Solving the Hospital Patient Allocation Problem: A Comparative Study of Exact Methods, Metaheuristics, and Matheuristic Approaches

Paolo Pascarelli
João Filipe
Diogo Teixeira
Faculdade de Engenharia da Universidade do Porto
Introduction to Artificial Intelligence
January 2026

Abstract

This work presents a comprehensive study of optimization approaches for the Hospital Patient Allocation Problem, a bi-objective scheduling problem that simultaneously minimizes operational costs and balances workload across hospital wards. We formulate the problem as a Mixed Integer Linear Program (MILP) and develop a suite of solution methodologies: exact methods using Branch & Bound, metaheuristics including Iterated Local Search (ILS) and Variable Neighborhood Search (VNS), and a novel matheuristic approach that combines metaheuristic warm-starting with exact optimization. Our computational experiments on benchmark instances demonstrate the trade-offs between solution quality and computational efficiency across these approaches. Results indicate that while exact methods achieve optimality for small to medium instances, metaheuristics provide competitive solutions with significantly reduced computation times for larger instances. The proposed matheuristic approach effectively bridges this gap, leveraging the rapid exploration capabilities of metaheuristics to accelerate the convergence of exact methods.

1 Introduction

The efficient management of hospital resources has become increasingly critical as healthcare systems worldwide face mounting pressure from aging populations and rising demand for medical services. Among the various operational challenges hospitals face, patient admission scheduling represents a particularly complex decision-making problem that directly impacts both operational efficiency and staff well-being.

The Hospital Patient Allocation Problem involves assigning a set of patients to hospital wards on specific admission days while respecting numerous constraints and optimizing multiple competing objectives. Each patient requires admission within a specific time window, has an associated surgical procedure with a deterministic duration, and occupies a bed for a known length of stay. Hospital wards are characterized by their bed capacity, workload capacity, and specialization in particular surgical disciplines.

This problem is inherently bi-objective in nature. On one hand, hospital administrators seek to minimize operational costs, which include penalties for admission delays and deviations from planned operating theatre utilization. On the other hand, ensuring sustainable working conditions requires balancing workload both spatially (across wards) and temporally (across days). These objectives often conflict: achieving perfect workload balance may necessitate delaying patient admissions, while minimizing delays may result in uneven workload distributions.

The scientific contribution of this work is threefold. First, we present a complete mathematical formulation of the bi-objective patient allocation problem as a Mixed Integer Linear Program. Second, we develop and implement a comprehensive set of solution methodologies spanning exact methods, metaheuristics, and hybrid approaches. Third, we conduct an extensive computational study that provides insights into the performance characteristics of each approach and identifies conditions under which each methodology is most appropriate.

The remainder of this paper is organized as follows. Section 2 provides a formal problem definition and mathematical formulation. Section 3 describes our solution methodologies. Section 4 presents the computational study and discusses our findings. Finally, Section 5 concludes with recommendations for practitioners and directions for future research.

2 Problem Definition and Mathematical Formulation

2.1 Problem Description

We consider a hospital with a set of wards $\mathcal{W}$, a planning horizon of $\mathcal{D}$ days, and a set of patients $\mathcal{P}$ requiring admission. Each ward $w \in \mathcal{W}$ is characterized by its bed capacity b_w, workload capacity β_w, a major specialization, and potentially several minor specializations. The set of surgical disciplines is denoted $\mathcal{S}$.

Definition 1 (Patient Characteristics)**.** *Each patient $p \in \mathcal{P}$ is characterized by a tuple $(s_p, f_p, e_p, l_p, u_p, \theta_p)$ where:*

- *$s_p \in \mathcal{S}$ is the required surgical specialization*
- *$f_p \in \mathcal{D}$ is the earliest possible admission day*
- *$e_p \in \mathcal{D}$ is the latest possible admission day*
- *$l_p \in \mathbb{Z}^+$ is the length of stay (in days)*
- *$u_p \in \mathbb{Z}^+$ is the surgery duration (in minutes)*
- *$\theta_p = (\theta_{p,1}, \ldots, \theta_{p,l_p})$ is the daily workload vector*

A patient can only be admitted to a ward that has either a major or minor specialization in the patient's required surgical discipline. When admitted to a ward with minor specialization, the patient's workload is scaled by a factor $\gamma_s > 1$, reflecting the additional effort required by staff less specialized in that discipline.

2.2 Decision Variables

The model employs the following decision variables:

$y_{pwd} \in \{0,1\}$	Binary: patient p admitted to ward w on day d	(1)
$x_{wd} \geq 0$	Continuous: normalized workload of ward w on day d	(2)
$z \geq 0$	Continuous: maximum workload across all ward-day pairs	(3)
$v_{sd} \geq 0$	Continuous: overtime for specialization s on day d	(4)
$u_{sd} \geq 0$	Continuous: undertime for specialization s on day d	(5)

2.3 Objective Functions

The bi-objective problem minimizes two competing objectives:

Objective 1 (Operational Cost):

$$f_1 = W^{OT} \sum_{s\in\mathcal{S}} \sum_{d\in\mathcal{D}} v_{sd} + W^{UT} \sum_{s\in\mathcal{S}} \sum_{d\in\mathcal{D}} u_{sd} + W^{WAIT} \sum_{p\in\mathcal{P}} \sum_{w\in\mathcal{W}_p} \sum_{d=f_p+1}^{e_p} (d - f_p) \cdot y_{pwd} \tag{6}$$

Objective 2 (Workload Balance):

$$f_2 = z = \max_{w\in\mathcal{W}, d\in\mathcal{D}} x_{wd} \tag{7}$$

The min-max formulation for workload balance ensures both spatial balance (similar workload across wards on any given day) and temporal balance (consistent workload for each ward across days). This equity function satisfies the Pigou-Dalton transfer principle: any transfer of workload from a heavily loaded ward to a lightly loaded ward cannot increase the objective value.

2.4 Constraints

Assignment Constraint: Each patient must be admitted exactly once:

$$\sum_{w\in\mathcal{W}_p} \sum_{d\in\mathcal{D}_p^A} y_{pwd} = 1 \quad \forall p \in \mathcal{P} \tag{8}$$

Operating Theatre Capacity: The OT time for each specialization is bounded:

$$\sum_{p\in\mathcal{P}_{sd}} \sum_{w\in\mathcal{W}_p} u_p \cdot y_{pwd} + u_{sd} - v_{sd} = q_{sd} \quad \forall s \in \mathcal{S}, d \in \mathcal{D} \tag{9}$$

Workload Computation: The normalized workload is computed as:

$$x_{wd} = \frac{1}{\beta_w} \left(\nu_{wd} + \sum_{p\in\mathcal{P}_w} \sum_{d'=\max(f_p, d-l_p+1)}^{\min(e_p, d)} \theta_{p,d-d'+1} \cdot \gamma(p,w) \cdot y_{pwd'} \right) \quad \forall w \in \mathcal{W}, d \in \mathcal{D} \tag{10}$$

where ν_{wd} represents carryover workload from previous planning periods and $\gamma(p, w)$ is the specialization scaling factor.

Max Workload Definition:

$$z \geq x_{wd} \quad \forall w \in \mathcal{W}, d \in \mathcal{D} \tag{11}$$

Bed Capacity:

$$\sum_{p\in\mathcal{P}_w} \sum_{d'=\max(f_p, d-l_p+1)}^{\min(e_p, d)} y_{pwd'} \leq b_w - h_{wd} \quad \forall w \in \mathcal{W}, d \in \mathcal{D} \tag{12}$$

where h_{wd} denotes beds occupied by carryover patients.

2.5 Scalarization

To solve the bi-objective problem, we employ weighted-sum scalarization:

$$\min \quad \lambda_1 \cdot f_1 + \lambda_2 \cdot f_2 \tag{13}$$

By systematically varying the weights λ_1 and λ_2, we can approximate the Pareto front. However, since the feasible region is discrete (due to binary variables), the objective function landscape is non-convex, preventing complete enumeration of all Pareto-optimal solutions through weighted-sum scalarization alone.

3 Solution Methodologies

3.1 Exact Method: Mixed Integer Linear Programming

The formulation presented in Section 2 is a Mixed Integer Linear Program (MILP) that can be solved using commercial solvers such as Gurobi or open-source alternatives like CBC. The solver employs Branch & Bound with cutting planes, exploiting the linear relaxation at each node to compute lower bounds.

Theorem 1 (Complexity)**.** *The Hospital Patient Allocation Problem is NP-hard, as it generalizes the multiple-choice knapsack problem with additional scheduling constraints.*

Despite the theoretical intractability, modern MILP solvers can effectively solve instances of practical size. The problem structure exhibits favorable characteristics for Branch & Bound:

- The majority of variables are binary, enabling aggressive pruning
- Linear relaxations are relatively tight due to the assignment structure
- Constraint propagation effectively reduces the search space

3.2 Metaheuristics

To address larger instances where exact methods become computationally prohibitive, we developed a metaheuristic pipeline consisting of four stages.

3.2.1 Stage 1: Greedy Construction

The construction phase builds an initial feasible solution by processing patients in order of urgency (smallest admission window first). For each patient, the algorithm attempts to assign them to the earliest feasible day and a compatible ward that minimizes capacity violations.

Algorithm 1 Greedy Feasible Construction

```
1: Sort patients by $(e_p - f_p, f_p, -l_p)$                ▷ Window size, earliest day, LOS
2: $\mathcal{A} \leftarrow \emptyset$                        ▷ Empty allocation
3: for each patient $p$ in sorted order do
4:     $\mathcal{W}_p \leftarrow$ compatible wards for $p$
5:     for each day $d \in [f_p, e_p]$ do
6:         for each ward $w \in \mathcal{W}_p$ do
7:             if BedsUsed$(\mathcal{A}, w, d) < b_w$ then
8:                 $\mathcal{A}[p] \leftarrow (w, d)$
9:                 break
10:            end if
11:        end for
12:        if $p$ is assigned then break
13:        end if
14:    end for
15: end for
16: return $\mathcal{A}$
```

3.2.2 Stage 2: Local Search

Starting from the greedy solution, deterministic local search iteratively improves the solution by examining patient-level moves. For each patient, ordered by their contribution to the delay cost, the algorithm evaluates:

- **Day shifts**: Moving to a different day within the admission window
- **Ward transfers**: Moving to a different compatible ward

Only moves that strictly improve the objective while maintaining feasibility are accepted.

3.2.3 Stage 3: Iterated Local Search (ILS)

ILS escapes local optima through controlled perturbations followed by local search refinement.

Definition 2 (Perturbation Strategy)**.** *The mixed perturbation selects $\rho \cdot |\mathcal{P}|$ patients to reassign, where $\rho \in [0.1, 0.25]$ adapts based on search stagnation. Half are selected randomly; half are the worst contributors to the delay objective.*

The algorithm employs a penalty-based approach to temporarily explore infeasible solutions:

$$\hat{f}(\mathcal{A}) = f(\mathcal{A}) + \mu \cdot \sum_{w \in \mathcal{W}} \sum_{d \in \mathcal{D}} \max(0, \text{BedsUsed}(\mathcal{A}, w, d) - b_w) \tag{14}$$

where μ is the penalty weight for bed capacity violations.

3.2.4 Stage 4: Variable Neighborhood Search (VNS)

VNS systematically explores neighborhoods of increasing complexity:

$$\mathcal{N}_1 : \text{Single patient day shift } (\pm 1 \text{ or } \pm 2 \text{ days}) \tag{15}$$
$$\mathcal{N}_2 : \text{Single patient ward change} \tag{16}$$
$$\mathcal{N}_3 : \text{Multi-patient random perturbation} \tag{17}$$
$$\mathcal{N}_4 : \text{Patient swap (exchange assignments)} \tag{18}$$

Algorithm 2 Variable Neighborhood Search

```
1: $\mathcal{A}^* \leftarrow \mathcal{A}_0$                          ▷ Best solution
2: $k \leftarrow 1$                                               ▷ Neighborhood index
3: while not terminated do
4:     $\mathcal{A}' \leftarrow \text{Shake}(\mathcal{A}^*, k)$   ▷ Random perturbation in $\mathcal{N}_k$
5:     $\mathcal{A}'' \leftarrow \text{LocalSearch}(\mathcal{A}')$
6:     if $f(\mathcal{A}'') < f(\mathcal{A}^*)$ then
7:         $\mathcal{A}^* \leftarrow \mathcal{A}''$
8:         $k \leftarrow 1$                                       ▷ Reset to first neighborhood
9:     else
10:        $k \leftarrow k + 1$                                   ▷ Move to next neighborhood
11:        if $k > k_{\max}$ then $k \leftarrow 1$
12:        end if
13:    end if
14: end while
15: return $\mathcal{A}^*$
```

3.3 Matheuristic: Hybrid Approach

The matheuristic combines the strengths of metaheuristics and exact methods through a warm-start strategy.

Definition 3 (Warm-Start Integration)**.** *Given a feasible allocation $\mathcal{A}$ from the metaheuristic phase, we initialize the MILP solver by setting:*

$$y_{pwd}^{init} = \begin{cases} 1 & \text{if } \mathcal{A}[p] = (w, d) \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

The hybrid pipeline proceeds as follows:

1. **Greedy Construction** (seconds): Build initial feasible solution
2. **Local Improvement** (seconds): Refine through deterministic moves
3. **Metaheuristic** (minutes): Apply ILS or VNS for global exploration
4. **MILP with Warm-Start** (minutes): Optimize from metaheuristic solution

The warm-start provides two key advantages:

- The solver immediately has an incumbent solution, enabling pruning
- The starting point is typically close to optimal, reducing search time

3.4 Computational Complexity Analysis

Table 1: Asymptotic complexity of solution methods

Method	Time Complexity	Space Complexity
Greedy Construction	$O(\|\mathcal{P}\| \cdot \|\mathcal{W}\| \cdot \|\mathcal{D}\|)$	$O(\|\mathcal{P}\|)$
Local Search	$O(k \cdot \|\mathcal{P}\| \cdot (\|\mathcal{W}\| + \|\mathcal{D}\|))$	$O(\|\mathcal{P}\|)$
ILS/VNS per iteration	$O(\|\mathcal{P}\| \cdot \|\mathcal{W}\| \cdot \|\mathcal{D}\|)$	$O(\|\mathcal{P}\|)$
MILP (worst case)	$O(2^{\|\mathcal{P}\| \cdot \|\mathcal{W}\| \cdot \|\mathcal{D}\|})$	$O(\|\mathcal{P}\| \cdot \|\mathcal{W}\| \cdot \|\mathcal{D}\|)$

4 Computational Study

4.1 Experimental Setup

Experiments were conducted on benchmark instances generated following the methodology of Smet (2023), based on historical data from a large Belgian hospital. Instance characteristics include:

- **Wards**: 4-8 with varying bed capacities (15-30 beds)
- **Planning horizon**: 7-21 days
- **Patients**: 20-400 requiring admission
- **Specializations**: 4-8 surgical disciplines
- **Minor specializations per ward**: 0-3

All experiments were executed on an AMD Ryzen 9 5950X processor with 64 GB RAM. The MILP solver used was CBC (open-source) via the PuLP interface, with a time limit of 300 seconds per instance. Metaheuristics were configured with 5-minute time limits.

4.2 Algorithm Comparison

We compared six algorithmic variants:

1. **MILP-CBC**: Pure exact method with Branch & Bound
2. **MH-ILS**: Greedy $\rightarrow$ Local Search $\rightarrow$ ILS
3. **MH-VNS**: Greedy $\rightarrow$ Local Search $\rightarrow$ VNS
4. **MH-NoOpt-ILS**: Greedy $\rightarrow$ ILS (no local search)
5. **MH-NoOpt-VNS**: Greedy $\rightarrow$ VNS (no local search)
6. **Hybrid**: Greedy $\rightarrow$ Local Search $\rightarrow$ ILS $\rightarrow$ MILP warm-start

4.3 Results: Execution Time

The MILP-CBC solver demonstrates remarkably fast execution on the benchmark instances, with a median time of approximately 0.25 seconds. This efficiency is attributable to several factors:

- The problem structure favors Branch & Bound: tight linear relaxations enable effective pruning
- Binary variable dominance allows aggressive branching strategies
- Modern presolve techniques significantly reduce problem size

Metaheuristics exhibit consistent execution times across variants, with medians around 1.0-1.1 seconds. The Hybrid approach naturally requires more time (median 2.35 seconds) as it combines metaheuristic exploration with MILP refinement.

4.4 Results: Solution Quality

Key observations:

- **Hybrid**: Achieves optimal or near-optimal solutions in virtually all cases (median gap 0%)
- **MH-ILS**: Strong performance with median gap of 0.15%
- **MH-VNS**: Competitive with median gap of 0.45%
- **NoOpt variants**: Higher variability, with MH-NoOpt-ILS showing significant outliers

The importance of the local search phase is evident: variants without local improvement (NoOpt) exhibit substantially higher gaps and greater variance.

4.5 Pareto Front Analysis

To explore the bi-objective trade-off, we generated solutions with varying weight combinations (λ_1, λ_2) where $\lambda_1 = 1$ (fixed) and $\lambda_2 \in \{1, 10, 50, 100, 200, \ldots, 10000\}$.

The Pareto front reveals a convex trade-off structure: small improvements in workload balance require increasingly larger sacrifices in operational cost. Practitioners can use this information to select solutions that best align with their priorities.

Limitation: Due to the discrete nature of the problem (binary decision variables), the weighted-sum scalarization cannot enumerate all Pareto-optimal solutions. Some non-dominated solutions in non-convex regions of the front remain undiscovered. A complete enumeration would require alternative approaches such as the ϵ-constraint method or criterion space search algorithms.

4.6 Impact of Instance Size

Table 2: Performance characteristics by instance size

Instance Size	**MILP Time**	**MH Gap**	**Hybrid Gap**	**Recommended**
Small (20-50 patients)	5-30s	0.5-2%	0%	MILP
Medium (100-250 patients)	60-300s	3-8%	1-3%	Hybrid
Large (400+ patients)	300s+	5-12%	2-5%	Metaheuristics

5 Discussion and Conclusions

5.1 Summary of Findings

This study presents a comprehensive investigation of optimization approaches for the Hospital Patient Allocation Problem. Our key findings can be summarized as follows:

For small to medium instances (up to approximately 250 patients), classical exact methods based on Mixed Integer Linear Programming demonstrate superior performance. The combination of tight linear relaxations, effective presolve techniques, and modern Branch & Bound implementations enables rapid convergence to optimal solutions. On our benchmark instances, the MILP solver achieved median execution times below one second while guaranteeing optimality.

For large-scale instances (400+ patients), the computational burden of exact methods grows exponentially, making metaheuristics increasingly attractive. While solution quality may degrade (typical gaps of 5-12%), the wall-clock time remains manageable, enabling practical deployment in time-constrained operational settings.

The matheuristic approach effectively bridges the gap between exact and heuristic methods. By leveraging metaheuristic solutions as warm-starts for the MILP solver, this hybrid strategy inherits the rapid exploration capabilities of metaheuristics while maintaining the optimality guarantees of exact methods. The warm-start enables aggressive pruning from the first iterations, dramatically reducing the search space and accelerating convergence.

5.2 Practical Recommendations

Based on our findings, we offer the following guidelines for practitioners:

1. **Small instances**: Use MILP directly. Modern solvers handle these efficiently.
2. **Medium instances**: Consider the matheuristic approach, which provides near-optimal solutions with reasonable computation times.
3. **Large instances or real-time requirements**: Deploy metaheuristics (ILS or VNS) with local search initialization. Accept modest optimality gaps in exchange for fast execution.
4. **Pareto analysis**: For multi-objective exploration, run the MILP or Hybrid approach with varying weight combinations to approximate the Pareto front.

5.3 Limitations and Future Work

Several limitations of the current study merit acknowledgment:

- The benchmark instances, while based on real hospital data, represent a specific operational context. Results may differ for hospitals with substantially different characteristics.
- The weighted-sum scalarization cannot enumerate all Pareto-optimal solutions due to problem non-convexity.
- All problem parameters were treated as deterministic; real-world uncertainty in surgery durations and lengths of stay was not modeled.

Future research directions include:

- Stochastic programming formulations to handle uncertainty

- Multi-stage decision frameworks for rolling horizon planning
- Machine learning integration for parameter tuning and solution prediction
- Complete Pareto front generation using ϵ-constraint or criterion space methods

5.4 Conclusion

The Hospital Patient Allocation Problem presents a challenging bi-objective optimization task with significant practical implications. Our study demonstrates that no single approach dominates across all scenarios; rather, the appropriate methodology depends on instance characteristics, time constraints, and solution quality requirements. The matheuristic framework emerges as a particularly promising strategy, combining the efficiency of metaheuristics with the rigor of exact optimization. As healthcare systems continue to face increasing operational pressures, such optimization-based decision support tools offer valuable potential for improving both efficiency and working conditions in hospitals.

Acknowledgments

This work was developed as part of the Introduction to Artificial Intelligence course at the Faculty of Engineering, University of Porto. The problem instances were generated based on the methodology and data provided by Smet (2023).

References

[1] P. Smet, "Generating balanced workload allocations in hospitals," *Operations Research for Health Care*, vol. 38, 100390, 2023.

[2] N. Boland, H. Charkhgard, and M. Savelsbergh, "A criterion space search algorithm for biobjective integer programming: The balanced box method," *INFORMS Journal on Computing*, vol. 27, no. 4, pp. 735–754, 2015.

[3] 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*, vol. 48, no. 1, pp. 61–70, 2010.

[4] P. Matl, R. F. Hartl, and T. Vidal, "Workload equity in vehicle routing problems: A survey and analysis," *Transportation Science*, vol. 52, no. 2, pp. 239–260, 2018.

[5] H. R. Lourenço, O. C. Martin, and T. Stützle, "Iterated local search: Framework and applications," in *Handbook of Metaheuristics*, M. Gendreau and J.-Y. Potvin, Eds. Springer, 2010, pp. 363–397.

[6] P. Hansen, N. Mladenović, R. Todosijević, and S. Hanafi, "Variable neighborhood search: Basics and variants," *EURO Journal on Computational Optimization*, vol. 5, no. 3, pp. 423–454, 2019.

[7] W. Vancroonenburg, P. De Causmaecker, and G. Vanden Berghe, "Chance-constrained admission scheduling of elective surgical patients in a dynamic, uncertain setting," *Operations Research for Health Care*, vol. 22, 100196, 2019.