CP-SAT Optimization for Hospital Room Assignment

Andrieta Irene Georghiadou

Master Dissertation
Data Science MSc
School of Computing
Newcastle University
Academic year 2024/2025

Contents

CP-SAT Optimization for Hospital Room Assignment

ANDRIETA IRENE GEORGHIADOU, School of Computing, Newcastle University,
MATTHEW FORSHAW, School of Computing, Newcastle University,

This dissertation project situates itself in the field of constraint programming optimisation, addressing the Patient Admission Scheduling (PAS) problem. A multi-phase framework was employed for the development of optimisation models using Google's CP-SAT solver, combined with a rolling horizon approach. The models incorporated both hard and soft logical constraints, achieving 100% feasibility and efficient run-times. The models were extended with stochastic variations to capture uncertain patient flows, and fairness mechanisms (historical bias and First Come, First-Served) resulted in more equitable allocations, recording improved fairness metrics. In the final phase, a Response Surface Methodology(RSM)-inspired approach was adopted for weight tuning of soft constraints, achieving reduced penalty variance, but revealing a fairness trade-off.

1 Introduction

Healthcare systems worldwide, but especially in the United Kingdom, are currently facing the consequences of rising patient numbers, while resources remain constraind [15], leaving NHS with one of the lowest bed-to-population ratios over the past decade. Routinely, occupancy rates in hospitals are well above the recommended safety threshold of 85% [20], frequently reaching 93-95%, the highest levels ever recorded [15]. Consequently, operational issues such as insufficient allocations, long wait times and congestion in the ER occur, compromising patient care delivery. Studies have linked persistent overcrowding to increased infection and mortality rates [17], as well as higher stress levels and burnout from frontline staff [9].

Healthcare is a priority concern due to its major effects on society's well-being and aim for universal health coverage [13]. This persistent and growing infrastructure management crisis requires more effective hospital resource planning strategies to enhance operational efficiency and avoid the aforementioned bottlenecks. Improvements in patient-to-room assignment support patient-centered care, operational efficiency in treatment delivery, and enhanced satisfaction from both professionals and patients.

Many papers applied metaheuristics to address problems like nurse rostering [4, 14] and overflow management [7, 19]. The patient admission scheduling (PAS) problem has been formalized and further studied by Demeester et al. [8] and was defined as assigning patients to hospital beds in a way that maximizes medical treatment effectiveness, efficiency management, and patient comfort [8]. Later, Ceschia and Schaerf [6] reformulated the problem in terms of room, instead of bed assignments, an approach that was also adopted in this project. Hence, the PAS problem in the current work is defined as efficiently assigning patients to hospital rooms over a planning horizon, in order to find the optimal solution that maximises patient needs, preferences, and resource utilization, while minimising constraint penalties.

The assignment of patients to hospital rooms can be a complex task, often characterized as "a game of Tetris", as continuous adaptations need to be made within a tight timeline [2]. The problem involves numerous shifting parameters that must be considered, such as varying room capacities, equipment availability and operational policies [21], which are imposed as additional constraints. An additional challenge can be the dynamic nature of patient flow, as patient arrivals and discharges can differ from predefined dates, introducing uncertainty [21]. This problem is proven to be NP-hard [21] as the solution space grows combinatorially and has been a long-standing challenge among researchers. Several methods and modeling strategies were studied in the previous years and while basic integer programming approaches [6, 8] failed to capture the full complexity, metaheuristics [6–9, 19] and constraint programming (CP) methods [4, 12, 18] have already been proven effective.

One of the most powerful tools for solving complex combinatorial optimization problems is CP-SAT, an advanced solver developed by Google [1]. This tool is designed to solve large problem instances efficiently, model a wide range of complex constraints (logical, temporal, capacity-related, etc.) and overcome limitations of other CP and Mixed Integer Linear Programming solvers. It can be characterized by robustness, because of its capacity to find high-quality solutions (often near-optimal or optimal) at impressive speed. CP-SAT was chosen in this project because its performance is promising and can often outperform the methods used by previous researchers [1]. Using this solver in hospital scheduling systems, shorter wait times, more efficient resource utilization and improved continuity of care can be achieved.

The primary objective of this project is the development of an optimisation model with the use of CP-SAT solver to address the PAS problem. In order to achieve this, other smaller objectives need to be met: The model aims for feasibility across the planning horizon and the identification of optimal, or near-optimal, solutions for the problem. Computational efficiency is an additional objective, seeking improved timeline in infrastructure management. Moreover, the model is oriented towards capturing the unpredictable nature of hospital stay length by incorporating an adaptable stochastic framework. Finally, the approach adopted, focused on enhanced equity in the hospital scheduling challenge with the use of fairness mechanisms, such as historical imbalance correction, First-Come, First-Served logic and and the application of RSM for tuning the soft constraints. Overall, the project was designed to explore the development of a decision-supporting tool that assists healthcare planning.

For the development of this project, five benchmark instances were used as data sources. These datasets consist of synthetic data published by Demeester et al. [8] for the needs of studying the PAS problem. The datasets were obtained, after formal request, by the authors Ceschia and Schaerf, who also used these instances for the development of their own method [6]. Realistic hospital environments are being simulated through these instances, including detailed attributes like patient demographics and preferences and room characteristics. Each scenario varies in the number of patients, rooms, and days of the planning horizons, representing both homogeneous and heterogeneous hospital settings.

Building upon the foundational work, this project aims to address the Patient Admission Scheduling problem with the use of a Constraint Programming approach, the CP-SAT solver. This dissertation report is organised as follows. In Section 2, Related Work, relevant literature is reviewed, after which Section 3, Methodology, where the main steps undertaken for the development of this project are described such as data handling, modeling phases and weight tuning. Section 4, Results, includes the evaluation and interpretation of the approaches used while, Section 5, concludes by discussing the strengths, limitations and future work for the project.

2 Related work

Due to its importance to society's well-being, the healthcare domain is popular among researchers who over the years have studied and tried to refine medical delivery-related problems, in order to offer their solutions to these long-standing challenges. Several studies have explored and applied optimization techniques to complex and real-world problems, trying to balance efficiency, uncertainty, and fairness. In this section, a literature review of the papers studied and used for the development of the current project is provided.

The patient admission scheduling (PAS) problem has been formalized and further studied by Demeester et al. [8] introducing a hybridized tabu search algorithm enhanced by token-ring structures. This approach intelligently assists the admission scheduler, while satisfying both hard and soft constraints. Because of its memory-based techniques decisions are taken fast, outperforming the integer programming approach they also tested. Local Search was also explored by Ceschia and Schaerf [6], this time with the use of the Simulated Annealing metaheuristic and dynamic lower bounds for improved solutions. By tailoring neighborhood combinations, flexible and satisfactory solutions were achieved quickly. The dynamic version of the problem was introduced in the aforementioned authors' follow up work [7], including several real-world features, such as the presence of emergency patients, uncertainty in the length of stay, and the possibility of delayed admissions. The Simulated Annealing metaheuristic was incorporated once again achieving minimal performance gap between the static and the dynamic case.

Following Ceschia and Schaerf's footsteps that managed to make some inroads into modelling uncertainty, many other researchers studied and developed different strategies trying to approximate real-life hospital settings. In their paper, Han et al. [9], describe the design of a queueing-based probabilistic framework, the P-Model, with which uncertainty is adjusted dynamically in order to embed the stochastic nature of the data. After its applications, noticeable reductions in both boarding-time and overflow rate were recorded proving that the method was responsive to demand fluctuations. In addition, Schäfer et al.[19] targeted overflow management with the development of a greedy look-ahead heuristic. This method predicts high inflow and reallocates resources preemptively. Improvements in patient overflow handling were achieved after this method was tested on a large German hospital dataset. The NHSX–Kettering project [16] showed how AI can support clinical environment scheduling, as a sophisticated hybrid approach was applied in the work. Bed demand was predicted with the use of Bayesian time-series forecasting, and allocations were dynamically optimised in real-time using Monte-Carlo Tree Search, paired with a greedy allocator. Finally, Vancroonenburg et al.[21] proposed two Integer Linear Programming models in order to address the room allocation problem in a dynamic setting. New arrivals were handled and optimized by the first model, while future admissions were predicted by the second model. Outcomes showed that the second model outperformed the first, revealing the value of incorporating future arrivals under uncertainty.

Another popular category of methods used to address and optimize problems in healthcare environments is Constraint Programming, that has proven to be effective for handling the complexity of both combinatorial and highly constrained scheduling and allocation problems. Alade and Amusat [4] developed for the needs of the Nurse Scheduling Problem, a single Constraint Programming solver framework, handling varying constraints, such as shift patterns, skill requirements, and staff preferences. Although in some cases metaheuristic approaches (e.g. Particle Swarm Optimization) provided solutions in smaller runtimes, the CP approach successfully modeled practical scheduling needs. Manlove et al. [12] applied Constraint Programming techniques to the Hospitals/Residents problem with Couples (HRC), a problem in which the addition of a constraint that required paired residents to be located together was necessary. Outcomes were promising as the model managed to produce near-stable matchings, often exceeding other metaheuristics' performance. Moreover, Ben Said and Mouhoub's [18] framework consisted of the combination of Machine Learning and Constraint Programming for the nurse assignment problem. Constraint patterns were learned from the data, enabling uncertain demand and adaptive scheduling handling.

An additional parameter many researchers considered within the framework of their projects, but also seemed interesting to study for the current work, was the concept of fairness in hospital infrastructure scheduling. In previous papers, several ways were applied in order to achieve a more equitable resource allocation in medical settings, aiming for enhanced patient-care delivery. A First Come, First-Served policy was introduced by Ala et al. [3] who with the use of NSGA-II and the Whale Optimization Algorithm (WOA) tried to benefit patients that arrived earlier. Their work, incorporating evolutionary heuristics, balanced both fairness and efficiency. Additionally, Lodi et al. [11], with the use of Jain's Index as a guide to fairness, proposed a hybrid column generation and constraint programming framework. Fairness over repeated allocations was achieved, demonstrating how long-term equity can be delivered. More recently, Brandt et al. [5], suggested the addition of roommate compatibility linked constraints like age, language, illness similarity, and surgery timing. CP-inspired Integer Programming formulations with minimum-weight perfect matching were used, exploring a novel fairness dimension beyond mere numerical balance.

Lastly, the work of Jamieson and Forshaw [10] was examined in order to explore how weight tuning can be addressed for further improvements in the modelling procedure as in their paper, Response Surface Methodology (RSM) is applied, a valuable approach for optimizing model performance. Within this framework, they address the challenge of modelling streaming system performance by experimenting with three key designs: 2^k Factorial, Central Composite, and Box-Behnken. Then goodness-of-fit tests were used to evaluate the system's performance under different configurations in a three-step Word Count topology. Their findings show that the combination of Box-Behnken Design and the Epps-Singleton (ES) statistic produced optimal results in high-latency scenarios modeling, suggesting that the choice of the experimental design and evaluation method has an impact on the performance of the model.

To sum up, previous research on methods like metaheuristics, constraint programming, and machine learning was promising while efforts to reflect real-world hospital dynamics led to the incorporation of additional factors such as dynamic and uncertainty-aware frameworks as well as fairness-oriented approaches to their models. This dissertation project situates itself at the intersection of the aforementioned developments as a constraint programming method is used and mechanisms for fairness and uncertainty are incorporated. However, this project differs by combining all these factors to a single model, using the powerful CP-SAT solver for optimised performance and introducing the historical bias violation, a novel constraint for imbalance correction.

3 Methodology

In this section, the main steps of the methodology adopted for the development and refinement of the optimisation models are outlined. Sections on data handling and preparation, the modeling phases followed, the addition of fairness constraints and uncertainty mechanism, as well as the weight tuning procedure, are described.

3.1 Data Handling and Preparation

In order to address the needs of the problem studied in this project, datasets of synthetic benchmark instances were used as data sources (created by Demester et al. [8], used and obtained by Ceschia and Schaerf [6]). To facilitate handling and integration, the files were converted into CSV format and preprocessed. Data types were standarized, and invalid or missing records were filtered out. The final version of the datasets contain fields like patient demographics, needs and preferences, and room characteristics. More details for the data structure and handling process can be found in the Appendix.

3.2 Iterative Development and Early Evaluation

In order to address the Patient Admission Scheduling (PAS) problem, the development of several models was necessary. With the use of a powerful tool, the CP-SAT solver [1], a model that aims to optimize the allocation of hospital rooms to patients was designed, taking into consideration several constraints.

The CP-SAT solver from Google OR-Tools ensures the satisfaction of the hard constraints in order to find a valid solution. Once this is achieved, the solver proceeds to refine it by searching for feasible solutions that better satisfy the objective function. In this case, the aim is to minimise the total penalty every day (Figure 1). The solution space is explored efficiently as CP-SAT uses advanced techniques that quickly narrow down options, avoid repeating bad choices and explore different solutions in a smart and guided way.

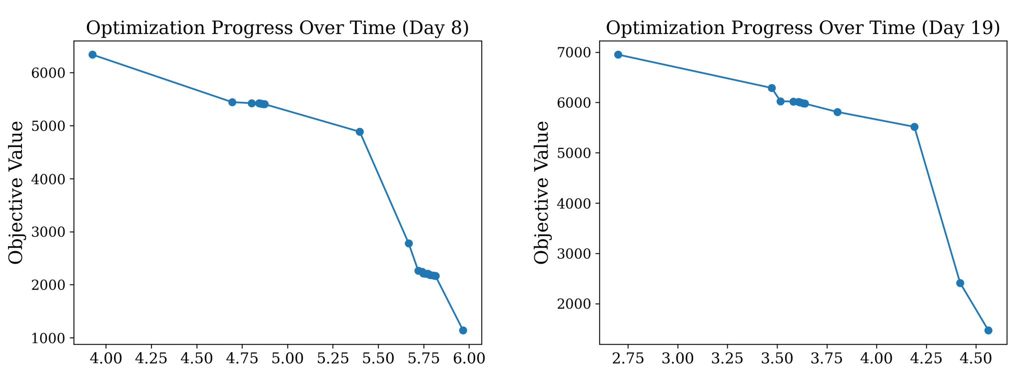


Fig. 1. Optimization progress over time on two typical days of the horizon.

The development of the project was divided into multiple phases, each one followed by an evaluation phase for early validation of the logic and the behavior of the model. In each phase, the model was altered and refined. Each version of the model was saved and labeled for later comparison. A summative table explaining the different versions is given at the end of this section (Table 1).

3.3 Development: Phase 1: The core model - Model 1

In the first phase, the static version of the problem was addressed, where admission and discharge dates were considered predefined. This simplification allowed us to place the focus on developing the base model. Prior work review revealed that a consistent set of constraints was used in the majority of the existing literature[4, 6–8]. After some background studying and analysis on the existing datasets, the main constraints were identified. Hard constraints need to be met, while soft constraints are penalised in the objective function.

Hard Constraints

- **Room assignment:** Each patient must be assigned to exactly one room per day. Let $s_{r,p,d} \in \{0,1\}$ indicate whether patient p is assigned to room r on day d:

$$\sum_{r} s_{r,p,d} = 1 \quad \forall p, d \tag{1}$$

- **Room capacity:** The number of patients being allocated in a room cannot exceed the room's maximum capacity. Let C_r be the capacity of room r:

$$\sum_{p} s_{r,p,d} \leq C_r \quad \forall r, d \tag{2}$$

- **Conditional equipment constraint:** Each patient needs to be assigned to a room equipped with the equipment their treatment requires, unless the equipment is insufficient to accommodate all patients in need. In that case, this requirement is enforced as a soft constraint. Let E be the set of rooms with the required equipment, and P_{need} the set of patients requiring it. Then, if $|E| \geq |P_{\text{need}}|$, the following must hold:

$$\sum_{r \in E} s_{r,p,d} \geq 1 \quad \forall p \in P_{\text{need}}, \forall d \tag{3}$$

Soft Constraints

- **Preferred room capacity:** Patients may prefer to stay in single or shared rooms.
- **Gender mismatch:** Room gender designation should ideally match patient gender.
- **Missing preferred equipment:** Patient preferences regarding equipment should be accommodated for enhanced patient satisfaction.
- **Missing required equipment:** Critical equipment (e.g. telemetry, oxygen) must ideally be present.
- **Patient movement:** Inter-days changes of room should be avoided and minimised during a patient's stay for improoved operational efficiency and patient satisfaction. Patients should remain in the same room during their stay, if possible (patents care needs are considered consistent).
- **Age violation:** Age-based room policies, as defined in the original datasets, should be respected.
- **Specialism compatibility:** Assignments should ideally match patient specialism needs, referring to the type of care a patient needs(e.g., cardiology for cardiac patients). This constraint penalizes minor, major and complete mismatch.

Due to its clinical importance for patients' care, equipment constraint needed to be modeled as a hard constraint. However, as in some cases this led to infeasibility, the constraint now has a soft constraint fallback. In the case that equipment was sufficiently available, hard constraints were enforced, while in the case of shortfalls, the assignment was permitted, but penalised. Additionally, any penalized assignment triggered a message warning about the missing equipment in order to be addressed by the health care professionals more effectively.

3.4 Development: Phase 2: Fairness Constraints - Models 2, 3

The second phase of development focused on enhancing the existing model by introducing the concept of fairness and equity across patients. Two different methods were used to reflect improved ethical principles in hospital care:

Historical Bias Penalties: The model calculated a fairness score for each possible patient-room pair, based on the compatibility penalty of the assignment and the historical bias extracted from patient's penalty from their past assignments. According to this score, patients were categorized into fairness bands (low, medium, high) using percentile thresholds and were differently penalized in the objective function in order to discourage assignments where some patients get repeatedly less ideal rooms.

First-Come First-Served: Inspired by Ala et al. [3] who worked on healthcare appointment scheduling the model incorporated the FCFS logic to achieve additional fairness across the assignments. This approach promotes systematic advantages for newcomers.

3.5 Development: Phase 3: Uncertainty Modeling - Models 4, 5, 6

In order to transform the model into a decision supporting tool in fast-changing environments, controlled uncertainty in patient schedules was introduced to the model. At the start of each day, stochastic variations of patient timelines such as Delayed Admissions (30%), Early Arrivals (30%), Late Discharges (30%) and Early Releases (30%) were incorporated. These changes were dynamically updated and logged, testing the resilience of the models.

3.6 Rolling Horizon

Initial attempts to solve over the complete scheduling period have proven inefficient due to computational expense and lack of real-life applications. Instead, a rolling horizon approach was adopted. For each day, the model checks present patients and new admissions, which can also occur unpredictably. The solver then finds the optimal allocation for that day, the one that gives the least soft constraint violation penalty. This process is repeated until the last day of the planning horizon. This approach not only introduced the concept of realism in the model design process, but also allowed the tracking of historical penalties, a key element for later phases. A flow chart explaining this process can be found in the Appendix.

3.7 Constraint Tuning and Penalty Scaling

The objective function aims to minimise the weighted sum of the soft constraints' penalties, normalised by by the number of assignments, ensuring comparability across days with different number of patients. Weights are aligned to the constraints' relative importance, prioritizing needs over preferences. The final version of the objective function is as follows:

$$\min \frac{1}{N} \sum_{k \in K} W_k \sum s_k \tag{4}$$

where K: set of soft constraint types, W_k: weight of constraint k, s_k: penalty of constraint k and N: total assignments

3.8 Weight Tuning

To further improve the model in terms of fairness, the soft constraints' weights were tuned, aiming to minimize predicted fairness variance. A Response Surface Methodology (RSM)-

inspired approach was adopted, similar to the one Jamieson and Forshaw [10] used in their work. More specifically, soft constraints were grouped based on their importance and experiments were performed with the use of an extended version of 2^k factorial method, where three levels per factor were explored, enabling to accurately capture the data's curvature. Fairness metrics (e.g., fairness standard deviation, Gini, Jain) and total cost were recorded after each experiment and a linear regression model was trained using these outcomes. The model was used to predict the optimal combination of weights that minimize average penalties' standard deviation.

To sum up, the modelling methodology followed a multi-stage process, starting from obtaining and cleaning the datasets, developing a constraint-based model with a rolling-horizon logic, and constraints for equity and uncertain scenarios. Finally, penalty normalization and weight tuning for balanced solutions. These steps positioned the model as a promising tool for practical hospital room allocations.

Model	Uncertainty	Historical Bias	FCFS
Model 1			
Model 2		✓	
Model 3		✓	✓
Model 4	✓		
Model 5	✓	✓	
Model 6	✓	✓	✓

Table 1. Features present in each model (checkmark indicates inclusion).

4 Results

In this section we evaluate the hospital allocation models in terms of effectiveness and robustness, under real-world scenarios of hospital environments. In order to enhance the validity of the results, five different benchmark instances were used (Test3, Test7, Test9, Test12 and Test15), each representing varying sets of hospital configurations (e.g. number of patients and rooms, planning horizons, types of equipment), chosen from the datasets obtained by the authors Carcia and Scharchef [6]. Details about the benchmark instances can be found in the Appendix A.

4.1 Key Metrics

In order to build a robust evaluation framework, a range of metrics and plots were generated and used to calculate and visualize system performance in terms of feasibility, fairness and computational efficiency. More particularly, summary statistics (mean, minimum, and maximum total penalties), fairness indicators like standard deviation, Gini coefficient and Jain's Index, and both daily and overall plots (histograms, pie charts, box plots) were created. A CSV file containing all statistics and a PDF showing all allocations were exported and saved to the project's directory. The full evaluation framework is available in the Appendix.

4.2 Results and Interpretation

In this section, raw and aggregated outcomes from the different models are presented and useful insights are extracted. To begin with, in terms of computational performance, the models demonstrated a 100% success rate for all test datasets as optimal solutions were found across all days of the different planning horizons, regardless of the different characteristics.

As shown in the summative table below, running times varied across models and datasets. Overall, the models were efficient since the maximum solving time recorded was under four minutes. In most cases, models where uncertainty was incorporated were more computationally expensive, something that was expected, as more procedures took place. The most processing-intensive dataset was “Test12” which is logical, given its planning horizon is over 80 days. (Table 5 in Appendix A)

Dataset	**Model 1**	**Model 2**	**Model 3**	**Model 4**	**Model 5**	**Model 6**
Test 3	88.29s	90.45s	90.03s	94.13s	98.59s	98.14s
Test 7	86.16s	85.89s	87.89s	88.57s	86.10s	84.50s
Test 9	138.18s	133.26s	143.33s	142.96s	144.03s	146.97s
Test 12	218.13s	213.15s	214.24s	230.54s	230.45s	237.78s
Test 15	133.91s	132.71s	132.51s	134.85s	136.96s	133.56s

Table 2. Runtime per dataset (seconds).

Most violations follow the same pattern, as they appear to be lower in the first few days of the planning horizon, but risen later with visible fluctuations. For test datasets 3, 7, and 15 preferred capacity and age principal violations are substantially higher than any other constraint cost, followed by specialty and preferred equipment violations, which are lower but distinct. In the remaining datasets, preferred equipment cost is higher, while needed equipment and fairness penalties are more often and greater than before, but still relatively low. Many constraints like gender, move, fairness and needed equipment violations rarely contribute to the overall penalty, validating that the solver aims to avoid penalizing heavily weighted constraints.

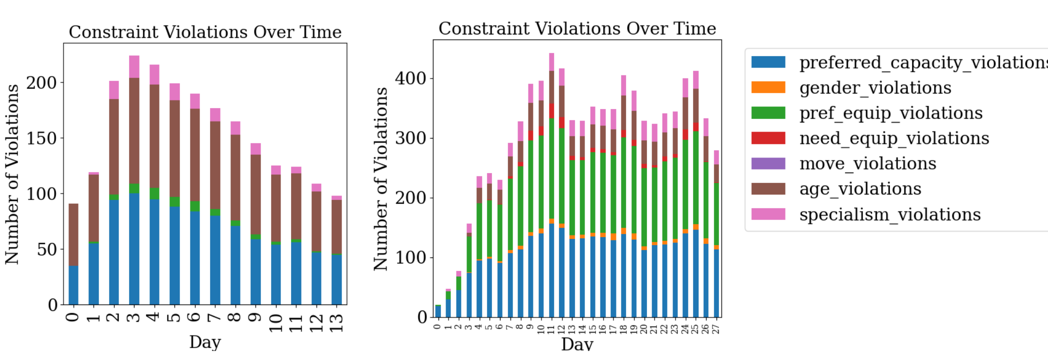


Fig. 2. Constraint violations over time for Test3 and Test9.

Fairness was extensively evaluated in this project, where several metrics such as standard deviation, Gini coefficients, Jain’s Index and multiple plots were dedicated to quantifying equity among patients. More specifically, applying the historical bias constraint achieved improved fairness metrics in most datasets. Standard deviation in Test7 and 12 dropped by 0.07 and 0.09 respectively, while the other metrics improved slightly. The contribution of

First Come First Served mechanism, while a logical approach to implement fairness, was minimal as the fairness metrics showed minimal improvement in most cases, suggesting that the constraint serves an intuitive purpose rather than a numerical one.

Dataset	Metric	Model 4 (U-Basic)	Model 5 (U-Historical)	Model 6 (U-FCFS)
	Std	1.81	1.74	1.75
Test 7	Gini	0.59	0.58	0.61
	Jain	0.44	0.45	0.42
	Std	2.91	2.62	2.62
Test 9	Gini	0.49	0.46	0.46
	Jain	0.52	0.56	0.56
	Std	2.41	2.32	2.32
Test 12	Gini	0.49	0.49	0.49
	Jain	0.54	0.55	0.55

Table 3. Fairness metrics for uncertainty models. Gini Coefficient and Jain's Index range between 0 and 1, Gini is better when closer to 0 and Jain when closer to 1. Full table in Appendix.

Noticeable advancement was visible in dataset 9 where strong fairness outcomes were achieved from fairness constraints. For the basic models, standard deviation stood at 2.60 (Model 1) and 2.91 (Model 4) and after the application of historical penalty constraints dropped to 2.38 and 2.62 accordingly while the addition of the first come first served constraint also helped it decrease at 2.26 and 2.62 respectively. (Results for models 1,2,3 available in the Appendix). Gini Coefficient and Jain's Index were improved by 0.35 on average. These observations are also visible in the accompanying plots of this dataset. After the application of fairness constraints, bar charts appear to be more evenly distributed, as more patients receive closer to average penalties and histograms seem to be less skewed.

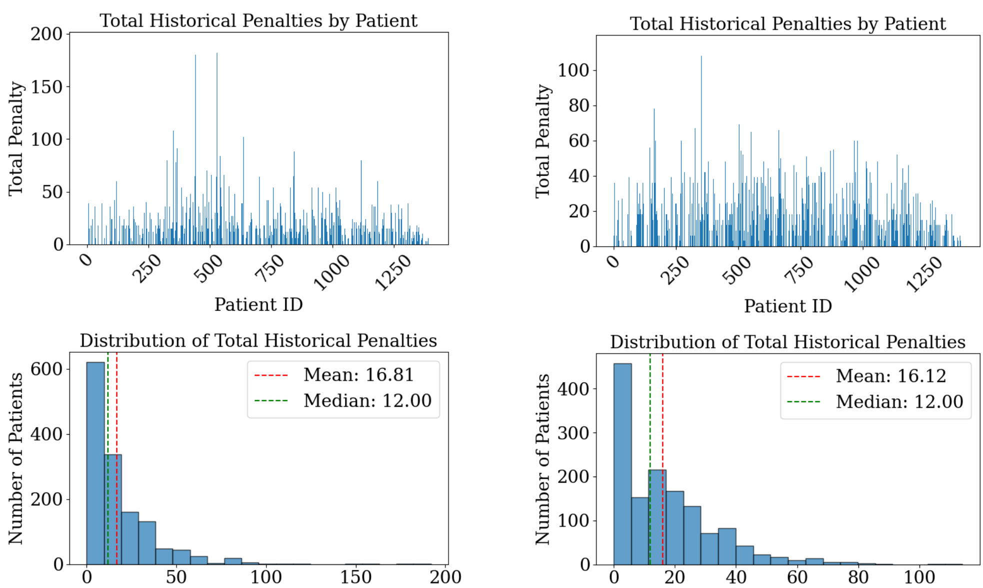


Fig. 3. Distributions of total penalties per patient before (left) and after (right) fairness constraints.

In both kind of plots, (Figure 3) the range of total penalties is narrowed down, approximately halved (from around 200 to 100), suggesting that extreme penalty values have been filtered out. All these are visual indications that fairness is improved as penalties are more uniformly shared across patients, suggesting a more balanced overall allocation.

4.3 Weight Tuning

Attempts to further improve fairness were made by tuning the weights of the objective function. Firstly, it was noted that several weights were strongly correlated with standard deviation. Weights of equipment needs, patient moves and fairness violations seem to be intercorrelated and negatively linked with fairness variance, suggesting a positive effect on overall equity. On the other hand, weights related to preffered capacity, minor specialism capacity, and low fairness showed strong positive relationship, indicating disparity promotion when heavily weighted. The full correlation heatmap is available in the Appendix. The log transformed model demonstrated strong predictive capabilities (Adjusted $R^2 = 0.9356$). Some weights are associated with zero values, something that is not ideal due to their logical importance to the optimisation procedure.

Metric	Before	Optimised
Std	2.62	1.77
Gini	0.46	0.68
Jain	0.56	0.35

Table 4. Fairness and dispersion metrics before versus after optimisation. Gini Coefficient and Jain's Index range between 0 and 1; Gini is better when closer to 0 and Jain when closer to 1.

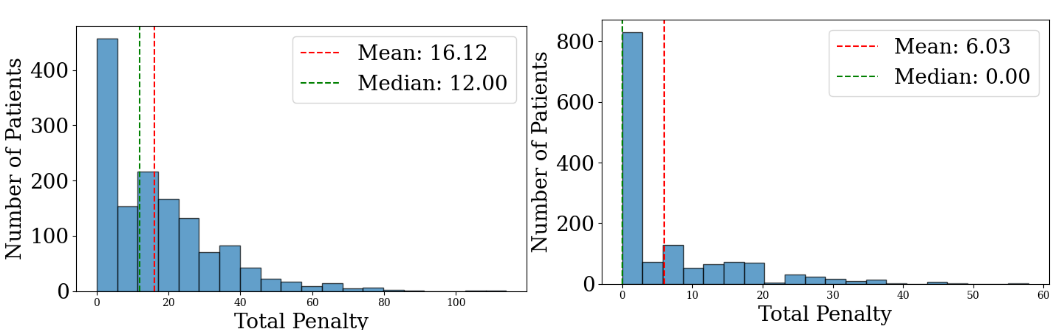


Fig. 4. Distributions of total penalties per patient before (left) and after (right) weight tuning.

After applying the predicted weights (section 3.9 Methodology) in the allocation framework, a noticeable reduction in standard deviation was observed dropping from 2.62 to 1.77. However, Gini coefficient increased from 0.46 to 0.68 and the Jain index dropped from 0.56 to 0.35, suggesting that results are now less equitable than before. Histograms show that the number of patients without penalization rose dramatically (from around 450 to just over 800) and while higher penalties are associated with a few patients, heavy penalties are now lower. Although the decrease of standard deviation indicates fairer allocations for the majority, penalties are concentrated on a few patients illustrating a fairness-efficiency trade-off: improving fairness in one direction (variance reduction) can have negative results in others.

5 Conclusion

This dissertation project explored the development of an optimisation framework using Google's CP-SAT solver in order to address the Patient Admission Scheduling (PAS) problem. A multi-modeling-phase was followed resulting in the design of six constraint-based models evaluated and validated on five synthetic benchmark instances. The models developed incorporated hard, soft and fairness constraints, as well as a stochastic approach to capture the unpredictable nature of real-life hospital settings. Weight tuning explored further improvements in fair allocations, aiming for enhanced resource management, while promoting equity.

5.1 Strengths

The project demonstrated multiple strengths. All models achieved 100% feasibility and found optimal solutions across planning horizons. Runtime was efficient, as the full-horizon, models needed under 4 minutes. Outcomes highlighted that, as stated before, the solver used is robust and scalable, suitable for computationally complex problems. The models adapted to realistic hospital settings, by incorporating a rolling-horizon approach, fairness and uncertainty mechanisms. Fairness enhancements were successful, as historical bias constraints reduced penalty imbalance in most datasets, and weight tuning experimentation resulted in decreased standard deviation. The current developed framework consists of a promising foundation for future work, providing useful insights and offering valuable guidance with the potential to evolve into a strong decision support tool for infrastructure management.

5.2 Limitations

While the models showed several strengths, limitations are unavoidable. To begin with, although fairness-focused constraints reduced disparities in some datasets, successful outcomes were inconsistent, suggesting that success might be data-related. Although FCFS constraint is conceptually logical, its impact was often minimal, revealing limited practical impact. An additional limitation lies in the nature of fairness metrics: although they offer a numerical indication of imbalance, they often overlook elements driven by reasoning (patient needs and urgency). Moreover, models that incorporated uncertainty recorded longer running times. This could make the allocation process more time consuming in real-life scenarios. Finally, weight tuning led to worse overall fairness performance as a narrow definition of fairness (variance reduction) was considered, exposing a trade-off.

5.3 Future Work

Future work could focus on addressing current limitations and also exploring additional aspects of the same problem. Firstly, a future direction could focus on refining current algorithms or adopting alternative approaches, such as multi-objective optimization, where fairness is treated as a co-primary objective in order to achieve more consistent results. Fairness distribution and evaluation could be made more patient-centric and clinically meaningful, rather than quantifiable, considering parameters like comfort, satisfaction, and ethical standards. Expanding the existing models to prioritize high-risk patient cases and address the possibility of overcrowding can enhance the possibilities of a reliable allocation. Lastly, selecting alternative or even multiple fairness metrics simultaneously as the objective for weight tuning optimisation could be explored, aiming for more balanced outcomes.

Acknowledgments

Special thanks to my supervisor, Dr. Matthew Forshaw for his continuous support and guidance throughout the development of this project. Gratitude also goes to the authors Ceschia Schaerf [6, 7] for providing the benchmark datasets used for the testing and validation of the developed models.

References

[1] The CPSAT primer: Basic modeling. https://d-krupke.github.io/cpsat-primer/04_modelling.html.

[2] Accelerated Capability Environment (ACE). Using AI to improve patient bed allocation decisions in hospital. https://www.gov.uk/government/case-studies/using-ai-to-improve-patient-bed-allocation-decisions-in-hospital, Oct. 2023.

[3] A. Ala, F. E. Alsaadi, M. Ahmadi, and S. Mirjalili. Optimization of an appointment scheduling problem for healthcare systems based on the quality of fairness service using whale optimization algorithm and NSGA-II. *Scientific Reports*, 11(1), Oct. 2021.

[4] O. M. Alade and A. O. Amusat. Solving nurse scheduling problem using constraint programming technique, 2019.

[5] T. Brandt, C. Büsing, and F. Engelhardt. Combinatorial and computational insights about patient-to-room assignment under consideration of roommate compatibility, 2025.

[6] S. Ceschia and A. Schaerf. Local search and lower bounds for the patient admission scheduling problem. *Computers Operations Research*, 38(10):1452–1463, 2011.

[7] S. Ceschia and A. Schaerf. Modeling and solving the dynamic patient admission scheduling problem under uncertainty. *Artificial Intelligence in Medicine*, 56(3):199–205, 2012.

[8] P. Demeester, W. Souffriau, P. De Causmaecker, and G. Vanden Berghe. A hybrid tabu search algorithm for automatically assigning patients to beds. *Artificial Intelligence in Medicine*, 48(1):61–70, 2010.

[9] S. Han, S. He, and H. C. Oh. Data-driven inpatient bed assignment using the P model, 2021.

[10] S. Jamieson and M. Forshaw. Reasoning over streaming system performance using response surface methodology. In *2024 IEEE International Conference on Big Data (BigData)*, pages 3702–3711, 2024.

[11] A. Lodi, P. Olivier, G. Pesant, and S. Sankaranarayanan. Fairness over time in dynamic resource allocation with an application in healthcare. *Mathematical Programming*, 203(1–2), Nov. 2022.

[12] D. F. Manlove, I. McBride, and J. Trimble. "almost-stable" matchings in the hospitals / residents problem with couples. *Constraints*, 22(1):50–72, Aug. 2016.

[13] R. Mbau, A. Musiega, L. Nyawira, B. Tsofa, A. Mulwa, S. Molyneux, I. Maina, J. Jemutai, C. Normand, K. Hanson, and E. Barasa. Analysing the efficiency of health systems: A systematic review of the literature. *Applied Health Economics and Health Policy*, 21(2):205–224, Dec. 2022.

[14] M. Mobini. Solving the nurse rostering problem using google or-tools. https://medium.com/@mmobini/solving-the-nurse-rostering-problem-using-google-or-tools-755589b877c0, Oct. 2022.

[15] NHS England. Bed availability and occupancy data. https://www.england.nhs.uk/statistics/statistical-work-areas/bed-availability-and-occupancy/, 2025. Accessed: 4 August 2025.

[16] NHSX AI Lab Skunkworks. NHS AI Lab Skunkworks project: Bed allocation (skunkworks-bed-allocation). https://github.com/nhsx/skunkworks-bed-allocation, 2021.

[17] A. Rojas-García, S. Turner, E. Pizzo, E. Hudson, J. Thomas, and R. Raine. Impact and experiences of delayed discharge: A mixed-studies systematic review. *Health Expectations*, 21(1):41–56, Sept. 2017.

[18] A. B. Said and M. Mouhoub. Machine learning and constraint programming for efficient healthcare scheduling, 2024.

[19] F. Schäfer, M. Walther, A. Hübner, and H. Kuhn. Operational patient-bed assignment problem in large hospital settings including overflow and uncertainty management. *Flexible Services and Manufacturing Journal*, 31(4):1012–1041, Jan. 2019.

[20] The King's Fund. NHS hospital bed numbers: Past, present, future. https://www.kingsfund.org.uk/insight-and-analysis/long-reads/nhs-hospital-bed-numbers, Nov. 2021.

[21] W. Vancroonenburg, P. De Causmaecker, and G. Vanden Berghe. Patient-to-room assignment planning in a dynamic context. In *Proceedings of the 9th International Conference on the Practice and Theory of Automated Timetabling*, pages 193–208, 2012.

Appendix

A Datasets and Pre-Processing

For the aims of this project, five synthetic benchmark instances were used (Test3, Test7, Test9, Test12, and Test15), created by Demeester et al. [8] in order to address the Patient Admission Scheduling (PAS) problem, and were obtained after formal request from Ceschia and Schaerf [6], who also used the full dataset (17 benchmark instances) for the development of their work.

These five were chosen because they contain different combinations of parameter settings allowing the testing and validation of the models in varying levels of complexity and diverse hospital conditions (Table 5).

The datasets were obtained in txt format, and they were then converted into structured CSV files for easier manipulation. Each dataset was converted into two separate CSV files, one containing patient information (Patients.csv) and the other containing room characteristics (Rooms.csv) for improved data handling.

Test Case	**Test 3**	**Test 7**	**Test 9**	**Test 12**	**Test 15**
Rooms	113	162	105	105	348
Beds	395	472	310	340	466
Departments	5	6	4	4	6
Specialisms	5	6	4	4	6
Patients	737	770	1400	2750	890
Planning Horizon	14	14	28	84	28
Properties	1, 2	1, 2, 3, 4	1, 2, 3, 4	1, 2, 3, 4	1, 2, 3, 4
Dept. Age Limits	None	Dep 1: <16, Dep 5: >65	Dep 1: <16, Dep 4: >65	Dep 4: >65	Dep 1: <16, Dep 2: >65, Dep 3: >65

Table 5. Dataset Structure. Association for Properties: 1 = Telemetry; 2 = Oxygen; 3 = Nitrogen; 4 = Television.

(a) Patients Dataset

One record per patient is included with the following attributes:

- PatientID: Unique identifier for each patient.
- Name: Anonymised label
- Age: Patient age in years
- Gender: Patient gender, M (Male) or F (Female)
- AdmissionDay: Planned day a patient enters the hospital
- ReleaseDay: Planned day a patient is discharged from the hospital (must be ≥ AdmissionDay).
- PreferredRoomCapacity: Desired number of roommates (e.g., 1 = single room, 2 = shared).
- Specialism requirements (1–2): One or two types of medical care needed (e.g., cardiology, orthopedics).

- Days of specialism: Days for which a specialism-specific treatment is required.
- NeedsTelemetry / NeedsOxygen / NeedsNitrogen / NeedsTV: Required equipment shown in binary flags (0 or 1).
- PrefersTelemetry / PrefersOxygen / PrefersNitrogen / PrefersTV: Preffered equipment shown in binary flags (0 or 1).

(b) Rooms Dataset (CSV)

One record per hospital room is included with the following attributes:

- RoomID: Unique identifier for each room
- RoomNumber: Room label
- DepartmentID: The department the room belongs to
- Capacity: Number of beds available in that room
- Gender policy: Some rooms are more suitable to accommodate patients of a specific gender:
  - M: Only male patients
  - F: Only female patients
  - N: Neutral/mixed-gender rooms
  - D: Depending on department rules
- Required Specialisms and Penalties: A room can be designated for up to three medical specialties and is associated with a penalty level for misallocations.
- Equipment availability: Availability of Telemetry, Oxygen, Nitrogen, Television shown in binary flags (0 or 1).

Department Age restrictions: Some datasets contain departments where age restrictions are applied. More specifically, paediatric departments require children under 16 years old, while geriatric wards are suitable for patients above 65 years.

The datasets were pre-processed in order to have the right form and be ready to be used in the modeling phase. Different fields were converted into the correct data types, unrealistic or inconsistent records like invalid ages or admission/release days were filtered out. No missing values were found. The finished datasets, the ones that were later used for the test and evaluation of the models, contained fields like patient demographics (PatientID, Name, Age, Gender), admission details (AdmissionDay, ReleaseDay), needs and preferences (e.g. PreferredRoomCapacity, NeedsTelemetry, NeedsOxygen, PrefersNitrogen, PrefersTV), room identification (RoomID, RoomNumber) and characteristics such as capacity, gender policy, available equipment and specialism compatibility.

B Key Metrics

Numeric indicators and visual diagnostics were used in order to build a robust evaluation framework that captures policy compliance, fairness, summary statistics, and system performance.

B.1 Policy compliance

- The model prints and saves statistics such as mean, standard deviation, minimum, and maximum total penalties, average penalties and identification of the most and least penalized patients.
- Stacked bar plot: Gives the breakdown of the total number of violations for each soft constraint, across the days of the planning horizons.

- Pie Chart: Gives the breakdown of the relative contribution of each soft constraint to the total number of violations across the whole horizon (GitHub repository [1](*)).

B.2 Solver Performance Metrics

- Total Solving Time (s): The total time the model needed to run (sum of all daily optimization runtimes).
- Success Rate (%): The number of days of the planning horizon the solver found an optimal or feasible solution.
  Computed as:

$$\text{Success Rate} = \frac{\#\text{days solved}}{\text{planning horizon (days)}} \times 100 \tag{5}$$

B.3 Day-Level Diagnostics

Due to limited space, these plots are available in the GitHub repository of the project [1](*).

- Daily Fairness Thresholds: For each day, both low (33rd percentile) and high (67th percentile) cutoffs of the fairness score for that day are printed and saved.
- Daily Penalty Plots:
  – Bar chart: Total penalties per patient
  – Bar chart: Average penalties per patient
  – Histograms: Distributions of total and average penalties (mean and median markers)
- Patient Timeline: Step plot showing in which rooms the patient was allocated during their stay and whether the patient is moved.

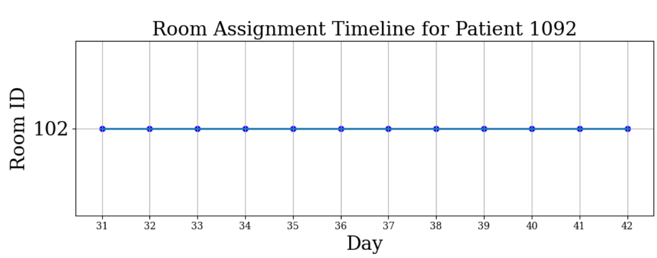


Fig. 5. Rolling-horizon daily optimization workflow for hospital room assignment.

B.4 Fairness Metrics

Fairness was the aim for which most metrics were computed.

- StdDev of Average Daily Penalties: showing whether the penalties varied a lot across patients, indicating unfairness (High spread = inconsistent treatment across patients).

$$\text{StdDev}\left(\frac{\text{penalties}_p}{\text{stay length}_p}\right) \quad \forall p \tag{6}$$

- Gini Coefficient: measures inequality in a distribution, often used in healthcare and AI machine learning fairness sectors [21]. Ranges between 0 and 1 and has better results

[1]GitHub Repository can be found in the end of the Appendix.

when closer to 0.

$$Gini(x) = \frac{\sum_i \sum_j |x_i - x_j|}{2n \sum_i x_i}, \quad x_i \geq 0 \tag{7}$$

- Jain's Index: Ranges between 0 and 1 and has better results when closer to 1.

$$Jain(x) = \frac{\left(\sum_i x_i\right)^2}{n \sum_i x_i^2} \tag{8}$$

- Boxplots: Generated for both total and average penalties showing the spread and outliers across patients (Section F.4 of Appendix)
- Histograms: Distributions of total and average penalties (mean and median markers).
- Bar charts: Total and average penalties across patients.

All plots and statistics are generated and saved in the directory of the project. A CSV containing all statistics and a PDF showing the allocations and patient details across the horizon are exported.

C Rolling Horizon Approach

The Patient Admission Scheduling (PAS) problem is inherently dynamic: new patients arrive, existing patients are discharged, and unforeseen changes in patient schedules occur daily. Attempting to compute a single, static allocation for the entire planning horizon is computationally expensive and unrealistic in practice. To address this, the rolling horizon strategy was implemented.

C.1 Daily Optimization Procedure

(1) Uncertainty Simulation: Stochastic modifications in patients' stay to represent real hospital settings:
   - Delayed Admission: Delay 1-2 days with 30% probability
   - Early Arrival: Advance admission 1-2 days with 30% probability
   - Late Discharge: Extend stay by 1-2 days with 30% probability
   - Early Release: Shorten stay by 1-2 days with 30% probability

(2) New admission and release days are updated.

(3) Patient Pool Selection: Identify present patients:

$$\text{admission day} \leq d < \text{release day}$$

(4) Fairness Banding: Calculation of thresholds for categorization of patients, according to their penalties (low, medium, or high fairness).

(5) Optimization: Solve the daily room allocation problem by finding the optimal allocation that minimises the penalty of the objective function and satisfies the hard constraints.

(6) Result Storage:
   - Room assignments for each patient
   - Violations of each constraint type.

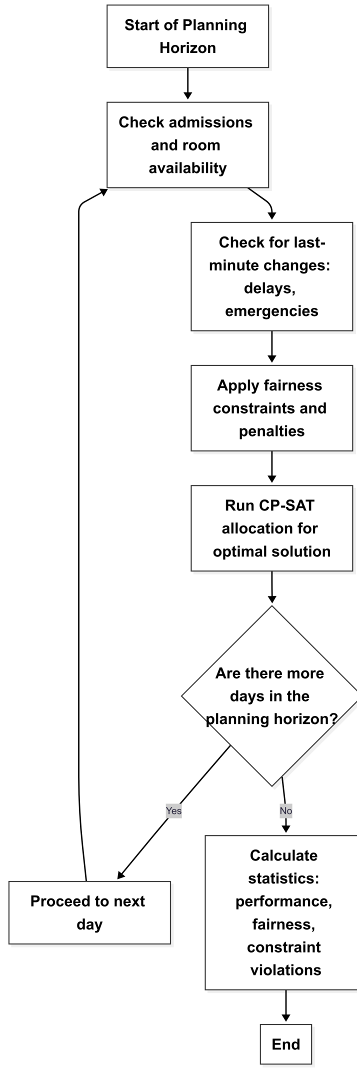


Fig. 6. Rolling-horizon daily optimization workflow for hospital room assignment.

D Further Fairness Results

As mentioned in the report, applying the historical bias constraint achieved improved fairness metrics in most datasets. Standard deviation in Test7, Test9 and Test12 dropped by 0.07, 0.29 (Model 5) and 0.09 respectively, while the other metrics also improved slightly. After the application contribution of First Come First Served mechanism, fairness metrics showed minimal improvement in most cases.

Outcomes for all models and datasets are available in the tables that follow. Overall, all the visual indications and statistics suggest improved fairness as penalties are more uniformly shared across patients.

Dataset	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
Test 3	1.34	1.36	1.35	1.55	1.60	1.57
Test 7	1.65	1.71	1.66	1.81	1.74	1.75
Test 9	2.60	2.38	2.26	2.91	2.62	2.62
Test 12	2.04	2.03	2.01	2.41	2.32	2.32
Test 15	1.86	1.93	1.88	2.02	2.01	2.02

Table 6. Standard Deviation for each model on the benchmark datasets.

Dataset	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
Test 3	0.71	0.71	0.71	0.76	0.75	0.75
Test 7	0.54	0.55	0.54	0.59	0.58	0.61
Test 9	0.44	0.42	0.41	0.49	0.46	0.46
Test 12	0.43	0.43	0.42	0.49	0.49	0.49
Test 15	0.63	0.65	0.63	0.67	0.67	0.67

Table 7. Gini Coefficient for each model on the benchmark datasets. Gini Coefficient ranges between 0 and 1 and is better when closer to 0.

Dataset	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
Test 3	0.30	0.30	0.30	0.26	0.27	0.27
Test 7	0.48	0.48	0.48	0.44	0.45	0.42
Test 9	0.57	0.61	0.62	0.52	0.56	0.56
Test 12	0.61	0.61	0.61	0.54	0.55	0.55
Test 15	0.38	0.36	0.38	0.35	0.34	0.34

Table 8. Jain's Index for each model on the benchmark datasets. Jain's Index ranges between 0 and 1 and is better when closer to 1.

E Mathematical Model

In this section of the appendix, the full Mathematical Model is explained in detail, and all notations are made available.

E.1 Decision Variables

If a patient p is placed in room r on a specific day d, then this binary flag is true. Let: $s_{r,p,d} \in \{0, 1\}$.

E.2 Hard Constraints: Need to be met in all allocations

1. Single-room assignment Each patient needs to be assigned to exactly one room (neither unassigned, nor double-booked). This translates into: For every day d and each patient p, the total number of rooms a patient is allocated in (sum of the assignment variable $s_{r,p,d}$ over all rooms) is equal to 1.

$$\sum_r s_{r,p,d} = 1 \quad \forall p, \forall d \tag{9}$$

2. Room capacity The number of patients being allocated in a room cannot exceed the room's maximum capacity. This translates into: For every room r and every day d, the total number of patients assigned in the room r (the sum of the assignment variable $s_{r,p,d}$ over all patients) cannot exceed the room's capacity C_r.

$$\sum_p s_{r,p,d} \leq C_r \quad \forall r, \forall d \tag{10}$$

3. Conditional equipment constraint Each patient needs to be assigned to a room equipped with the equipment their treatment requires, unless the equipment is insufficient to accommodate all patients in need. In that case, this requirement is enforced as a soft constraint. This translates into: If on day d, the number of required equipment is greater than or equal to the number of patients needing the equipment, then for every patient p that needs this equipment and for every day d, there must be at least one assignment placing them in a room where the equipment is available (the sum of the assignment variables $s_{r,p,d}$ over all rooms r that have the equipment must be at least 1).

Let:

- E: set of rooms with required equipment
- P^{need}: set of patients requiring the equipment.

If $|E| \geq |P^{need}|$, then:

$$\sum_{r \in E} s_{r,p,d} \geq 1 \quad \forall p \in P^{need}, \forall d \tag{11}$$

E.3 Soft Constraints

Each constraint violation adds a penalty to the objective function. Indicator functions $\{\cdot\}$ flag violations.

- **Preferred room capacity:** Patients may prefer to stay in single or shared rooms. For patient p on day d, one violation is added if the room they are going to be assigned to has more beds than the patient prefers. Let C_r be the capacity of room r, and c_p^{pref} the preferred capacity for patient p. Violations occur when:

$$capViol_{p,d} = \sum_r s_{r,p,d} \, \mathbb{1}\{C_r > c_p^{\text{pref}}\} \tag{12}$$

- **Gender mismatch:** Room gender designation should ideally match patient gender. For patient p on day d, one violation is added if the room they are going to be assigned to carry gender restrictions that do not match patients' gender. Let g_r and g_p denote the gender of room r and patient p, respectively:

$$genderViol_{p,d} = \sum_r s_{r,p,d} \, \mathbb{1}\{g_r \in \{M, F\} \wedge g_r \neq g_p\} \tag{13}$$

- **Missing preferred equipment:** Patient preferences regarding equipment should be accommodated for enhancing patient satisfaction. For patient p on day d, one violation is added if the room they are going to be assigned to is missing at least one piece of equipment the patient prefers. Let $e_{r,i}$ indicate room r has equipment i, and $n_{p,i}^{\text{pref}} = 1$ if patient p prefers equipment i:

$$prefEqViol_{p,d} = \sum_r s_{r,p,d} \, \mathbb{1}\{\exists\, i \in \{1, \ldots, 4\} : \; n_{p,i}^{\text{pref}} = 1 \wedge e_{r,i} = 0\}. \tag{14}$$

- **Missing required equipment:** Critical equipment must ideally be present. For patient p on day d, one violation is added if the room they are going to be assigned to is missing at least one piece of equipment the patient needs. Let $n_{p,i}^{\text{need}} = 1$ if patient p needs equipment i:

$$needEqViol_{p,d} = \sum_r s_{r,p,d}\, \mathbb{1}\{\exists\, i \in \{1,\dots,4\} :\ n_{p,i}^{\text{need}} = 1 \wedge e_{r,i} = 0\}. \tag{15}$$

- **Patient movement (continuity of care):** Changes of room should be avoided and minimised during a patient's stay for improved operational efficiency and patient satisfaction. For patient p on day d, one violation is added if the room they are going to be assigned to is different from the one they were assigned to the previous day. Let $r_{p,d-1}$ be the previous day's room for patient p:

$$moveViol_{p,d} = \sum_r s_{r,p,d}\, \mathbb{1}\{r \neq r_{p,d-1}\}. \tag{16}$$

- **Age policy violation:** The age-based room policies certain departments have should be respected. Age thresholds are provided in the original datasets and vary. For patient p on day d, one violation is added if the room they are going to be assigned to carries an age-policy that is not aligned with the patient's age. In this case department 1 is geriatrics and department 4 is pediatrics. Let a_p be the age of patient p, and d_r the department of room r:

$$ageViol_{p,d} = \sum_r s_{r,p,d}\, \mathbb{1}\{(d_r = 1 \wedge a_p < 65)\ \vee\ (d_r = 4 \wedge a_p > 16)\}. \tag{17}$$

- **Specialism compatibility:** Assignments should ideally match patient specialism needs, referring to the type of care a patient requires according to their condition (e.g., cardiology for cardiac patients). Let spec_p be the patient's required specialism, spec_r the set of supported specialisms by room r, and $pen_{p,r}$ the associated penalty level (found in datasets):
  Mismatch: For patient p on day d, one violation is added if the room they are going to be assigned to does not support the medical specialisms the patient requires.

$$specMismatch_{p,d} = \sum_r s_{r,p,d}\, \mathbb{1}\{\text{spec}_p \notin \text{spec}_r\} \tag{18}$$

  Minor mismatch: For patient p on day d, one violation is added if the room they are going to be assigned to supports the required specialisms, but carries a penalty level of 2.

$$specMinor_{p,d} = \sum_r s_{r,p,d}\, \mathbb{1}\{\text{pen}_{p,r} = 2\} \tag{19}$$

  Major mismatch: For patient p on day d, one violation is added if the room they are going to be assigned to supports the required specialisms, but carries a penalty of 3 or higher. This constraint is associated with heavier penalty than the other specialism constraints.

$$specMajor_{p,d} = \sum_r s_{r,p,d}\, \mathbb{1}\{\text{pen}_{p,r} \geq 3\} \tag{20}$$

- **Historical Bias Penalties:** Patients with high penalties in the previous days of their stay should get better allocations, so as the overall penalty for each patient to be around the same level. Let:
  - $H_p(d-1)$: how much penalty a patient p has up until the previous day.

- $basePen(p, r)$: how compatible (what is the compatibility penalty) is assigning a patient p to a room r on day d. The compatibility penalty of allocating a patient p to a room r on a day d is the weighted sum of all violated conditions.

$$\begin{aligned} basePen(p, r) = {} & W_{cap}[\cdot] + W_{gen}[\cdot] + W_{peq}[\cdot] + W_{neq}[\cdot] + W_{age}[\cdot] \\ & + W_{\text{spec\_minor}}[\cdot] + W_{\text{spec\_major}}[\cdot] + W_{\text{spec\_mismatch}}[\cdot] \end{aligned} \tag{21}$$

- $fairScore_{p,r,d} = basePen(p, r)\left(1 + \frac{H_p(d-1)}{\max(1,d+1)}\right)$: adjusted fairness score for assignment.
- Low threshold: $q_{33}(d)$: the 33rd percentile thresholds of fairness scores of all patients' assignments (1/3 of the possible fairness scores are below this value)
- High threshold: $q_{67}(d)$: the 67th percentile thresholds of fairness scores of all patients' assignments (1/3 of the possible fairness scores are above this value)

Assignments are then penalized based on which of the three bands they fall into. Higher fairness band cases are heavily penalised in order to discourage the solver for continuing inequitable allocations.
Low fairness band: For patient p on day d, one violation is added if the room they are going to be assigned to has a fairness score that is lower than the low threshold.

$$fairLow_{p,d} = \sum_r s_{r,p,d}\, \mathbb{1}\{fairScore_{p,r,d} < q_{33}(d)\} \tag{22}$$

Medium fairness band: For patient p on day d, one violation is added if the room they are going to be assigned to has a fairness score that is between the low and high thresholds.

$$fairMed_{p,d} = \sum_r s_{r,p,d}\, \mathbb{1}\{q_{33}(d) < fairScore_{p,r,d} \leq q_{67}(d)\} \tag{23}$$

High fairness band: For patient p on day d, one violation is added if the room they are going to be assigned to has a fairness score that is higher than the high threshold.

$$fairHigh_{p,d} = \sum_r s_{r,p,d}\, \mathbb{1}\{fairScore_{p,r,d} > q_{67}(d)\} \tag{24}$$

- **First-Come First-Served (FCFS):** Patients who arrived earlier in the hospital should be allocated in rooms that match their needs and preferences at a higher level than patients arriving afterwards. Let: M_{d-1} be the maximum penalty experienced by any patient on the previous day. For patient p that was newly admitted on day d, one violation is added if the room they are going to be assigned to has a compatibility penalty (base penalty) strictly better than the historical maximum (maximum penalty of the previous day).

$$fcfsViol_{p,d} = \sum_r s_{r,p,d}\, \mathbb{1}\{newAdmission(p, d) \ \wedge \ basePen(p, r) < M_{d-1}\}. \tag{25}$$

E.4 Objective Function

All soft constraints' penalties are weighted according to the relative importance of the constraint, prioritizing needs over preferences, and added in the objective function after

they are normalised by $N_d = |P_d| \cdot |R|$. Different weight combinations can result in altered allocations.
The model minimizes the following objective function:

$$\begin{aligned}
\min \sum_{d\in D} \frac{1}{N_d} \Big(& W_{cap} \sum_p capViol_{p,d} + W_{gen} \sum_p genderViol_{p,d} + W_{peq} \sum_p prefEqViol_{p,d} \\
& + W_{neq} \sum_p needEqViol_{p,d} + W_{age} \sum_p ageViol_{p,d} + W_{specmin} \sum_p specMinor_{p,d} \\
& + W_{specmaj} \sum_p specMajor_{p,d} + W_{specmis} \sum_p specMismatch_{p,d} \\
& + W_{fairL} \sum_p fairLow_{p,d} + W_{fairM} \sum_p fairMed_{p,d} + W_{fairH} \sum_p fairHigh_{p,d} \\
& + W_{move} \sum_p moveViol_{p,d} + W_{fcfs} \sum_p fcfsViol_{p,d} \Big) \qquad (26)
\end{aligned}$$

F Weight Tuning Procedure and Results

In this section, additional information regarding the weight tuning procedure and results are provided. This method was applied on Model 5 that incorporates Uncertainty and Historical Bias Penalties and tested on dataset Test9.

F.1 Experimental Setup

Soft constraints were grouped based on their importance and four (A-D) experiments were performed with the use of an extended version of 2k factorial method, where three levels per factor were explored, enabling to accurately capture the data's curvature. Experiments were grouped as follows:

Weights	**Weight Range**	**Experiment**
Wmove	[5, 8, 10]	A
Wfair_high	[5, 7, 9]	A
Wneq	[6, 8, 10]	A
Wspec_mismatch	[4, 6, 7]	B
Wage	[4, 6, 8]	B
Wspec_major	[4, 6, 8]	B
Wfair_med	[2, 3, 5]	C
Wgen	[2, 4, 6]	C
Wpeq	[3, 5, 7]	C
Wcap	[2, 3, 5]	D
Wspec_minor	[1, 2, 3]	D
Wfair_low	[0, 1, 2]	D

Table 9. Experiments grouped by the importance of weights

For each weight combination, Fairness Standard Deviation, Gini Coefficient, Jain Index, Total Cost and Status (Feasibility) were recorded and saved into a CSV file.

F.2 Correlation Analysis

Correlations between all weight parameters, fairness standard deviation, and total cost.

- Negatively correlation with fairness variance: $W_{neq}, W_{gen}, W_{peq}, W_{fair}^{high}, W_{move}$
- Positive correlation with fairness variance: $W_{cap}, W_{spec}^{minor}, W_{fair}^{low}$ (disparity is promoted)
- Highest intercorrelations: W_{neq} and W_{move} (0.95), W_{move} and W_{fair_high} (0.94), W_{spec_major} and $W_{spec\_mismatch}$ (0.92), and W_{age} and W_{spec_major} (0.91), suggesting that they are dependent, change together in same directions and also influence similar outcomes.

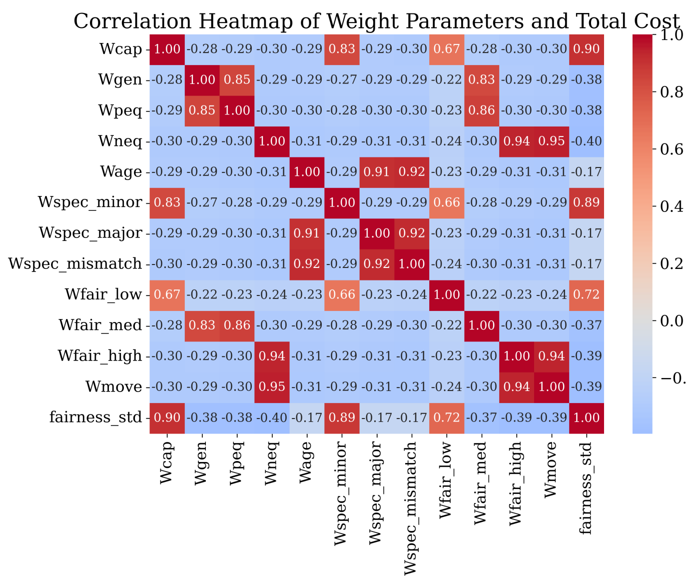


Fig. 7. Correlation heatmap of weight parameters.

F.3 Regression Modeling

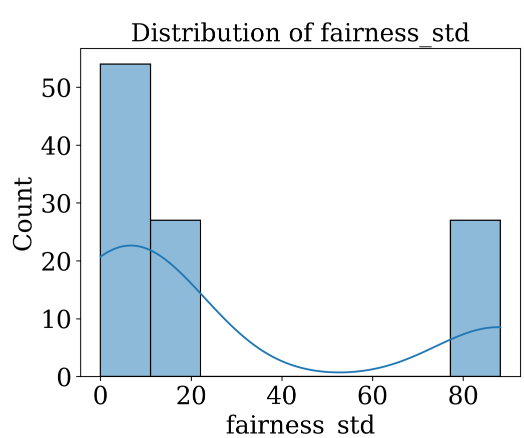


Fig. 8. Distribution plots of fairness Standard Deviation as extracted from test cases.

As observed, the distribution of standard deviation was highly right-skewed as most of the values were close to zero. In order to get a more stabilised image of standard deviation for regression purposes, a logarithmic transformation of standard deviation ($fairness_std$-targert variable) was applied and then a multiple linear regression model was trained on the outcomes of the experiments.

- Train R^2: 0.9450
- Test R^2: 0.9506
- Adjusted R^2: 0.9356

The model's coefficients and the predicted optimal combinations for minimised fairness standard deviation are showed above: Predicted minimum fairness standard deviation: 0.0000 (theoretical). The fact that some weights are associated with zero values is not ideal as they have a logical importance to the optimisation procedure.

Weights	**Coefficients**
Wmove	0.0256
Wfair_high	-0.0650
Wneq	-0.7120
Wspec_mismatch	-0.0003
Wage	-0.0002
Wspec_major	-0.0001
Wfair_med	0.1949
Wgen	-0.5028
Wpeq	-0.1271
Wcap	0.1806
Wspec_minor	0.3300
Wfair_low	0.1762

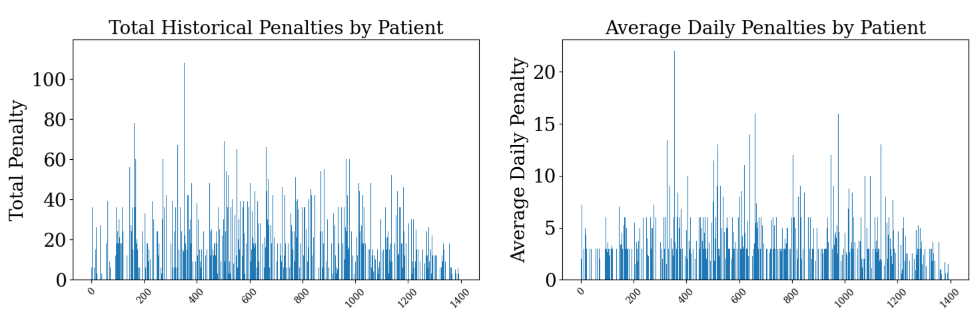
Table 10. Model coefficients (log-transformed target, *Regression intercept: 2.8795*)

Weights	**Optimised Value**
Wmove	0.5257
Wfair_high	2.2058
Wneq	4.1758
Wspec_mismatch	1.0049
Wage	1.0031
Wspec_major	1.0049
Wfair_med	0.0000
Wgen	10.3243
Wpeq	3.3562
Wcap	0.0000
Wspec_minor	0.0000
Wfair_low	0.0000

Table 11. Optimized Weights

F.4 Further Results

Penalties in bar plots seem to be more dispersed before tuning while after the optimisation all columns are narrowed down, with the majority having values around zero. Several bars seem to be taller indicating that although most patients get low penalties, penalties are concentrated on a few people.



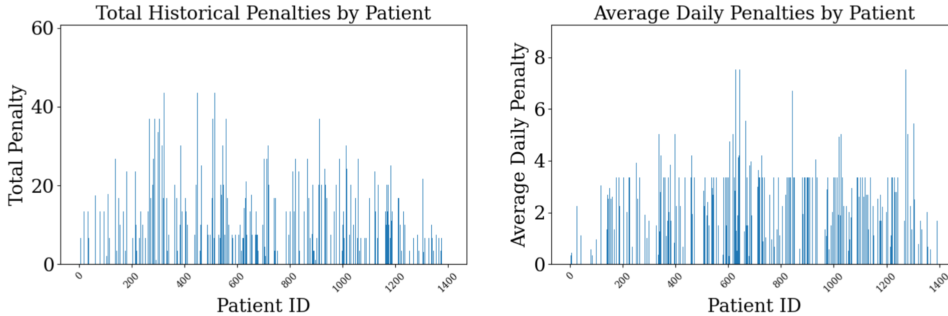


Fig. 9. Distribution plots of penalties before (above) and after (below) weight tuning.

Before the application of the optimal combination of weights, penalties in boxplots are shown to be more widespread with a higher median and multiple outliers, suggesting uneven distribution of penalties across patients. On the other hand, after the application of the predicted weights, the spread appears to be narrower, and the median dropped close to zero. Outliers are reduced and less extreme than before.

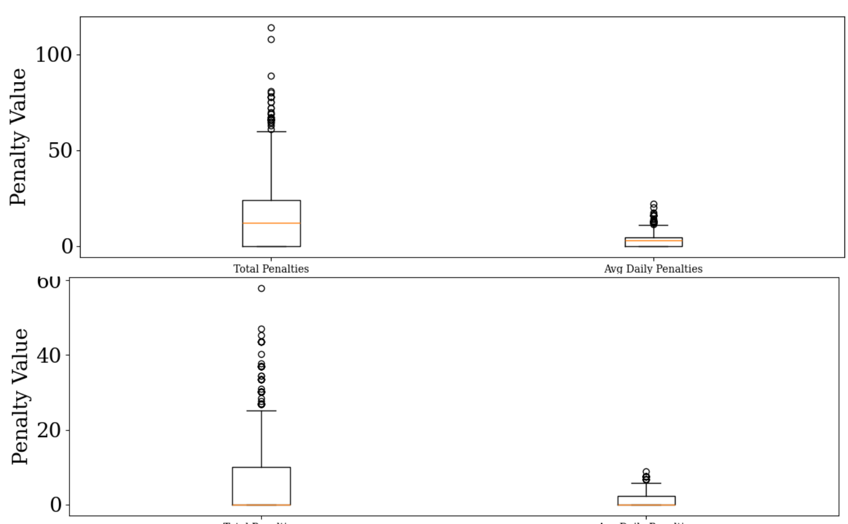


Fig. 10. Boxplots of distribution of penalties before (above) and after (below) weight tuning.

All visualisations, suggest that after the optimisation of the weights, penalties are now concentrated on fewer patients, while most patients have lower penalties, explaining the rise in Gini coefficient and drop in Jain's index, even as the standard deviation improved, suggesting unfair overall allocations.

(*) Due to limited space, further results and plots are available in the GitHub repository of the project. **https://github.com/Andrieta64/patient-scheduling-cpsat/tree/main**