #12	#6 #7 #13 #14 #15 #16
01/14/2024	#7 #8 #15 #16 #23 #24	#9 #10 #17 #18 #25	#1 #2 #4 #5 #11 #12 #19 #20	#3 #6 #13 #14 #21 #22

Table 1: Example of shift scheduling with the *classical* scheme under the *Fair* assumption

	MORNING	LATE MORNING	AFTERNOON	LATE AFTERNOON	DAILY	NIGHT	REST
01/01/2024	#4 #19 #21 #23 #24		#1 #5 #20 #22 #25			#6 #7 #14 #15 #16 #17	#2 #3 #8 #9 #10 #11 #12 #13 #18
01/02/2024	#2 #19 #21 #23	#8	#1 #4 #5 #22 #25	#17	#24	#7 #9 #14 #15 #16 #20	#3 #6 #10 #11 #12 #13 #18
01/03/2024	#2 #10 #11 #12 #18 #19 #23	#8 #24	#1 #3 #4 #5 #21	#13 #25		#16 #17 #20 #22	#6 #7 #9 #14 #15
01/04/2024	#2 #6 #14 #15 #18 #24	#8 #9 #10	#11 #13 #21 #23 #25	#4 #5	#3	#1 #12 #16 #17 #22	#7 #19 #20
01/05/2024	#3 #8 #14 #15 #24	#6 #10	#4 #13 #25	#2 #18		#1 #5 #9 #12 #23	#7 #11 #16 #17 #19 #20 #21 #22
01/06/2024	#6 #8 #19 #20	#7	#10 #17 #25	#3		#1 #2 #4 #12 #13	#5 #9 #11 #14 #15 #16 #18 #21 #22 #23 #24
01/07/2024	#20 #21 #22	#16	#3 #7 #10 #17	#6		#2 #11 #12 #13 #19	#1 #4 #5 #8 #9 #14 #15 #18 #23 #24 #25
01/08/2024	#14 #16 #21 #22 #23 #24	#18	#5 #6 #9 #20	#11	#7 #17	#3 #12 #13 #15 #19	#1 #2 #4 #8 #10 #25
01/09/2024	#1 #18 #21 #23 #25	#7 #17 #22	#4 #5 #8 #9 #10 #14 #16	#20 #24		#6 #11 #15 #19	#2 #3 #12 #13
01/10/2024	#2 #17 #18 #21 #23	#7 #22	#1 #4 #8 #10 #16 #20	#5 #14 #15	#25	#9 #11 #19 #24	#3 #6 #12 #13
01/11/2024	#2 #7 #13 #14 #15 #18 #22 #25	#3 #12	#5 #16 #17 #20 #21 #23		#8	#1 #4 #9 #10 #11 #19 #24	#6
01/12/2024	#2 #3 #6 #12 #25	#8 #13 #18	#14 #15 #16 #17 #21 #22 #23	#4		#1 #5 #9 #10 #11 #24	#7 #19 #20
01/13/2024	#2 #3 #8 #13 #14 #18	#6	#4 #17 #21 #22 #25			#1 #5 #9 #10 #12 #23 #24	#7 #11 #15 #16 #19 #20
01/14/2024	#6 #15 #16 #19 #20		#3 #7 #8 #25			#1 #2 #4	#5 #9 #10 #11 #12 #13 #14 #17 #18 #21 #22 #23 #24

Table 2: Example of shift scheduling with the *additional* scheme under the *Fair* assumption

	MORNING	LATE MORNING	EARLY AFTERNOON	AFTERNOON	LATE AFTERNOON	SPLITTED DAILY	DAILY	EARLY NIGHT	NIGHT	REST
01/01/2024	#1 #18 #19 #23 #24		#5 #15 #22	#20 #21				#7 #17	#4 #16 #25	#2 #3 #6 #8 #9 #10 #11 #12 #13 #14
01/02/2024	#2 #9 #18 #19 #23	#5	#1 #24	#8 #20	#16	#15 #22		#17	4 #7 #21 #25	#3 #6 #10 #11 #12 #13 #14
01/03/2024	#2 #3 #10 #11 #13 #23	#1	#8 #14 #21	#5 #6 #12	#9	#22 #24		#19 #25	#4 #16 #17 #20	#7 #15 #18
01/04/2024	#1 #2 #13 #22 #24	#23	#6 #9	#5 #12	#17	#8 #10 #14		#4	#3 #11 #21 #25	#7 #15 #16 #18 #19 #20
01/05/2024	#8 #13 #15 #23 #24	#14	#10	#7 #12	#6	#2		#3 #4 #5 #25	#1 #9	#11 #16 #17 #18 #19 #20 #21 #22
01/06/2024	#2 #8 #14 #15 #20		#12 #13	#6 #16		#10		#3 #25	#5 #7 #18	#1 #4 #9 #11 #17 #19 #21 #22 #23 #24
01/07/2024	#10 #15 #21 #22		#6	#11 #17	#18	#20		#8 #14 #16	#7 #12	#1 #2 #3 #4 #5 #9 #13 #19 #23 #24 #25
01/08/2024	#1 #19 #21 #22 #23 #24		#10 #17 #18	#6 #11	#4	#9 #15		#14	#7 #12 #16 #20	#2 #3 #5 #8 #13 #25
01/09/2024	#2 #5 #13 #19 #23 #25	#1 #18	#24	#3 #9 #21	#16 #20	#17		#11 #22	#4 #7 #15	#6 #8 #10 #12 #14
01/10/2024	#1 #2 #13 #17 #18 #19 #23		#5 #8	#3 #9 #20 #21		#24 #25		#15	#4 #7 #11 #16 #22	#6 #10 #12 #14
01/11/2024	#2 #6 #13 #14 #18 #24 #25	#23	#1 #9	#5 #10 #12 #19	#4	#8 #17		#20 #22	#3 #11 #16 #21	#7 #15
01/12/2024	#2 #8 #17 #23 #24 #25	#1	#5 #10	#4 #9 #12 #19		#13		#21	#3 #6 #11 #14 #22	#7 #15 #16 #18 #20
01/13/2024	#2 #8 #13 #15 #23 #24		#9	#4 #7 #10		#25		#3 #5 #12 #14	#1 #6	#11 #16 #17 #18 #19 #20 #21 #22
01/14/2024	#15 #16 #18 #20 #25		#2 #13	#7 #8 #10				#3 #5 #6	#12 #14	#1 #4 #9 #11 #17 #19 #21 #22 #23 #24

Table 3: Example of shift scheduling with the *flexible* scheme under the *Fair* assumption