A Control-Algorithm Approach to Multi-Priority Advance Patient Scheduling

by

Gatik Gola

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Management Sciences

Waterloo, Ontario, Canada, 2026

© Gatik Gola 2026

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

We study a two-priority advance patient scheduling problem where patients of high priority incur higher waiting costs than low priority. Our study is motivated by Ontario's surgical scheduling framework, where patients of varying priority levels must be scheduled within provincially mandated wait time targets. We formulate the patient wait time minimization problem as a stochastic dynamic program, where large uncertainty in daily demand makes it intractable to solve the program exactly and build optimal scheduling policies. To mitigate these challenges, we propose a control-algorithm approach, where future demand uncertainty is replaced with certainty equivalent estimates, resulting in a deterministic program that can be solved at periodic intervals. The solutions to these programs instantiate reservation-based policy parameters that protect capacity for anticipated high-priority arrivals against consumption by low-priority demand. Solving the deterministic model periodically updates these reservations using realized remaining capacity, keeping the policy adaptive to the evolving state of the system. We validate the performance of our proposed Reservation policy using numerical experiments, examining the performance under several operational and demand parameter regimes. Our results show that the Reservation policy performs competitively against the Clairvoyant policy, which assumes perfect knowledge of future demand. Furthermore, the model significantly outperforms the Myopic policy, which represents patient scheduling without any capacity reservations for future demand.

Acknowledgements

I would like to thank my supervisors, Dr. Houra Mahmoudzadeh and Dr. Hossein Abouee-Mehrizi, for their continued support, investment in my personal growth, and encouragement throughout my graduate studies at Waterloo. I am incredibly grateful for the opportunity to learn from them and hone my skills as a researcher and a problem solver. They have helped me find my own way of thinking, and their guidance has shaped how I reason about challenges.

None of this would have been possible without the support of my friends. Chunrui, Amir, Rana, and Sean, thank you for the camaraderie, the laughter, and for making this journey so much more enjoyable than it would have been otherwise.

I am deeply grateful to my partner, Erika Groves, for her unwavering support and for believing in me, especially during the moments when I found it hardest to believe in myself. Finally, I want to thank my parents, Subhash Kumar and Rashmi, and my dear little sister, Trigya, whose sacrifices and unconditional love have been the foundation of everything I have accomplished. This work is as much theirs as it is mine.

Dedication

This thesis is dedicated to my late grandfather, Shri Tejram Bhandoria, whose memory continues to guide me.

Table of Contents

Author's Declaration ii

Abstract iii

Acknowledgements iv

Dedication v

List of Figures ix

List of Tables xi

1 Introduction 1

2 Literature Review 6

2.1 Patient Scheduling 6

2.2 Advance Scheduling 8

2.3 Certainty Equivalence Approximation and Re-Solving 9

3 Problem Definition and Formulation **11**

3.1 Decision and State Variables . . . 12

3.2 State Dynamics . . . 13

3.3 Feasible Set . . . 14

3.4 Cost function . . . 14

3.5 Stochastic Dynamic Programming (SDP) Model . . . 15

4 Control Algorithm **17**

4.1 Re-solve Period . . . 19

4.2 Updating Reservation Profile . . . 23

4.3 Interval Period . . . 24

4.3.1 Inner Program . . . 25

4.3.2 Outer Program . . . 26

4.4 Reservation Policy and Algorithm . . . 28

4.5 Illustrative Example . . . 29

5 Case Study **32**

5.1 Data Description and Experimental Setup . . . 32

5.1.1 Patient Priorities . . . 33

5.1.2 Service Capacity . . . 34

5.1.3 Service Duration . . . 35

5.1.4 Experiment Setup . . . 37

5.2 Numerical Results . . . 39

5.2.1 Impact of Operational Parameters on Waiting Costs . . . 39

5.2.2 Comparison of Patient Wait Time between Myopic and Reservation Policies 47
5.2.3 Out-of-Sample Performance Under Demand Shifts 48
5.2.4 Performance on Covariate-Based Demand Model 56

6 Conclusion **60**

References **62**

APPENDICES **67**

A Notation **68**

B Fitting Demand Distributions for High- and Low-Priority Patients **72**

List of Figures

1.1 Patient Surgical Journey in Ontario 2

4.1 Control algorithm flowchart. At each decision period, the algorithm receives the realized high- and low-priority demand and the remaining capacities as inputs, alongside certainty-equivalent demand estimates. We determine whether the current period is a re-solve period. If yes, the approximation program is solved to instantiate a new reservation profile, which is fed back into the algorithm to shape future scheduling decisions. If no, the existing reservation profile is carried forward and the current demand is scheduled against the residual capacity. 18

4.2 Re-solve procedure within the control algorithm. High-priority reservations over the future are fixed at the beginning of the solve interval and are dynamically updated throughout the interval as the demand is realized 19

4.3 Numerical Example illustrating the control algorithm 30

4.3 Numerical example illustrating the control algorithm (Continued) 31

5.1 Arrivals from 2019 to 2024 33

5.2 Patient composition and specialty distribution 34

5.3 Daily Capacity 37

5.4 Performance of all three policies for $B \in \{1, 5, 10, 20, 60, 120, 180, 260\}$ across variations of solve interval (N). The grey, green and purple lines correspond to Myopic, Clairvoyant and Reservation policies respectively. 41

5.4 Performance of all three policies for $B \in \{1, 5, 10, 20, 60, 120, 180, 260\}$ across variations of solve interval (N) (Continued). 42

5.4 Performance of all three policies for $B \in \{1, 5, 10, 20, 60, 120, 180, 260\}$ across variations of solve interval (N) (Continued). 43

5.5 Sensitivity Analysis on Capacity 45

5.6 Sensitivity Analysis on Service Time (s_p) 46

5.7 Distribution of the wait time (days) under Myopic and Reservation policies 47

5.8 Out-of-Sample Performance Under High-Priority Arrival Mean Perturbations 49

5.9 Out-of-Sample Performance Under High-Priority Arrival Variance Perturbations 51

5.10 Out-of-Sample Performance Under Low-Priority Arrival Mean Perturbations 53

5.11 Out-of-Sample Performance Under Low-Priority Arrival Variance Perturbations 55

5.12 Distribution of Absolute Cost Differences between Reservation and Clairvoyant Policies under IID and Covariate-based Negative Binomial Demand 58

B.1 Fitted demand distributions for high- and low-priority patients 73

List of Tables

5.1 Sample Master Surgical Schedule 35

5.2 Service durations 36

5.3 Parameters 39

5.4 Days within which 90% of patients received surgery 47

5.5 Negative binomial regression results for daily demand 57

5.6 Statistical Tests: IID vs. Covariate-based NB Demand Model 59

B.1 Descriptive statistics of daily surgical arrivals. 73

B.2 Poisson versus negative binomial model comparison for surgical arrival distributions. 74

Chapter 1

Introduction

In the healthcare system, the Operating Room (OR) is a highly resource-intensive facility within a hospital, acting as a primary driver of both clinical outcomes and financial sustainability (Bello et al., 2022). In response to these requirements, Ontario has implemented the Quality-Based Procedures (QBP) framework under the broader Health System Funding Reform (HSFR) (Ontario Hospital Association, 2018). Under this framework, the hospital receives funds from the Ministry of Health for a specific volume of procedures, contingent upon the facility meeting provincial access targets (Ontario Ministry of Health and Long-Term Care, 2012). The access targets represent the maximum medically recommended wait times for different priority levels (Ontario Health, 2020). In Ontario, wait times are categorized into Wait 1 (Referral to Consultation) and Wait 2 (Decision-to-Treat to Surgery), with the latter serving as the primary metric for surgical scheduling performance (Ontario Health, 2020). These targets are strictly defined by clinical urgency and the type of surgery. Priority 2 patients require intervention within 7–56 days, Priority 3 within 28–112 days, and Priority 4 within 84–182 days (Ontario Health, 2025).

To receive surgical services from a healthcare facility in Ontario, each patient progresses through a series of administrative and clinical stages within the hospital's surgical workflow. The typical sequence of events is summarized in Figure 1.1, and the details of the important dates in the timeline are outlined below.

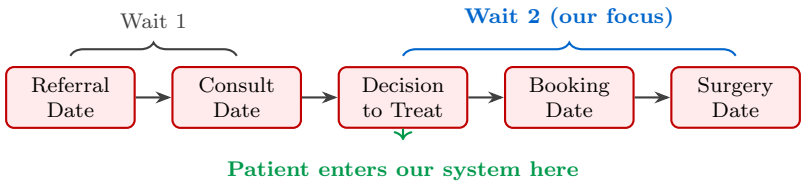


Figure 1.1: Patient Surgical Journey in Ontario

- *Referral date:* Patients are referred by a primary care physician or another specialist to a surgeon or surgical service for an initial assessment. This marks the point at which the patient's information is first entered into the hospital's surgical records.
- *Consult date:* Patients attend a consultation with the surgeon, during which a comprehensive clinical evaluation is conducted to determine the necessity of a surgical intervention.
- *Decision-to-treat date:* Based on the clinical assessment, the surgeon decides whether a surgery is required. The patient is also assigned a priority level that reflects the urgency of the procedure. Clinicians use PAT (Priority Access Tool) as a guide to set priority levels for their patients based on their symptoms, diagnosis and other clinical factors (see Ontario Health (2020) for more details). As the patient gets approved for the surgery on this date, we assume decision-to-treat date as the patient's arrival day into the system for surgical services.
- *Booking date:* The patient is scheduled for surgery by the hospital provider. The booking decision is made in accordance with the assigned priority level and the availability of surgical capacity.
- *Surgery date:* The surgical procedure is performed by the surgeon on the booked date.

To better manage the limited OR capacities and meet the provincial volume quotas and clinical access targets, the hospital builds a Master Surgical Schedule (MSS) which acts

as a budgetary tool for each specialty throughout the planning horizon. Within an MSS, each surgical specialty is allocated total capacity based on the specific volume targets set by the province for that fiscal year. While the design of an MSS is a critical strategic task, it is not the primary focus of this thesis. Instead, we treat the MSS as a fixed structural constraint that establishes the baseline capacities for each specialty. Our core challenge is to schedule patients of different priority levels in advance while minimizing total weighted waiting costs, where high-priority patients incur a larger per-period waiting cost for each day they remain unscheduled. As high-priority patients carry a larger penalty than low-priority patients, it becomes necessary to allocate capacity judiciously to serve them faster. At the same time, low-priority patients cannot be deferred indefinitely, as excessive delay risks idle capacity. To this end, an advance scheduling system becomes a practical necessity to curb the degradation of patients' quality of life, increasing wait lists and growing wait times within a resource-constrained environment (Wiebe et al., 2022). The advance surgical scheduling challenge is not unique to Ontario; Barrera et al. (2020) document a similar problem arising from Chile's GES health reform, which established maximum waiting times between diagnosis and treatment for covered pathologies. However, instead of receiving funding clawbacks for not meeting the targets by the end of fiscal year, Chile's healthcare system enforces financial penalties on the provider for each surgery violating the access target.

Since patients arrive randomly over time and scheduling decisions must be made sequentially under future demand uncertainty, the problem has a natural dynamic structure. We therefore formulate it as a stochastic dynamic program; however, due to the curse of dimensionality, solving such programs exactly to produce optimal scheduling policies remains computationally intractable (Bertsekas, 2017; Patrick et al., 2008; Truong, 2015). Rather than optimizing over all possible policies, we build a control algorithm that optimizes over a specific subset of control policies based on protection levels, which govern how much future surgical capacity is held back for high-priority patients. We refer to these protection levels as *reservations*, and to the resulting policy class as the Reservation policy. This policy class is motivated by the observation that high-priority patients incur the greatest waiting costs, and these reservations allow them to be scheduled closer to their

arrival date, reducing total weighted waiting costs across the system. We compute these reservations by solving an approximation program that employs the *certainty equivalence principle* to replace future arrival uncertainty with expected values (Bertsekas, 2017). The approximation program is solved periodically such that these reservation plans are updated to reflect the realized system state, allowing the policy to adapt over time. We evaluate the algorithm in a case study and compare the performance of our policy against two other benchmarks.

Our main contributions and results can be summarized in three parts. First, we develop a control algorithm for advance patient scheduling under uncertain demand. Second, we conduct a sensitivity analysis on operational and demand parameters to identify key system-stress drivers and set baseline parameters. Third, we quantify the impact of the Reservation policy on patient wait times relative to two benchmark policies: the Myopic policy, which schedules greedily without any look-ahead, and the Clairvoyant policy, which assumes perfect knowledge of future demand.

- *Control algorithm*: We develop a control algorithm as an approximate solution method for an advance patient scheduling problem in which each patient priority level is subject to distinct access targets. The algorithm achieves tractability through two components: an approximation program and a control policy class. At certain decision periods, the approximation program is solved optimally with integer decisions to determine how much future capacity to reserve for high-priority patients. This reservation mechanism acts as the control policy class of interest.

- *Parameter Sensitivity*: We study the impact of different values of operational parameters and compare the performance of the Reservation policy against two other benchmarks. We observe that the Reservation policy performs the best when the approximation program is solved in every decision period. With such insights, we appropriately set the baseline operational parameters to test the out-of-sample performance of the policy under different demand regimes. The low-priority arrival mean was observed to be the dominant system-stress driver, where the Reservation policy performed favorably compared to the Myopic policy.

- *Impact on Patient Wait Time*: The Reservation policy reduces the 90^{th} percentile wait time for high-priority patients by 50% relative to the Myopic policy, while incurring only an 11% increase in low-priority wait times, demonstrating a favorable trade-off in patient access to care.

This thesis is structured as follows. In Chapter 2, we provide a literature review. In Chapter 3, we present the mathematical formulation of the problem, including the state dynamics and feasible action set. In Chapter 4, we propose our control algorithm. In Chapter 5, we conduct a case study using real arrival data from a local hospital. Finally, in Chapter 6, we conclude with a summary of findings and directions for future work.

Chapter 2

Literature Review

In this chapter, we review three relevant streams of literature, i.e., patient scheduling, advance scheduling, and certainty equivalence control.

2.1 Patient Scheduling

Managing capacity within a tight, resource-constrained environment is a problem that concerns a lot of healthcare professionals. The patient arrival process in such systems is highly variable, and the resource capacity is fixed. Such circumstances warrant the development of scheduling systems that guarantees an efficient delivery of surgical services such that the patient's health does not deteriorate and certain predefined metrics are met. These metrics vary based on the specific problems faced by the institution. For example, some hospitals experience high over/under utilization of the operating room, or they face a high risk of surgery cancellations due to patient factors or operational reasons. For more information on such problems faced by healthcare operations managers, we direct the readers to surveys Cardoen et al. (2010) and Van Riet and Demeulemeester (2015).

The OR planning problem can be split into three levels: strategic, tactical, and operational (Abdalkareem et al., 2021; Zhu et al., 2019). The strategic and tactical levels

together solve a decision problem of planning and allocating capacity. The decisions in the strategic level include the amount of time each surgical specialty gets, so the output is essentially a budget. Whereas, the decisions in the tactical level include the exact time and duration of the blocks allocated to the specialty based on the strategic allocation above. For this reason, the development of a Master Surgical Schedule falls in the tactical level. The operational level follows the tactical level decisions, where the decision makers are concerned with the patient scheduling problem. Once the exact capacity split for each specialty is known from the upper levels, the operational level solutions provide a structured method to fill the provided capacity. According to Magerlein and Martin (1978), there are two kinds of patient scheduling problems, *allocation* scheduling and *advance* scheduling. Allocation scheduling requires the provider to schedule the patients on the same day while queuing the rest in a wait list that is maintained throughout the planning horizon, whereas in advance scheduling, the provider is allowed to schedule the patients into future days. There exists a line of work where the tactical and operational levels are both considered together. Aringhieri et al. (2015), Tànfani and Testi (2010), and Testi et al. (2007) have studied the joint problem of designing the master surgical schedule and scheduling patients from a wait list into a future day from the MSS.

There has been a considerable amount of research performed on allocation scheduling. One of the earliest works came from Gerchak et al. (1996), where they leveraged dynamic programming to build an optimal policy for two priority classes with explicit capacity reservations for emergency patients. Their model acted as a foundation for Min and Yih (2010) and Ayvaz and Huh (2010), where the former proposed an optimal policy based on the number of patients of each priority in the waiting list and the latter developed a threshold heuristic policy for patients of different priority classes. While some have been able to characterize an optimal allocation scheduling policy using stochastic DP for two priority levels, Wang and Truong (2018) were the first to provide performance guarantees for the general multiclass allocation scheduling problem using the tools of competitive analysis.

In this thesis, we draw our attention to multi-priority advance patient scheduling literature.

2.2 Advance Scheduling

Patrick et al. (2008) consider a dynamic multi-priority advance scheduling problem where a patient is scheduled in a future time slot on the day of arrival. They propose an MDP model and use ADP techniques to produce scheduling policies that provide guarantees for certain combination of waiting costs, overtime costs and patient diversion costs. Sauré et al. (2020) build on their framework to study the problem with both deterministic and stochastic service durations. For the deterministic case, they provide a closed form solution in terms of overtime costs and service time means. Astaraky and Patrick (2015) consider advance scheduling of patients of multiple classes while minimizing patient wait time and optimizing recovery bed utilization. Gocgun and Puterman (2014) study the advance patient scheduling problem under chemotherapy scheduling setting and use LP-based ADP techniques to solve the problem. Gocgun (2018) studies the same problem with cancellation under radiotherapy setting and use simulation-based ADP methods to solve the problem. While the lineage of dynamic advance patient scheduling originating from Patrick et al. (2008) provides good approximation policies for a wide range of scheduling applications, Truong (2015) characterize an optimal advance scheduling policy for the case of two classes (elective and emergency) by connecting advance scheduling with allocation scheduling. More specifically, they propose the concept of successive refinability, using which they show that optimal advance schedule produces same objective value as optimal allocation schedule. However, their results only hold true if the emergent patient gets scheduled on the same day of arrival, which can be restrictive in contexts where classes can hold different wait time windows. Zacharias et al. (2024) use Truong's optimal scheduling model as a building block to add intraday scheduling feature, allowing them to make an optimal decision for both the appointment date and time for the patients.

Rolling horizon control has emerged as a practical mechanism for implementing complex optimization programs developed using MDP or multi-stage stochastic programming. For example, Andersen et al. (2020) formulate daily OR scheduling as a finite-horizon MDP and use a simulation-based rollout algorithm to derive scheduling decisions at each decision period without solving the full MDP exactly. Keyvanshokooh et al. (2022) similarly

propose a data-driven rolling horizon procedure, where they re-solve the stochastic program over a rolling window at each planning epoch to incorporate updated patient arrival scenarios while avoiding the full computational burden of the multi-stage program. For more applications of rolling horizon control in general multi-period problems, we direct the readers to Glomb et al. (2022).

There is an alternative paradigm that models advance patient scheduling problem as an online problem. A problem is *online* if, in each period, one is only allowed to look at information available till the current period. Stein et al. (2020) propose an online algorithm for an advance reservation system with non-stationary arrival process and derive a competitive ratio. Wang et al. (2018) propose a competitive online algorithm for advance scheduling problem with resource perishability. Keyvanshokooh et al. (2021) derive competitive ratio for an online primal-dual algorithm that schedules patients of multiple classes in resources where extra capacity can be accessed at additional overtime costs.

2.3 Certainty Equivalence Approximation and Re-Solving

Certainty Equivalence is an approximation strategy in which all the random variables are replaced with the expected values (Bertsekas, 2017). This transforms the problem into a deterministic optimization problem that acts as an approximation method to the original stochastic problem (Bertsekas, 2017). This approach turns out to be optimal for very specific problems under certain conditions. For these specific problems, refer to Simon (1956), Theil (1957), and Chuang et al. (2014).

Certainty-equivalence principles have been widely applied in the field of network revenue management (NRM), which encompasses large-scale resource allocation problems with uncertain, multi-class demand. Random demand variables are replaced with their expected values and fed into a Deterministic Linear Program (DLP), which serves as a computationally tractable model for studying the marginal value of resources and developing pricing policies (Talluri and van Ryzin, 1998). Even though the plan built using DLP informs

decent decision rules, the quality of the decisions can drift as the plan does not adapt to realized demand. To combat the problem, the DLP can be solved periodically to build new decision rules on updated remaining capacities. Cooper (2002) pointed out that periodic re-optimization does not necessarily provide direct benefit in all problems. Secomandi (2008) delivered further insight into this issue by providing structural conditions under which re-solving helps. They prove that, under certain conditions, the value of computing new policy parameters via daily re-optimization is weakly higher than that of following the policy computed from a one-shot solve at the beginning of the horizon. Jasin and Kumar (2012) quantify the benefit of frequent re-solving, where under the non-degeneracy assumption of the DLP, they claim that the revenue loss relative to the optimal dynamic policy is $\mathcal{O}(1)$. Bumpensanti and Wang (2020) further lift the assumption of non-degeneracy to show that the uniform bounded loss is achievable, provided re-solving occurs at carefully selected decision periods.

Our advance surgical scheduling problem is an integer decision problem. Fractional patient allocations computed from continuous programs like a DLP are physically meaningless in a clinical setting. There exist LP-based control methods for making integer allocation decisions in resource-constrained settings under stochastic demand. Notably, Reiman and Wang (2008), Jasin and Kumar (2012), Ciocan and Farias (2012) use Probabilistic Allocation Control (PAC) to interpret the LP solutions as per-arrival acceptance probabilities to make integer decisions. However, these methods are designed for a fundamentally different decision structure. They operate on a *time-aggregate* formulation in which, at the time of re-solve, the stochastic demand is aggregated into a single scalar representing total expected arrivals over the remaining horizon (Secomandi, 2008). One could in principle solve the LP relaxation and apply a rounding procedure to obtain integer reservation quantities. However, as total capacity is partitioned into specialty-specific blocks with disjoint capacity constraints, solving the integer program at each re-solve period becomes computationally tractable. This means that we can afford to solve integer programs for each specialty independently to build future reservations with optimality. In this thesis, we do not provide any performance guarantees for our proposed method, but we validate the performance of our algorithm through extensive numerical experiments.

Chapter 3

Problem Definition and Formulation

Consider an appointment service system responsible for scheduling two patient priority classes over a *finite planning horizon*

$$[T] = \{\, t \in \mathbb{Z} : 1 \leq t \leq T \,\}.$$

Patients arrive sequentially over time and are assigned a surgery date on/after their arrival day. At each decision period, the provider attempts to schedule patients within a finite window of B days, which we refer to as the *booking horizon*. Specifically, a patient considered for scheduling on day $t \in [T]$ can be scheduled on any day within day t and day $t + B - 1$. If no capacity is available within this window, the patient is deferred to a queue and reconsidered for scheduling in the following period. We assume that each patient is assigned a surgery start and end time within the chosen day. Each patient holds a priority class that denotes how severe their medical condition is. We define the priority set as $P = \{1, 2\}$, where $p = 1$ corresponds to high priority and $p = 2$ to low priority. The purpose of these priorities is to help the provider decide how quickly the patient should receive the surgical treatment. We assume that each day $t \in [T]$ has fixed, finite capacity, that is available and known to the provider at the beginning of the planning horizon. We further assume deterministic surgery durations for all priority classes and denote the service time for a patient of priority p by s_p.

At the beginning of each period t, the provider observes the arrivals for each priority class and schedules them within the booking horizon. We denote the realized number of priority p patients arriving on day t by $D_t^{(p)}$. We assume that the provider has no information about the arrivals in future periods; however, they can be estimated using historical data and arrivals up to the current period. We denote the certainty equivalent estimate of priority p patients arriving on day t by $\hat{D}_t^{(p)}$. With all the important parameters defined, we are now ready to provide a mathematical formulation of the scheduling system.

3.1 Decision and State Variables

Let $\mathcal{I}_t = \{(t_1, t_2) \in \mathbb{Z}^2 : 1 \leq t_1 \leq t, t \leq t_2 \leq T\}$ be the index set of all valid arrival day and surgery appointment day pairs at period t, ordered lexicographically. The decision variable in period $t \in [T]$ is defined as

$$\mathbf{x}(t) := \left(\mathbf{x}^{(1)}(t), \mathbf{x}^{(2)}(t)\right) \in X_t, \tag{3.1}$$

where $\mathbf{x}(t) \in X_t \subseteq \mathbb{Z}_{\geq 0}^{2\cdot|\mathcal{I}_t|}$. $\mathbf{x}^{(1)}(t)$ and $\mathbf{x}^{(2)}(t)$ are vectors of high-priority and low-priority decisions such that

$$\mathbf{x}^{(p)}(t) := \left(x_{t_1,t_2}^{(p)}(t)\right)_{(t_1,t_2)\in\mathcal{I}_t}. \tag{3.2}$$

In the definition above, $x_{t_1,t_2}^{(p)}(t)$ represents the number of priority p patients arriving on day t_1 whose surgeries are scheduled for day t_2 based on the decisions made in period t.

The system state at the beginning of period t consists of three components:

$$\begin{aligned} \mathbf{c}(t) &:= \left(c_{t_2}(t)\right)_{t_2\in[T]} \\ \mathbf{q}^{(1)}(t) &:= \left(q_{t_1}^{(1)}(t)\right)_{t_1\in\{1,\ldots,t-1\}} \\ \mathbf{q}^{(2)}(t) &:= \left(q_{t_1}^{(2)}(t)\right)_{t_1\in\{1,\ldots,t-1\}} \end{aligned} \tag{3.3}$$

where $\mathbf{c}(t) \in \mathcal{C}_t \subseteq \mathbb{R}^T_{\geq 0}$, $\mathbf{q}^{(1)}(t) \in \mathcal{Q}^{(1)}_t \subseteq \mathbb{Z}^{t-1}_{\geq 0}$, and $\mathbf{q}^{(2)}(t) \in \mathcal{Q}^{(2)}_t \subseteq \mathbb{Z}^{t-1}_{\geq 0}$. Here $c_{t_2}(t)$ is the available service capacity on day t_2 as observed at the beginning of period t, and $q^{(p)}_{t_1}(t)$ is the number of unscheduled priority-p patients who arrived on day t_1 and remain waiting at the beginning of period t. We assume that the capacity available for all days $t_2 \in [T]$ at the beginning of the planning horizon equals the maximum MSS-specified capacity and the queue is initially empty.

3.2 State Dynamics

The evolution of the system is governed by the state transition dynamics of the capacity and queue states. The element-wise dynamics of the capacity state are defined as

$$\begin{aligned} c_{t_2}(t+1) &= \Big(c_{t_2}(t) - \sum_{p\in P}\sum_{t_1=1}^{t} s_p\, x^{(p)}_{t_1,t_2}(t)\Big)^+, \quad t \leq t_2 \leq \min(t+B-1, T) \\ c_{t_2}(t+1) &= c_{t_2}(t), \qquad\qquad\qquad\qquad\qquad\qquad\quad t_2 > \min(t+B-1, T). \end{aligned} \tag{3.4}$$

The dynamics above track the available capacity, in the next decision period, on day t_2 due to scheduling decisions $x^{(p)}_{t_1,t_2}(t)$ made in the current decision period t. The capacity state for days outside the booking horizon gets carried over to the next decision period unaffected. $y^+ = \max(y, 0)$ ensures that the capacity does not become negative.

The element-wise dynamics of the queue states are defined as

$$\begin{aligned} q^{(p)}_{t_1}(t+1) &= \Big(q^{(p)}_{t_1}(t) - \sum_{t_2=t}^{\min(t+B-1,T)} x^{(p)}_{t_1,t_2}(t)\Big)^+, \quad t_1 < t,\ p \in P \\ q^{(p)}_{t_1}(t+1) &= \Big(D^{(p)}_{t_1} - \sum_{t_2=t}^{\min(t+B-1,T)} x^{(p)}_{t_1,t_2}(t)\Big)^+, \quad\ \ t_1 = t,\ p \in P. \end{aligned} \tag{3.5}$$

The first case tracks the residual backlog for patients who arrived on day $t_1 < t$ and remain unscheduled, whereas the second case initializes the queue entry for patients arriving on

the current day $t_1 = t$. Again, $y^+ = \max(y, 0)$ ensures that the queue does not become negative.

3.3 Feasible Set

At each stage t, the provider chooses a scheduling decision $\mathbf{x}(t)$ from the feasible action set under state $\big(\mathbf{c}(t),\, \mathbf{q}^{(1)}(t),\, \mathbf{q}^{(2)}(t)\big)$. The feasible set enforces the following conditions: capacity on each surgery day is not exceeded, the number of patients scheduled from any arrival cohort does not exceed those waiting in the queue, and the state transition equations are satisfied. The capacity constraint is defined as

$$\sum_{p \in P} \sum_{t_1=1}^{t} s_p \, x_{t_1,t_2}^{(p)}(t) \le c_{t_2}(t), \quad \forall\, t_2 \in [t, \min(t + B - 1, T)], \tag{3.6}$$

and the queue constraints are defined as

$$\begin{aligned} \sum_{t_2=t}^{\min(t+B-1,T)} x_{t_1,t_2}^{(p)}(t) &\le q_{t_1}^{(p)}(t), \quad \forall\, t_1 < t,\ p \in P, \\ \sum_{t_2=t}^{\min(t+B-1,T)} x_{t_1,t_2}^{(p)}(t) &\le D_{t_1}^{(p)}, \quad t_1 = t,\ p \in P, \end{aligned} \tag{3.7}$$

The feasible action set at period t can then be compactly written as

$$\mathcal{Y}_t\big(\mathbf{c}(t),\, \mathbf{q}^{(1)}(t),\, \mathbf{q}^{(2)}(t)\big) = \big\{\mathbf{x}(t) \mid (3.4),\, (3.5),\, (3.6),\, (3.7)\big\}. \tag{3.8}$$

3.4 Cost function

The goal is to minimize the total weighted patient wait time over the planning period. By assuming that the system incurs waiting cost w_p per period for each patient of priority p

the instantaneous scheduling cost for all the arrivals of both priorities till period t can be written as

$$f_t(\mathbf{x}(t)) = \sum_{p\in P}\sum_{t_1=1}^{t}\sum_{t_2=t}^{\min(t+B-1,T)} (t_2 - t_1 + 1) w_p x^{(p)}_{t_1,t_2}(t). \tag{3.9}$$

As a patient who remains unscheduled at the beginning of period $T+1$ has waited $T - t_1 + 1$ periods, the terminal cost can be written as

$$g_{T+1}(\mathbf{q}^{(1)}(T+1),\ \mathbf{q}^{(2)}(T+1)) = \sum_{p\in P}\sum_{t_1=1}^{T}(T - t_1 + 1)\, w_p\, q^{(p)}_{t_1}(T+1). \tag{3.10}$$

3.5 Stochastic Dynamic Programming (SDP) Model

Following the standard dynamic program formulation techniques from (Bertsekas, 2017), we build an SDP for our problem. We look for an *admissible* policy or a control law $\pi = \{\mu_t\}_{t=1}^{T}$, that is a sequence of decision functions, where each μ_t maps the current system state to a feasible scheduling decision:

$$\mu_t \ :\ \big(\mathcal{C}_t, \mathcal{Q}^{(1)}_t,\ \mathcal{Q}^{(2)}_t\big) \ \longrightarrow\ \mathcal{Y}_t\big(\mathbf{c}(t),\, \mathbf{q}^{(1)}(t),\, \mathbf{q}^{(2)}(t)\big), \qquad \mathbf{x}(t) \ =\ \mu_t\big(\mathbf{c}(t),\, \mathbf{q}^{(1)}(t),\, \mathbf{q}^{(2)}(t)\big). \tag{3.11}$$

Let Π denote the set of such admissible policies. An optimal policy $\pi^* \in \Pi$ can be found such that

$$\begin{aligned}\mu^*_t(\mathbf{c}(t),\, \mathbf{q}^{(1)}(t),\, \mathbf{q}^{(2)}(t)) \ =\ \underset{\mathbf{x}(t)\,\in\,\mathcal{Y}_t\left(\mathbf{c}(t),\,\mathbf{q}^{(1)}(t),\,\mathbf{q}^{(2)}(t)\right)}{\arg\min} \Big[\ & f_t\big(\mathbf{x}(t)\big) + \\ & \mathbb{E}\big[J_{t+1}\big(\mathbf{c}(t+1),\, \mathbf{q}^{(1)}(t+1),\, \mathbf{q}^{(2)}(t+1)\big)\ \Big|\ \big(\mathbf{c}(t),\, \mathbf{q}^{(1)}(t),\, \mathbf{q}^{(2)}(t)\big)\big]\Big],\end{aligned} \tag{3.12}$$

where the expectation is conditional on the current state $(\mathbf{c}(t), \mathbf{q}^{(1)}(t), \mathbf{q}^{(2)}(t))$. The cost-

to-go function satisfies the recursion

$$J_t(\mathbf{c}(t), \mathbf{q}^{(1)}(t), \mathbf{q}^{(2)}(t)) = \min_{\mathbf{x}(t) \in \mathcal{Y}_t\left(\mathbf{c}(t), \mathbf{q}^{(1)}(t), \mathbf{q}^{(2)}(t)\right)} \Big[f_t\big(\mathbf{x}(t)\big) + \mathbb{E}\big[J_{t+1}\big(\mathbf{c}(t+1), \mathbf{q}^{(1)}(t+1), \mathbf{q}^{(2)}(t+1)\big) \mid \big(\mathbf{c}(t), \mathbf{q}^{(1)}(t), \mathbf{q}^{(2)}(t)\big)\big] \Big],$$

with terminal condition

$$J_{T+1}\big(\mathbf{c}(T+1), \mathbf{q}^{(1)}(T+1), \mathbf{q}^{(2)}(T+1)\big) = g_{T+1}\big(\mathbf{q}^{(1)}(T+1), \mathbf{q}^{(2)}(T+1)\big).$$

While the Bellman recursion exactly characterizes the optimal policy, it is computationally intractable at realistic planning horizons. The state space suffers from the curse of dimensionality, that is, the capacity vector and queue vectors grow in dimension with each passing period and minimizing over such growing state space is not a trivial task. This intractability motivates the use of approximation methods. In the next chapter, we develop a control algorithm that replaces future demand uncertainty with certainty-equivalent estimates and solves a deterministic program periodically to generate a scheduling policy.

Chapter 4

Control Algorithm

In this chapter, motivated by (Secomandi, 2008), we build a control algorithm and solve approximation programs periodically to produce optimal solutions that define the decision rules for scheduling patients within the planning horizon. These decision rules belong to a special control policy class which comprises of reservations for future high-priority patients. The approximation program is periodically re-solved to produce new Reservation policy parameters, which shape future scheduling decisions till the next re-solve. Figure 4.1 provides a brief overview of the procedure.

To introduce the periodic re-solve mechanism into the system, we define the *solve interval* (N) parameter as the number of periods elapsed between successive re-solves. Setting $N = 1$ means the approximation program re-solves to build new reservations every period. When $N > 1$, the reservation profile from earlier re-solve simply carries forward with necessary updates (more on this in Section 4.2). It is important to note that the reservations developed at each re-solve period is not a fixed schedule for high priority patients, that is, we do not fill the reserved capacity with high-priority patients. The reservations act as a pure protection mechanism to deter the low priority patients from consuming all the available capacity in the system. Thereafter, these reservations are subject to revision at the next re-solve period.

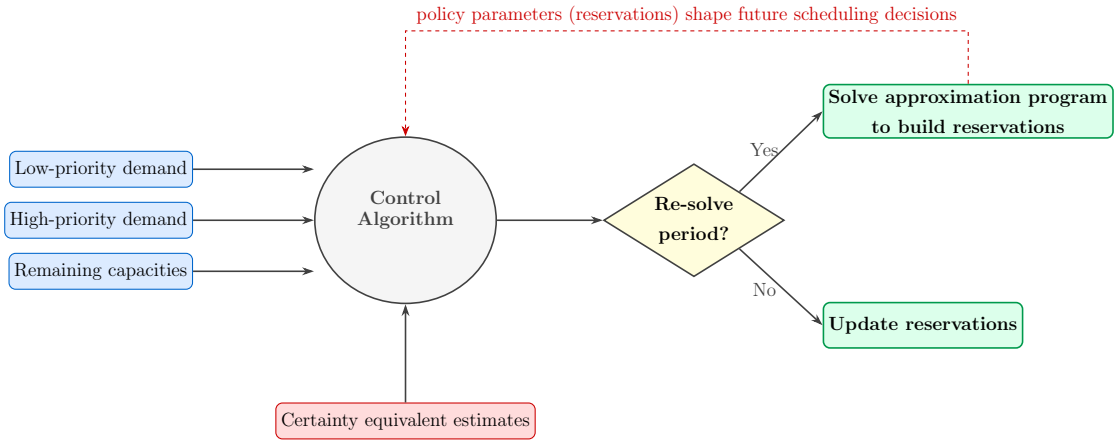


Figure 4.1: Control algorithm flowchart. At each decision period, the algorithm receives the realized high- and low-priority demand and the remaining capacities as inputs, alongside certainty-equivalent demand estimates. We determine whether the current period is a re-solve period. If yes, the approximation program is solved to instantiate a new reservation profile, which is fed back into the algorithm to shape future scheduling decisions. If no, the existing reservation profile is carried forward and the current demand is scheduled against the residual capacity.

There is a clear distinction between two types of periods within the planning horizon: re-solve periods, where the approximation programs are solved to produce policy parameters, and interval periods, where the reservations shape the scheduling decisions between two successive re-solves. We denote the set of re-solve periods as $\mathcal{K} = \{1, N+1, 2N+1, \ldots\}$ and the set of interval periods associated with the preceding re-solve period $k \in \mathcal{K}$ as $\mathcal{T}_k = \{k+1, \ldots, k+N-1\}$. In each decision period, we solve a math program to schedule the realized arrivals. We denote the program solved in re-solve period k by $\mathcal{RP}_k$ and it produces a high-priority reservation profile over the future periods. Whereas, the program solved in the interval period τ is denoted by $\mathcal{IP}_\tau$, and it uses the high-priority reservations to schedule in the interval periods. Figure 4.2 provides a high-level view of the re-solve procedure.

The system dynamics for the re-solve periods $\mathcal{K}$ and the interval periods $\mathcal{T}_k$ are structurally different, so we present them separately. We begin with presenting the optimization at re-solve period k in Section 4.1; we present an update mechanism for the reservations

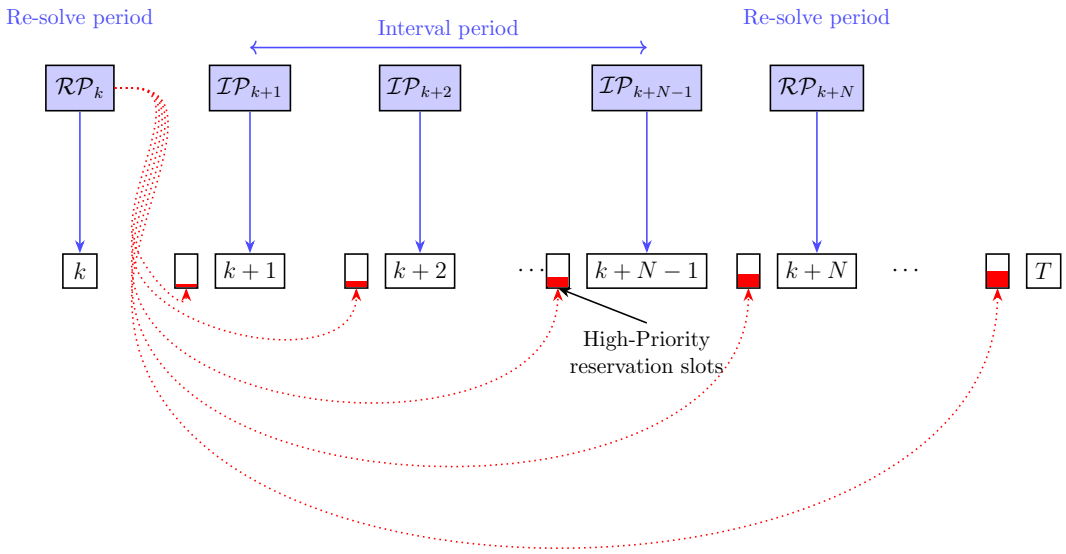


Figure 4.2: Re-solve procedure within the control algorithm. High-priority reservations over the future are fixed at the beginning of the solve interval and are dynamically updated throughout the interval as the demand is realized

over the interval periods in Section 4.2; we finally elaborate on the optimization in interval periods in Section 4.3.

4.1 Re-solve Period

We employ the certainty equivalence principle to replace uncertain demand with the expected values and formulate the approximation program $\mathcal{RP}_k$ as a finite-horizon optimization program with the same constraints and objective as the original formulation. Throughout this section, we use $\tilde{(\cdot)}$ to denote a quantity that evolves under the certainty-equivalent approximation. This helps us differentiate the state and decision variables of the approximation program from the true stochastic system. Note that $\big(\tilde{\mathbf{c}}(k),\, \tilde{\mathbf{q}}^{(1)}(k),\, \tilde{\mathbf{q}}^{(2)}(k)\big) = \big(\mathbf{c}(k),\, \mathbf{q}^{(1)}(k),\, \mathbf{q}^{(2)}(k)\big)$ and $\tilde{\mathbf{x}}(k) = \mathbf{x}(k)$ as we make decisions for the current period de-

mand and the future expected demand at the same time.

The element-wise dynamics of the capacity state between two consecutive periods are defined as

$$\begin{aligned}
\tilde{c}_{t_2}(t+1) &= \Big(\tilde{c}_{t_2}(t) - \sum_{p\in P}\sum_{t_1=1}^{t} s_p\, \tilde{x}^{(p)}_{t_1,t_2}(t)\Big)^+, \quad t \le t_2 \le \min(t+B-1, T), \\
\tilde{c}_{t_2}(t+1) &= \tilde{c}_{t_2}(t), \qquad\qquad\qquad\qquad\qquad\qquad t_2 > \min(t+B-1, T).
\end{aligned} \tag{4.1}$$

The element-wise dynamics of the queue states between two consecutive periods are defined as

$$\begin{aligned}
\tilde{q}^{(p)}_{t_1}(t+1) &= \Big(\tilde{q}^{(p)}_{t_1}(t) - \sum_{t_2=t}^{\min(t+B-1,T)} \tilde{x}^{(p)}_{t_1,t_2}(t)\Big)^+, \qquad t_1 < t,\ p \in P, \\
\tilde{q}^{(p)}_{t_1}(t+1) &= \Big(D^{(p)}_{t_1} - \sum_{t_2=t}^{\min(t+B-1,T)} \tilde{x}^{(p)}_{t_1,t_2}(t)\Big)^+, \quad t = t_1 = k,\ p \in P, \\
\tilde{q}^{(p)}_{t_1}(t+1) &= \Big(\hat{D}^{(p)}_{t_1} - \sum_{t_2=t_1}^{\min(t_1+B-1,T)} \tilde{x}^{(p)}_{t_1,t_2}(t)\Big)^+, \quad t = t_1 > k,\ p \in P.
\end{aligned} \tag{4.2}$$

The first case tracks the residual backlog for patients who arrived before the current re-solve period $t_1 < k$. The second case initializes the queue entry for patients arriving in the current period. The third case initializes queue state for future arrivals using the certainty equivalent estimates.

The capacity constraint for the re-solve program is defined as

$$\sum_{p\in P}\sum_{t_1=1}^{t} s_p\, \tilde{x}^{(p)}_{t_1,t_2}(t) \le \tilde{c}_{t_2}(t), \quad \forall\, t_2 \in [t, \min(t+B-1, T)], \tag{4.3}$$

and the queue constraints are defined as

$$
\begin{aligned}
&\sum_{t_2=t}^{\min(t+B-1,T)} \tilde{x}^{(p)}_{t_1,t_2}(t) \le \tilde{q}^{(p)}_{t_1}(t), \quad \forall\, t_1 < k,\ p \in P, \\
&\sum_{t_2=t}^{\min(t+B-1,T)} \tilde{x}^{(p)}_{t_1,t_2}(t) \le D^{(p)}_{t_1}, \quad t_1 = k,\ p \in P, \\
&\sum_{t_2=t_1}^{\min(t_1+B-1,T)} \tilde{x}^{(p)}_{t_1,t_2}(t) \le \hat{D}^{(p)}_{t_1}, \quad t_1 > k,\ p \in P.
\end{aligned} \tag{4.4}
$$

The feasible action set for the re-solve program at period t can then be compactly written as

$$
\tilde{\mathcal{Y}}_t\big(\tilde{\mathbf{c}}(t),\, \tilde{\mathbf{q}}^{(1)}(t),\, \tilde{\mathbf{q}}^{(2)}(t)\big) = \big\{\tilde{\mathbf{x}}(t) \mid (4.1),\, (4.2),\, (4.3),\, (4.4)\big\}. \tag{4.5}
$$

The math program at re-solve period k seeks to determine the optimal deterministic schedule over the future stages based on the future expected demand. The problem can then be formulated as:

$$
\begin{aligned}
\mathcal{RP}_k \;:=\; \min_{\tilde{\mathbf{x}}} \quad & \sum_{t=k}^{T} f_t(\tilde{\mathbf{x}}(t)) + g_{T+1}\big(\tilde{\mathbf{q}}^{(\mathbf{1})}(T+1),\, \tilde{\mathbf{q}}^{(2)}(T+1)\big) && \text{(4.6a)} \\
\text{s.t.} \quad & && \text{(4.6b)} \\
& \big(\tilde{\mathbf{c}}(k),\, \tilde{\mathbf{q}}^{(1)}(k),\, \tilde{\mathbf{q}}^{(2)}(k)\big) = \big(\mathbf{c}(k),\, \mathbf{q}^{(1)}(k),\, \mathbf{q}^{(2)}(k)\big) && \text{(4.6c)} \\
& \tilde{\mathbf{x}}(t) \in \tilde{\mathcal{Y}}_t\big(\tilde{\mathbf{c}}(t),\, \tilde{\mathbf{q}}^{(1)}(t),\, \tilde{\mathbf{q}}^{(2)}(t)\big) \qquad \forall t \in \{k, \ldots, T\} && \text{(4.6d)}
\end{aligned}
$$

The program above is a deterministic program where all the data is known in advance. Specifically, we solve the math program with the knowledge of realized demand till the current period and the deterministic estimate of the future arrivals. In the classical scheduling literature, when all jobs are available at the solve time, the optimal sequence for minimizing total weighted cost can be found by ordering the patients in decreasing order of $\frac{w_p}{s_p}$ and scheduling them greedily (Pinedo, 2022). In our setting, since we assume equal service

durations across priority classes, this reduces to ordering patients by priority weight alone. However, one might expect that a low-priority patient who arrives sufficiently early should, at some point, be scheduled before a later-arriving high-priority patient. In Proposition 1, we show that this intuition is incorrect. Even though the arrivals occur at different time periods, it is optimal to schedule all the high-priority arrivals over the horizon before the low-priority arrivals. This insight can be used to solve a smaller program, with future estimates of only high-priority arrivals.

Proposition 1. *It is optimal to schedule all high-priority patients before all low-priority patients, regardless of the relative order of their arrival days.*

Proof. We use a pairwise interchange argument. Suppose there exists an optimal schedule $\mathcal{S}$ in which a low-priority patient i is scheduled before a high-priority patient j, with arrival days t_1^i and t_1^j and assigned surgery days $t_2^i \leq t_2^j$ respectively. If $t_2^i = t_2^j$, both patients are served on the same day and no priority ordering violation exists; the interchange is trivially unnecessary. We therefore restrict attention to the case $t_2^i < t_2^j$. To ensure the interchanged assignments remain feasible, both surgery days must fall after the later of the two arrival days. If not, an interchange argument won't work as it would force the construction of a schedule where one of the patients is scheduled before their arrival day, resulting in infeasibility. Hence, any suboptimality can only occur from the following cases: $t_1^i < t_1^j \leq t_2^i < t_2^j$ and $t_1^j \leq t_1^i \leq t_2^i < t_2^j$. We consider them below.

Case 1: $t_1^i < t_1^j \leq t_2^i < t_2^j$. Define the interchanged schedule $\mathcal{S}'$ such that the interchanged assignments are (t_1^i, t_2^j) and (t_1^j, t_2^i). Note that all other assignments are unchanged, hence the difference in the scheduling costs only arises from the interchange.

The difference in total scheduling cost is

$$\begin{aligned} d \;=\; \text{cost}(\mathcal{S}) - \text{cost}(\mathcal{S}') &= \Big[w_2(t_2^i - t_1^i + 1) + w_1(t_2^j - t_1^j + 1)\Big] \\ &\quad - \Big[w_2(t_2^j - t_1^i + 1) + w_1(t_2^i - t_1^j + 1)\Big] \\ &= w_2(t_2^i - t_2^j) + w_1(t_2^j - t_2^i) \\ &= (w_1 - w_2)(t_2^j - t_2^i). \end{aligned}$$

Since $t_2^j > t_2^i$ and $w_1 > w_2$, we have $d > 0$, so $\mathcal{S}'$ strictly improves upon $\mathcal{S}$. This contradicts the optimality of $\mathcal{S}$.

Case 2: $t_1^j \leq t_1^i \leq t_2^i < t_2^j$. We can construct a similar argument for this case as well, where the only difference arises in the arrival days. The arrival day terms cancel out, leaving a cost difference that is identical to Case 1 and is strictly positive, again contradicting optimality of $\mathcal{S}$.

Since the cost improvement holds in both cases regardless of the relative order of arrival days, any schedule in which a low-priority patient is assigned before a high-priority patient can be strictly improved by interchange. It follows that any optimal schedule must assign all high-priority patients before all low-priority patients. □

Based on Proposition 1, we can produce optimal reservations with only estimates for future high-priority arrivals. Restricting the future state space to strictly high priority further reduces the computational complexity for solving programs with longer horizons and compute reservations to optimality.

4.2 Updating Reservation Profile

As high-priority demand arrives over the interval and gets scheduled or deferred right away, we must update the reservations computed from the re-solve periods dynamically. Otherwise, we risk over-reserving future capacity for high-priority arrivals that have already been registered into the system. $\mathcal{RP}_k$ produces a solution $\tilde{\mathbf{X}}$ that instantiates the reservation profile in period k. Given the solution $\tilde{\mathbf{X}}$, we can obtain updated $\tilde{\mathbf{X}}'$ for all periods $t > k$ as follows. If t is a re-solve period, the old reservations are purged, and new reservations are freshly computed with the updated remaining capacity state. Otherwise, the existing reservation profile from period t is updated according to

$$\tilde{x}^{(1)'}_{t_1,t_2}(t') := \begin{cases} \tilde{x}^{(1)}_{t_1,t_2}(t') & \text{if } t_1 > t,\ t' \leq t_2, \\ 0 & \text{otherwise.} \end{cases} \tag{4.7}$$

This update zeros out all the reservations whose arrival day $t_1 \leq t$, since these correspond to periods that have already executed or are being handled in the current period. Retaining older reservations as policy parameters for future periods would attempt to reserve capacity for arrivals whose window has already closed. The update preserves $\tilde{x}^{(1)}_{t_1,t_2}$ for all future arrival days $t_1 > t$, which constitute the unrealized portion of the reservation profile. This also remains consistent with the re-solve case ($N = 1$), in which $\mathcal{RP}_k$ is solved fresh at each period over the remaining booking horizon and naturally produces a reservation profile indexed only over future arrival days. In both cases, whether updating an existing profile or computing a new one, the active reservation profile at period t contains entries only for arrival days $t_1 > t$, ensuring a consistent future aggregate demand across all re-solve intervals N.

4.3 Interval Period

Between consecutive re-solve periods, the system evolves through a sequence of interval periods $\tau \in \mathcal{T}_k$, during which realized demand arrives and scheduling decisions must be made with respect to existing reservations. The interval period model $\mathcal{IP}_\tau$ is formulated as a bi-level program. At each interval period $\tau \in \mathcal{T}_k$, high-priority patients are scheduled first via the inner problem $\mathcal{IP}^{(1)}$. The resulting decisions $\mathbf{x}^{(1)}(\tau)$ are then passed to the outer problem, which schedules low-priority patients over the residual capacity. Crucially, this residual capacity accounts not only for the high-priority decisions made at period τ, but also for the updated reservation profile $\tilde{\mathbf{X}}'$ carried forward from the most recent re-solve period via (4.7). We first provide the formulation for the inner program.

4.3.1 Inner Program

The element-wise transition dynamics for the capacity and high-priority queue states between two consecutive interval periods are defined as

$$\begin{aligned} c_{t_2}(\tau+1) &= \Big(c_{t_2}(\tau) - \sum_{t_1=1}^{\tau} s_1\, x^{(1)}_{t_1,t_2}(\tau)\Big)^+, \quad \tau \le t_2 \le \min(\tau+B-1, T) \\ c_{t_2}(\tau+1) &= c_{t_2}(\tau), \qquad\qquad\qquad\qquad\qquad\qquad t_2 > \min(\tau+B-1, T). \end{aligned} \tag{4.8}$$

The element-wise dynamics of the queue states between two consecutive periods can be written as

$$\begin{aligned} q^{(1)}_{t_1}(\tau+1) &= \Big(q^{(1)}_{t_1}(\tau) - \sum_{t_2=\tau}^{\min(\tau+B-1, T)} x^{(1)}_{t_1,t_2}(\tau)\Big)^+, \quad t_1 < \tau \\ q^{(1)}_{t_1}(\tau+1) &= \Big(D^{(1)}_{t_1} - \sum_{t_2=\tau}^{\min(\tau+B-1, T)} x^{(1)}_{t_1,t_2}(\tau)\Big)^+, \qquad t_1 = \tau. \end{aligned} \tag{4.9}$$

The capacity constraint for the inner program is defined as

$$\sum_{t_1=1}^{\tau} s_1\, x^{(1)}_{t_1,t_2}(\tau) \le c_{t_2}(\tau), \quad \forall\, t_2 \in [\tau, \min(\tau+B-1, T)], \tag{4.10}$$

and the queue constraints are defined as

$$\begin{aligned} \sum_{t_2=\tau}^{\min(\tau+B-1,T)} x^{(1)}_{t_1,t_2}(\tau) &\le q^{(1)}_{t_1}(\tau), \quad \forall\, t_1 < \tau, \\ \sum_{t_2=\tau}^{\min(\tau+B-1,T)} x^{(1)}_{t_1,t_2}(\tau) &\le D^{(1)}_{t_1}, \quad t_1 = \tau. \end{aligned} \tag{4.11}$$

The feasible action set for the inner program at interval period τ can then be compactly

written as

$$\mathcal{Y}_\tau^{(1)}\big(\mathbf{c}(\tau),\, \mathbf{q}^{(1)}(\tau)\big) = \big\{\mathbf{x}^{(1)}(\tau) \mid (4.8),\, (4.9),\, (4.10),\, (4.11)\big\}. \tag{4.12}$$

The inner problem $\mathcal{IP}_\tau^{(1)}$ is formulated as:

$$\mathcal{IP}_\tau^{(1)} := \min_{\mathbf{x}^{(1)}(\tau)} \quad f_\tau(\mathbf{x}^{(1)}(\tau)) + g_{\tau+1}(\mathbf{q}^{(1)}(\tau+1)) \tag{4.13a}$$

$$\text{s.t.} \quad \mathbf{x}^{(1)}(\tau) \in \mathcal{Y}_\tau^{(1)}\big(\mathbf{c}(\tau),\, \mathbf{q}^{(1)}(\tau)\big) \tag{4.13b}$$

4.3.2 Outer Program

We now provide the formulation for the outer program. The element-wise transition dynamics for the capacity and low-priority queue states between two consecutive interval periods are defined as

$$\begin{aligned} c_{t_2}(\tau+1) = \Big(c_{t_2}(\tau) &- \sum_{p\in P}\sum_{t_1=1}^{\tau} s_p\, x_{t_1,t_2}^{(p)}(\tau) \\ &- \sum_{t_1=\tau+1}^{t_2}\sum_{t'=t_1}^{t_2} s_1\, \tilde{x}_{t_1,t_2}^{(1)'}(t')\Big)^+, \quad \tau \le t_2 \le \min(\tau+B-1, T) \\ c_{t_2}(\tau+1) = c_{t_2}(\tau), &\qquad\qquad t_2 > \min(\tau+B-1, T). \end{aligned} \tag{4.14}$$

Precisely, the capacity available for low-priority patients on day t_2 is the residual capacity after high-priority scheduling, reduced further by the active reservations protecting that day for future high-priority arrivals. The second summation term captures these reservations by aggregating, over all future high-priority arrival days $t_1 \in (\tau, t_2]$ and all intermediate decision periods $t' \in [t_1, t_2]$, the reserved slots that have not yet been consumed. This construction becomes necessary when the booking horizon is short, as reservations placed by $\mathcal{RP}_k$ may span multiple decision periods.

The element-wise dynamics of the queue states between two consecutive periods can be written as

$$\begin{aligned} q_{t_1}^{(2)}(\tau+1) &= \Big(q_{t_1}^{(2)}(\tau) - \sum_{t_2=\tau}^{\min(\tau+B-1,\,T)} x_{t_1,t_2}^{(2)}(\tau)\Big)^{+}, \quad t_1 < \tau \\ q_{t_1}^{(2)}(\tau+1) &= \Big(D_{t_1}^{(2)} - \sum_{t_2=\tau}^{\min(\tau+B-1,\,T)} x_{t_1,t_2}^{(2)}(\tau)\Big)^{+}, \qquad t_1 = \tau. \end{aligned} \tag{4.15}$$

The capacity constraint for the outer program ensures that low-priority scheduling respects the residual capacity after high-priority commitments and active reservations:

$$\sum_{p\in P}\sum_{t_1=1}^{\tau} s_p\, x_{t_1,t_2}^{(p)}(\tau) + \sum_{t_1=\tau+1}^{t_2}\sum_{t'=t_1}^{t_2} s_1\, \tilde{x}_{t_1,t_2}^{(1)\prime}(t') \le c_{t_2}(\tau), \quad \forall\, t_2 \in [\tau, \min(\tau+B-1, T)], \tag{4.16}$$

and the queue constraints are defined as

$$\begin{aligned} \sum_{t_2=\tau}^{\min(\tau+B-1,T)} x_{t_1,t_2}^{(2)}(\tau) &\le q_{t_1}^{(2)}(\tau), \quad \forall\, t_1 < \tau, \\ \sum_{t_2=\tau}^{\min(\tau+B-1,T)} x_{t_1,t_2}^{(2)}(\tau) &\le D_{t_1}^{(2)}, \quad t_1 = \tau. \end{aligned} \tag{4.17}$$

The feasible action set for the outer program at interval period τ can then be compactly written as

$$\mathcal{Y}_{\tau}^{(2)}\big(\mathbf{c}(\tau),\, \mathbf{q}^{(2)}(\tau),\, \tilde{\mathbf{X}}',\, \mathbf{x}^{(1)}(\tau)\big) = \big\{\mathbf{x}^{(2)}(\tau) \mid (4.14),\, (4.15),\, (4.16),\, (4.17)\big\}. \tag{4.18}$$

Putting everything together, the program solved at each interval period $\tau \in \mathcal{T}_k$ can be formulated as

$$
\begin{aligned}
\mathcal{IP}_\tau\big(\tilde{\mathbf{X}}'\big) \;:=\; \min_{\mathbf{x}(\tau)} \quad & f_\tau(\mathbf{x}^{(2)}(\tau)) + g_{\tau+1}(\mathbf{q}^{(2)}(\tau+1)) && \text{(4.19a)}\\
\text{s.t.} \quad & \mathbf{x}^{(1)}(\tau) = \arg\min_{\mathbf{x}^{(1)}} \Big\{\mathcal{IP}^{(1)}\Big\} && \text{(4.19b)}\\
& \mathbf{x}^{(2)}(\tau) \in \mathcal{Y}_\tau^{(2)}\big(\mathbf{c}(\tau),\, \mathbf{q}^{(1)}(\tau),\, \mathbf{q}^{(2)}(\tau),\, \tilde{\mathbf{X}}',\, \mathbf{x}^{(1)}(\tau)\big) && \text{(4.19c)}
\end{aligned}
$$

4.4 Reservation Policy and Algorithm

The re-solve program (4.6) and interval program (4.19) together induce a *Reservation policy* $\pi = \{\mu_t\}_{t=1}^{T}$, where the decision rule at each period $t \in [T]$ is

$$
\mu_t\big(\mathbf{c}(t),\, \mathbf{q}^{(1)}(t),\, \mathbf{q}^{(2)}(t),\, \tilde{\mathbf{X}}'\big) \;:=\; \begin{cases} \Big[\arg\min_{\tilde{\mathbf{x}}} \;\mathcal{RP}_t\Big]_{\cdot,\,t} & \text{if } t \in \mathcal{K}, \\ \arg\min_{\mathbf{x}(t)} \;\mathcal{IP}_t\big(\tilde{\mathbf{X}}'\big) & \text{otherwise} \end{cases} \tag{4.20}
$$

At re-solve periods, $(\tilde{\mathbf{x}}^{(1)}(t), \tilde{\mathbf{x}}^{(2)}(t))$ are extracted from the current-period columns of $\mathcal{RP}_t$, while at interval periods, the decisions $(\mathbf{x}^{(1)}(t), \mathbf{x}^{(2)}(t))$ are obtained by solving the interval scheduling problems with the reservations produced from the most recent re-solve and accurate representation of the reservation profile, $\tilde{\mathbf{X}}'$, as mentioned in (4.7). We also present an algorithm that makes the decision rule (4.20) operational by specifying the exact sequence of steps executed at each period $t \in [T]$.

The algorithm operates on a rolling basis over the planning horizon T, alternating between re-solve periods and interval periods. If current period is a re-solve period, then the $\mathcal{RP}_k$ is solved over the remaining horizon using the latest realized state $\big(\mathbf{c}(k),\, \mathbf{q}^{(1)}(k),\, \mathbf{q}^{(2)}(k)\big)$, producing a reservation profile $\tilde{\mathbf{X}}'$ that is carried forward into the subsequent interval periods. During each interval period, realized demand is scheduled with the reservation profile being rolled forward via (4.7) to purge reservations for arrivals whose scheduling window has closed. This rolling structure ensures that the active reservation profile at any period

contains only future arrival days. The procedure is summarized below.

Algorithm 1 Control Algorithm

Require: Planning horizon T, Solve interval N, Booking horizon B, MSS capacity $\mathbf{c}$

$k \leftarrow 1$

$\mathbf{X} \leftarrow \mathbf{0}$ ▷ Global schedule matrix, columns indexed by $t \in [T]$

for $k \in \mathcal{K}$ **do**

  Solve $\mathcal{RP}_k$ (4.6) to obtain $(\tilde{\mathbf{X}}^{(1)*},\ \tilde{\mathbf{x}}^{(2)*}(k))$ ▷ Re-solve period

  $\mathbf{X}[\cdot, k] \leftarrow \big(\tilde{\mathbf{x}}^{(1)*}(k),\ \tilde{\mathbf{x}}^{(2)*}(k)\big)$

  $\tilde{\mathbf{X}}' \leftarrow \tilde{\mathbf{X}}^{(1)*}[\cdot, k+1:T]$ ▷ Set reservations

  for $\tau \in \mathcal{T}_k$ **do** ▷ Interval periods

    $\tilde{\mathbf{X}}'[\cdot, \tau+1:T] \leftarrow$ Update reservations with (4.7)

    Solve $\mathcal{IP}_\tau\big(\tilde{\mathbf{X}}'\big)$ (4.19) to obtain $(\mathbf{x}^{(1)*}(\tau),\ \mathbf{x}^{(2)*}(\tau))$

    $\mathbf{X}[\cdot, \tau] \leftarrow \big(\mathbf{x}^{(1)*}(\tau),\ \mathbf{x}^{(2)*}(\tau)\big)$

  end for

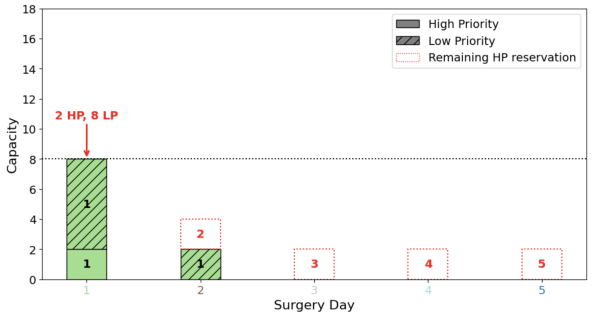
end for

return X

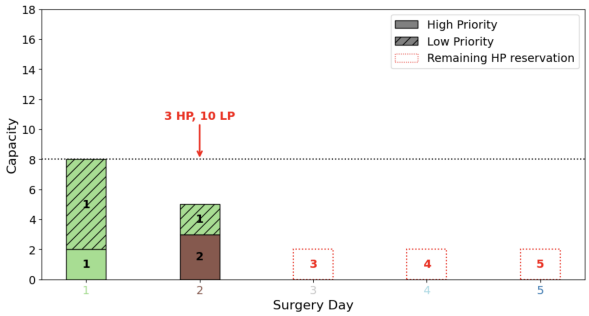
4.5 Illustrative Example

In this section, we walk through a numerical example and illustrate the mechanics of Algorithm 1. We consider a small instance with a planning horizon of $T = 5$ days, a booking horizon of $B = 5$ days, and a re-solve interval of $N = 5$ days, meaning $\mathcal{RP}_k$ is solved only once at $k = 1$ and the system evolves entirely through interval periods $\tau \in \{2, 3, 4, 5\}$. We restrict this numerical example to $N = T$ to illustrate the reservation update mechanics of (4.7) across the full sequence of interval periods, as the re-solve case $N = 1$ trivially reduces to daily re-solve per period with no reservation carry-forward. The OR capacity vector is $\mathbf{c} = (8, 8, 8, 8, 8)$. There are two priority classes: high-priority with weight $w_1 = 84$ and low-priority with weight $w_2 = 12$, both with unit service time $s_p = 1$. The figure below traces the evolution of the committed schedule and the active reservation profile $\tilde{\mathbf{X}}'$ across all five periods. In each figure, filled bars represent committed patient assignments and dotted red boxes represent the remaining high-priority reservations on

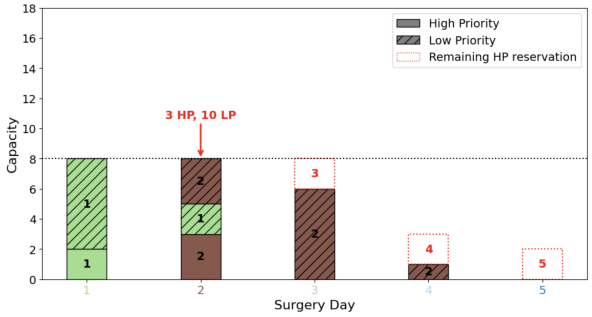
each surgery day. Interior labels on filled bars indicate the realized arrival day t_1, while interior labels on dotted bars indicate the predicted arrival day t_1 for which the reservation was originally placed.



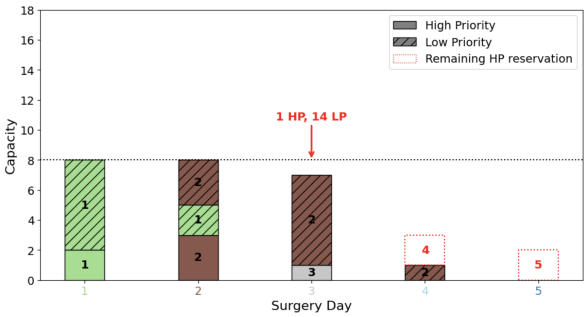
(a) Re-solve period $\mathcal{RP}_1$: committed schedule and active high-priority reservation profile after solving $\mathcal{RP}_1$. We set $\hat{D}_t^{(1)} = 2$ for all $t \in \{2, 3, 4, 5\}$



(b) Interval period $\mathcal{IP}_2$, inner program: high-priority patient arriving on day 2 is scheduled and the reservation for $t_1 = 2$ has been retired while the future reservations persist.

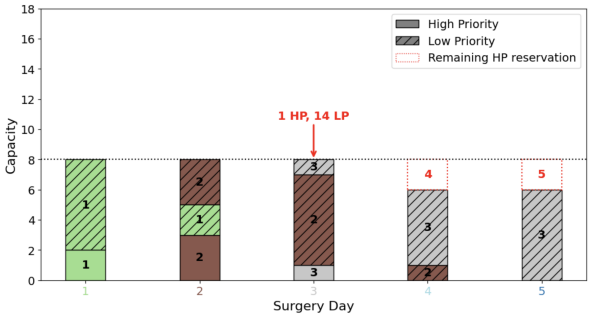


(c) Interval period $\mathcal{IP}_2$, outer program: low-priority patients are scheduled in residual capacity after day 2 high-priority commitments and active reservations are enforced.

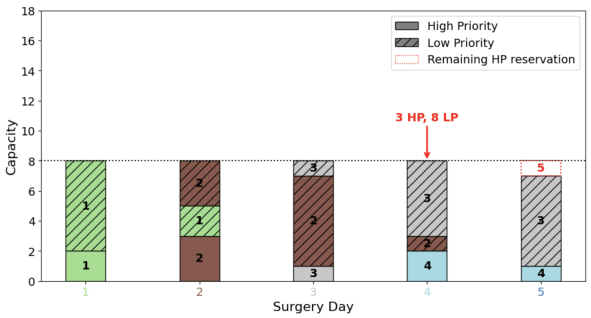


(d) Interval period $\mathcal{IP}_3$, inner program: high-priority patient arriving on day 3 is scheduled and the reservation for $t_1 = 3$ has been retired while the future reservations persist.

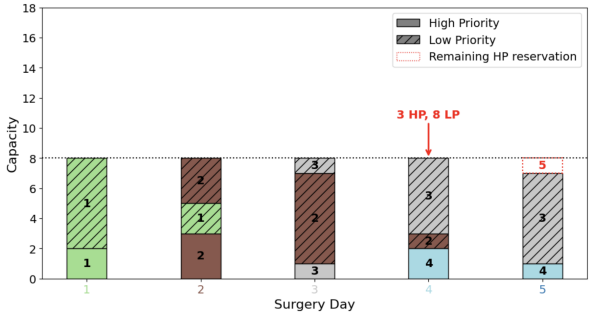
Figure 4.3: Numerical Example illustrating the control algorithm



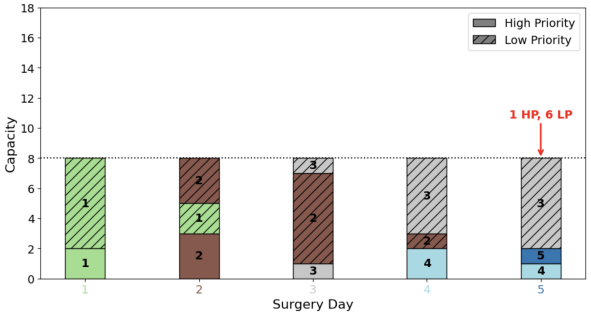
(e) Interval period $\mathcal{IP}_3$, outer program: We can observe the impact of the reservations on Day 4 and Day 5. Without them, day 3 low-priority demand would have forced potential day 3 and day 4 high-priority patients to be scheduled later.



(f) Interval period $\mathcal{IP}_4$, inner program: high-priority patients arriving on day 4 are scheduled and the reservation for $t_1 = 4$ has been retired while the future reservations persist.



(g) Interval period $\mathcal{IP}_4$, outer program: no low-priority patients are scheduled as there is no capacity remaining in the planning window



(h) Interval period $\mathcal{IP}_5$, inner program: There is no further capacity remaining for low-priority arrivals after 1 high-priority arrival gets scheduled.

Figure 4.3: Numerical example illustrating the control algorithm (Continued)

Chapter 5

Case Study

In this chapter, we present a case study to evaluate the performance of our algorithm on arrival sequences sampled from a distribution that is constructed using real arrival data from a local hospital. We perform numerical experiments on key problem parameters and provide scheduling insights. We first provide detailed description of the dataset and the experiment setting. Then, we perform the numerical experiment at the end.

5.1 Data Description and Experimental Setup

The dataset provided to us includes patient and administrative records from 2019 till 2025. However, we only use patient records from recent years, with arrival dates starting from January 1, 2022 and ending on December 31, 2025. The exclusion of data prior to 2022 in our analysis is justified by any post-COVID changes to the hospital operations. One such possible change is the change in arrival pattern of the patients, which can be observed in Figure 5.1. We can notice that the total arrivals are at the lowest between 2019 and mid-2021. The arrivals gradually increase mid-2021 onward as the hospital allows more surgeries after COVID. To reflect accurate total demand and hospital resources, we only consider the arrival and capacity data past 2022.

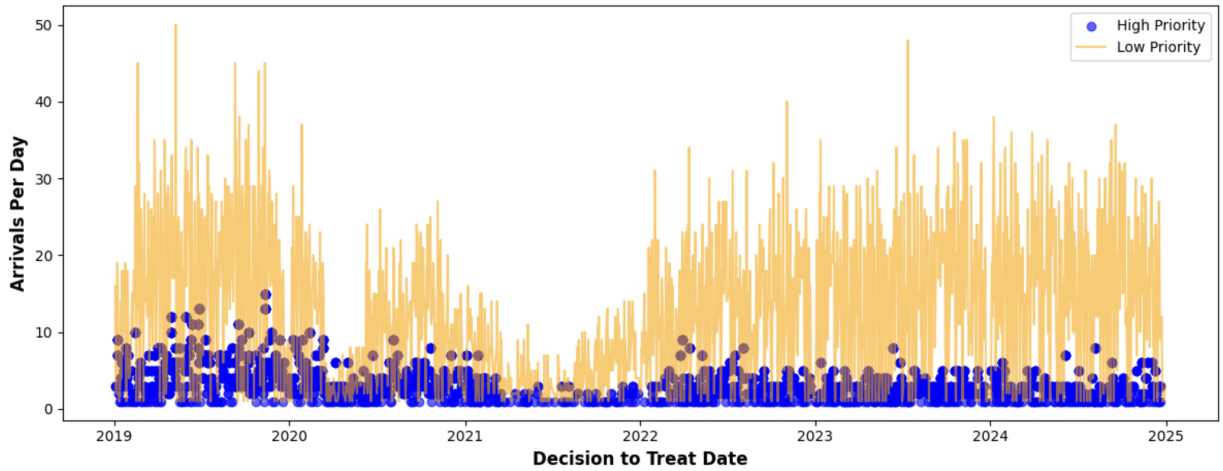
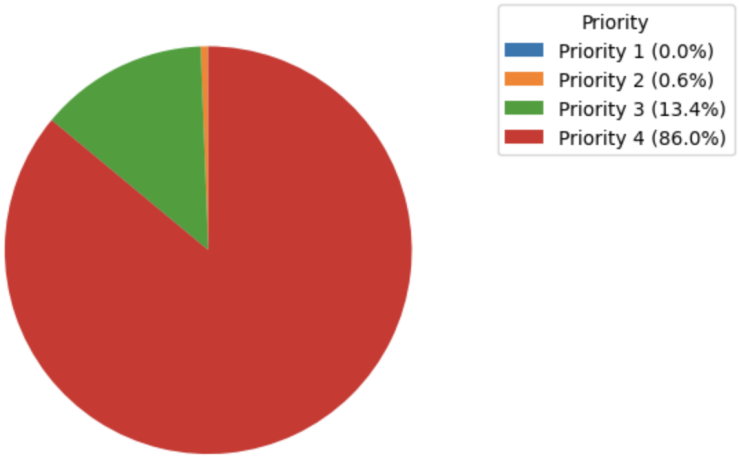


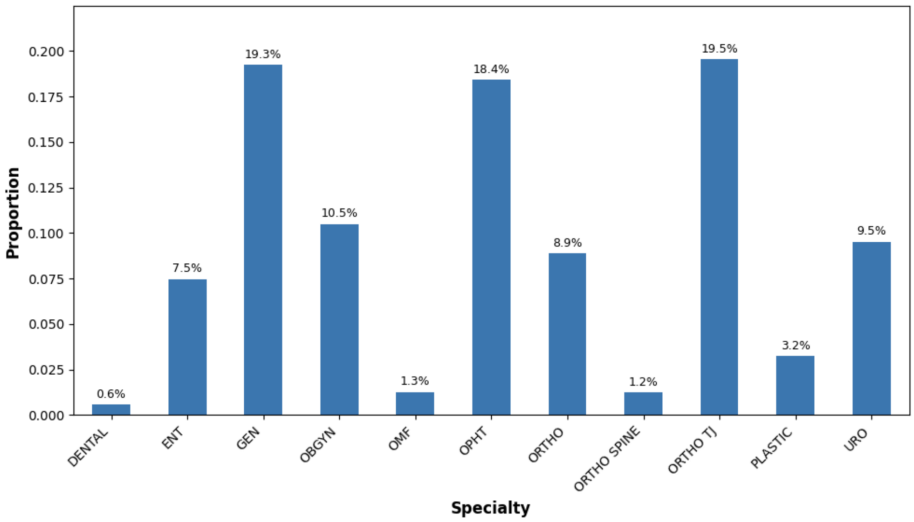
Figure 5.1: Arrivals from 2019 to 2024

5.1.1 Patient Priorities

Patients are assigned one of four priority levels based on clinical urgency as determined by the surgeon. Figure 5.2a illustrates the priority composition for patients with recorded priorities; some entries were missing, possibly due to incomplete updates from the WTIS system. Among patients with recorded priorities, 86% are Priority 4, over 13% are Priority 3, and fewer than 1% are Priorities 1 and 2 combined. To align with our two-priority scheduling framework, we consolidate Priorities 1, 2, and 3 into high priority and treat Priority 4 as low priority. Figure 5.2b shows the surgical specialty composition across the full dataset. As no missing values were present in the specialty column, it reflects the complete patient mix across all specialties at the hospital.



(a) Number of patients by priority



(b) Proportion of specialties

Figure 5.2: Patient composition and specialty distribution

5.1.2 Service Capacity

An important aspect of OR planning is the development of the Master Surgical Schedule (MSS). An MSS specifies for each day t_2 of the planning horizon, a list of ORs available

for operation and surgical specialties. If there are n ORs available on day t_2, each OR will be assigned a specialty for the day. A patient arriving on day t_1 must be scheduled to a day $t_2 \geq t_1$ in the planning horizon based on the capacity available for that specialty on that day. Each OR that is available on a given day is assumed to contribute 7.5 hours of scheduling capacity for the given specialty, representing the typical duration of a full working day at the hospital (8:00 AM – 3:30 PM). Table 5.1 illustrates a sample structure of the master surgery schedule used by the hospital for scheduling.

Table 5.1: Sample Master Surgical Schedule

Date	**Day of Week**	**Day**	**Week**	**Room 1**	**Room 2**
31-Mar-2025	2	Monday	1	OPHT	GEN
1-Apr-2025	3	Tuesday	1	URO	GEN
2-Apr-2025	4	Wednesday	1	URO	GEN
3-Apr-2025	5	Thursday	1	ORTHOS	OPHT
4-Apr-2025	6	Friday	1	ORTHOS	GEN

5.1.3 Service Duration

The hospital offers over 700 distinct surgical procedures. Their durations depend on a lot of factors such as the patient's condition, surgeon, type of procedure, staff availability, etc. To combat the randomness in the duration of the procedure, we simply approximate the duration for each surgery within a specialty group using the mean of all the surgery durations within the specialty. It is worth noting that the data set includes both the actual duration of the surgery and booked duration but we only use booked duration for our calculations to better match the scheduling environment of the hospital. The average durations for each specialty are mentioned in Table 5.2.

Table 5.2: Service durations

Surgery specialty group	**Average duration (hour)**
URO	1.2
GEN	1.5
OPHT	0.3
DENTAL	1.6
ENT	1.4
OBGYN	1.6
OMF	1.8
ORTHO	1.5
ORTHO SPINE	1.7
ORTHO TJ	1.2
PLASTIC	3.1

After removing patients with missing priority information (as discussed in Section 5.1.1), the per-specialty demand and capacity utilization were too low to yield meaningful simulation results for several specialties. Rather than imputing the missing priority levels, we preserve the priority composition of the observed data and consolidate three specialties — GEN, ORTHO TJ, and OPHT — into a single combined specialty, aggregating sufficient demand and utilization. To set the total capacity, we refer to the combined Master Surgical Schedules for the three specialties as used by the hospital in 2025. This yielded a total capacity of 5,647 surgical hours. We compute the service duration for the combined specialty as the average booked duration across all procedures within the three constituent specialties, yielding $s_p = 1$ hour for all priority classes. This remains consistent with our equal service duration assumption and isolates the effect of the scheduling policy from variation in service durations across the considered specialties, as observed in Table 5.2. Since the simulation demand is derived only from patients with recorded priority levels, using the full MSS capacity would imply an unrealistically low utilization ratio. We therefore scale the total capacity down proportionally to the fraction of aggregate demand retained, resulting

in a truncated capacity of 3,060 hours. Figure 5.3 illustrates the capacity available on each working day of the year. Note that the figure only shows the capacities for the weekdays as the data reported no surgeries to be booked during the weekends.

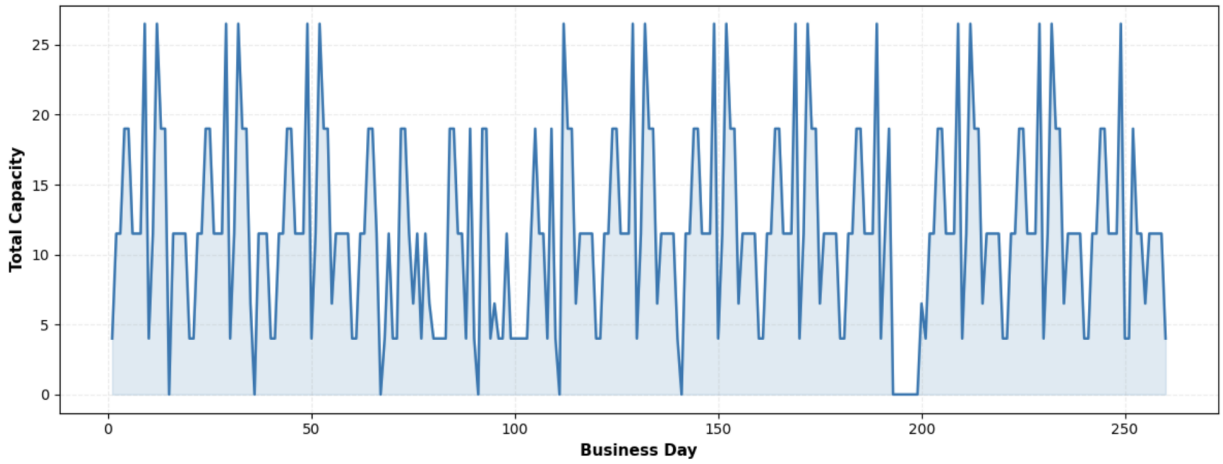


Figure 5.3: Daily Capacity

5.1.4 Experiment Setup

We evaluate the performance of the proposed policy in relation with two other policies, namely *Clairvoyant policy* and *Myopic policy*. A Clairvoyant policy assumes knowledge of the arrivals over the full planning horizon. As demand is known in advance, we simply solve a deterministic version of the problem. Finally, as the name suggests, the Myopic policy allocates capacity at each timestep to higher priority patients before lower priority patients without any look-ahead. The pseudocode for the Myopic policy is provided in Algorithm 2. Benchmarking our proposed Reservation policy against the Clairvoyant and Myopic allows us to study the value of future reservations as we vary different problem parameters. To compare the policy performance, we employ Common Random Numbers (CRN) technique. In this, we generate M independent demand paths for the full planning horizon from a fitted distribution. Each demand path is constructed using a fixed base random seed with deterministic offsets for each simulation run. This maintains reproducibility and ensures

that the exact sequence of demand paths is consistently used across different experimental runs (Glasserman and Yao, 1992). We aim to compute the *expected cost* (C_{policy}) of each the policy over all the simulations and compare them with each other. The expected cost can be computed as the ratio of sum of total waiting costs across each run and the total number of runs.

$$C_{\text{policy}} = \frac{1}{M}\left(\sum_{t=1}^{T}\sum_{t_1=1}^{t}\sum_{t_2=t}^{\min(t_1+B-1,\,T)}(t_2-t_1+1)\,w_p\,x^{(p)}_{t_1,t_2}(t) \;+\; \sum_{t_1=1}^{T}(T-t_1+1)\,w_p\,q^{(p)}_{t_1}(T+1)\right)$$

Algorithm 2 Myopic Policy

1: **for** $t_1 = 1 : T$ **do**
2: **Schedule** high-priority demand
3: Update remaining capacity/queue
4: **Schedule** low-priority demand
5: Update remaining capacity/queue
6: **end for**

We use the arrival data from the hospital to decide the distribution parameters for both high- and low-priority demand. We fit a negative binomial distribution on the arrival data (see Appendix B for more details on the distribution fit). The problem parameters used in the numerical experiment are given in Table 5.3. We define baseline values for some parameters based on the available data. We conduct sensitivity analysis on operational parameters to study their impact on the waiting costs and set the baseline values accordingly.

We fix the certainty equivalent estimates for future high-priority arrivals at $\hat{D}^1_t = \mu_1$ for all $t \in \{1, \ldots, T\}$. Since high-priority patient arrivals are modeled as i.i.d. samples from NB distribution, the mean μ_1 minimizes the mean squared prediction error (MSPE), and acts as an optimal predictor of future demand random variable. Hence, the setup of this experiment isolates the evaluation of the policy performance from prediction noise, ensuring that any suboptimality observed in the proposed policy's costs comes from model approximation.

Finally, we present a minor note on the implementation of the algorithm. We borrow the notion of horizon truncation from Powell (2022) and truncate the terminal period in

the approximation program $\mathcal{RP}_k$ (4.6) from T to $k + N + B - 2$. The intuition follows directly from the booking horizon constraint and the solve interval parameter. The last interval period before the next re-solve is $k + N - 1$, and a patient arriving on that day can be scheduled at most $B - 1$ days further into the future. Any capacity beyond day $k + N + B - 2$ is therefore unreachable by decisions made within the current interval and carries no influence on the current re-solve. Hence, truncating to this horizon reduces the size of the approximation program without sacrificing any relevant lookahead, keeping the computation tractable.

Table 5.3: Parameters

Parameter	**Value**
T (Planning Horizon)	260 working days
$\mathbf{c}_t$ (Daily capacity)	Refer to Figure 5.3
w_1 (High Priority weight)	84
w_2 (Low Priority weight)	12
s_p (Service Duration)	1 hour
B (Booking Horizon)	20 working days
N (Solve Interval)	1
M (Simulation runs)	100

5.2 Numerical Results

5.2.1 Impact of Operational Parameters on Waiting Costs

In this section, we conduct a numerical study on the operational parameters to determine baseline values for the booking horizon and solve interval, which are subsequently used to evaluate the out-of-sample performance of all three policies under different demand regimes (more on this in Section 5.2.3).

Booking Horizon (B) and Solve Interval (N)

We compare the performance of all three policies with each other for both priorities to study the cost trade-offs between both the priorities. In Figure 5.4, we examine the performance of all three policies for different pairs of (B, N). For each booking horizon (B), we plot the expected cost on the y-axis and solve interval (N) on the x-axis. The Clairvoyant policy, with perfect knowledge of future arrivals, reserves capacity for high-priority patients in advance, leaving smaller capacity share for the low-priority patients. Whereas, the Myopic policy schedules greedily without anticipating future high-priority demand, achieving lower low-priority waiting costs at the expense of high-priority's patients. Hence, the Clairvoyant policy achieves lower high-priority waiting costs than Myopic at the expense of higher low-priority waiting costs, reflecting a fundamental trade-off between both priorities that exists across all the numerical experiments hereafter.

First, we will discuss the monotonic increase in expected costs for the Reservation policy as solve interval increases. Recall that at each re-solve period k, $\mathcal{RP}_k$ produces reservations based on the remaining capacity $\mathbf{c}(k)$ available at period k. As the horizon progresses, remaining capacity decreases as patients are scheduled into the future. Since reservations produced in period k are computed on relatively larger future remaining capacity than the ones computed in period $k + N$, the program overestimates the effective capacity available for low-priority patients, allowing more low-priority patients to be scheduled within the booking horizon. At $N = 1$, the program incorporates realized demand and updated remaining capacity into reservations at each period. As the remaining capacity decreases over the periods, the reinstated reservations get increasingly valuable, allowing better performance as capacity becomes more restricted in the future periods. Under infrequent re-solving, stale protection levels computed on larger remaining capacity leave more unprotected capacity available to low-priority patients in the near term, allowing more of them to be scheduled closer to their arrival dates and reducing their waiting costs. We can observe this tradeoff directly, where at lower N, the waiting costs are the highest for low-priority patients and the lowest for high-priority patients. Whereas, the opposite

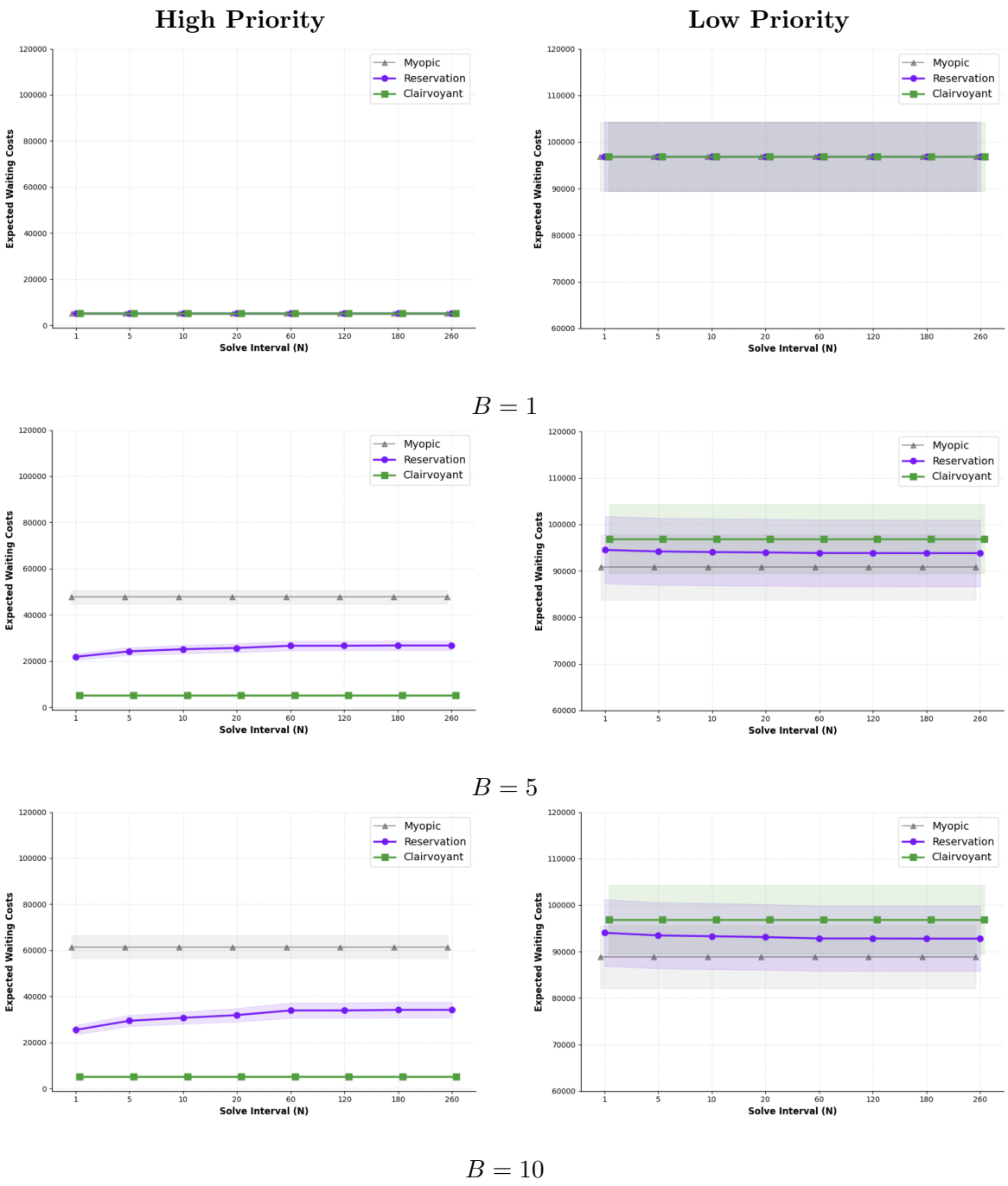


Figure 5.4: Performance of all three policies for $B \in \{1, 5, 10, 20, 60, 120, 180, 260\}$ across variations of solve interval (N). The grey, green and purple lines correspond to Myopic, Clairvoyant and Reservation policies respectively.

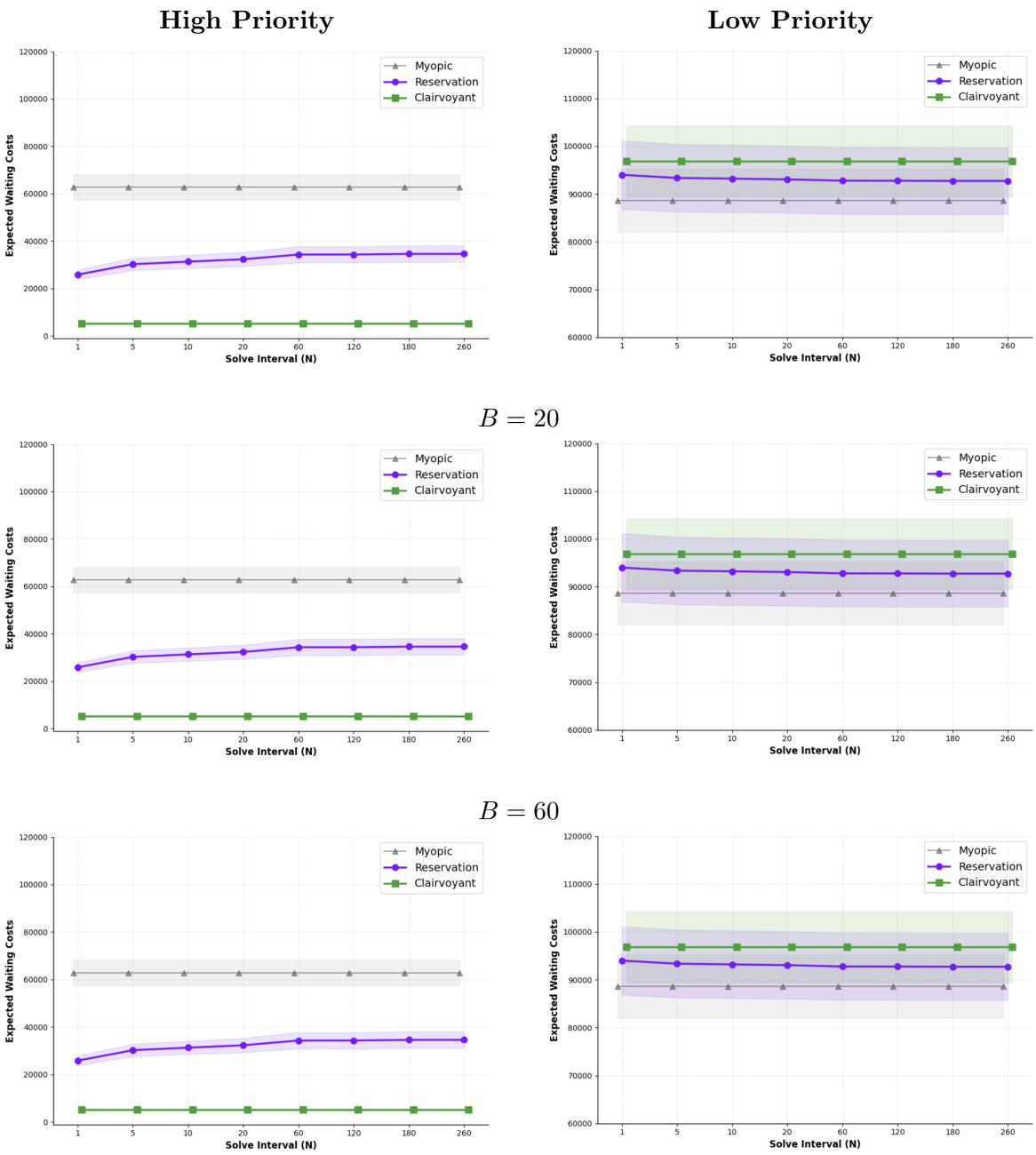
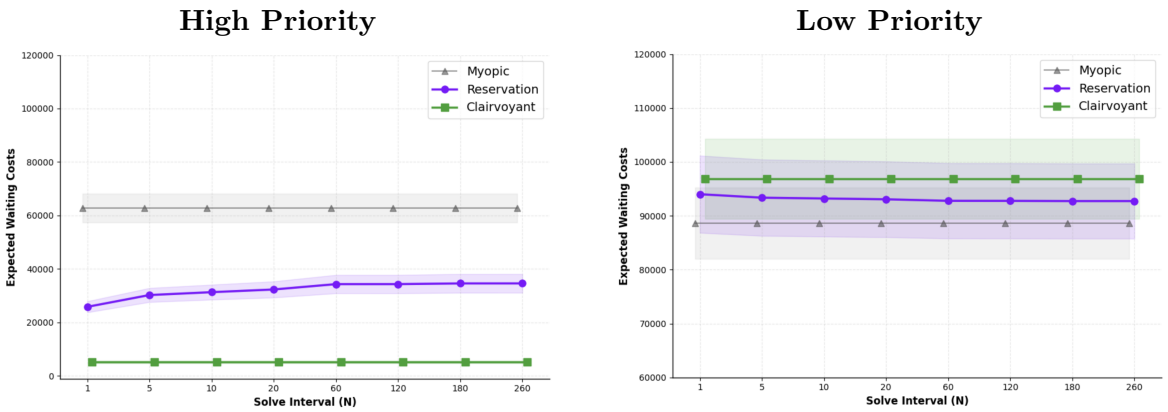
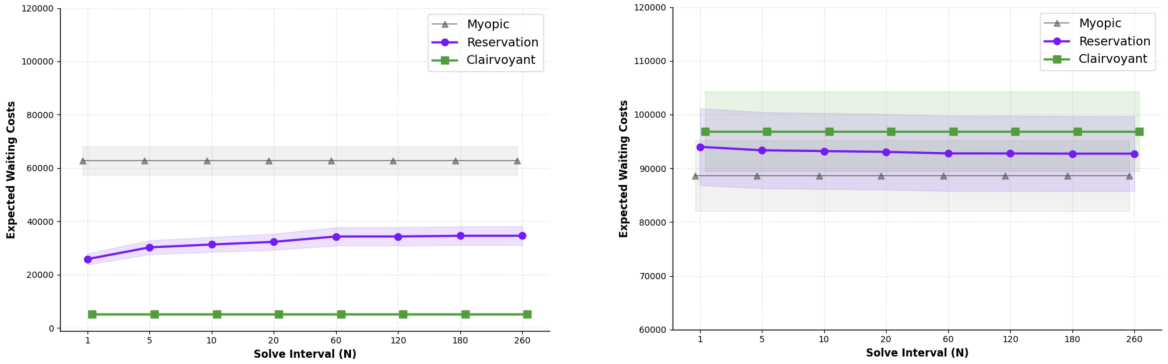


Figure 5.4: Performance of all three policies for $B \in \{1, 5, 10, 20, 60, 120, 180, 260\}$ across variations of solve interval (N) (Continued).



$B = 180$



$B = 260$

Figure 5.4: Performance of all three policies for $B \in \{1, 5, 10, 20, 60, 120, 180, 260\}$ across variations of solve interval (N) (Continued).

holds true for higher N.

Our result also corroborates the insights from Secomandi (2008), where they establish that sequential consistency is upheld when the approximation program is time-dependent and re-solved at each decision period ($N = 1$). In simpler words, sequential consistency guarantees that the updated older solution of a time-dependent program remains feasible under any realization of the future random state. In our case, the reservations that are computed in re-solve period k and if updated in a way such that the future reservations remain feasible for period $k+1$, the re-solving is weakly beneficial. Our control algorithm $\mathcal{RP}_k$ satisfies this property since $\mathcal{RP}_k$ is a time-dependent approximation program due to future demand being split by time periods, and the update specified in Section 4.2 keeps reservations for future periods untouched.

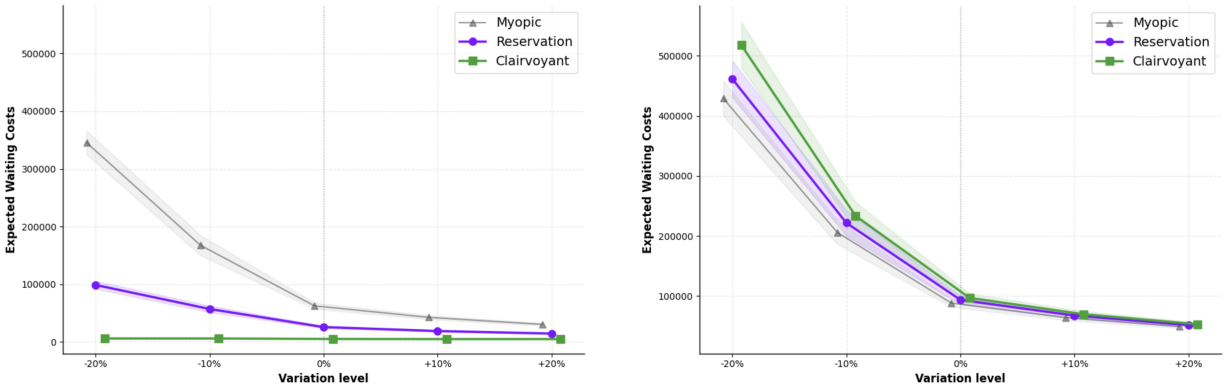
Now, we discuss the system behavior for different values of booking horizon. We observe that all three policies yield identical high-priority expected costs at $B = 1$. At a booking horizon of one day, every policy schedules only the current day's demand, and since all policies respect priority ordering, high-priority patients are always scheduled first on their arrival day. With no future commitment involved, the Reservation and Myopic policies produce optimal decisions in every decision period. Secondly, we observe that the high-priority waiting costs for the Clairvoyant policy remain flat due to full look-ahead over the planning horizon. As all the arrivals are known in advance, Proposition 1 applies, and the offline integer program makes an optimal allocation of high-priority patients for each booking horizon.

As B increases, the expected costs shift upward for all three policies. At smaller booking horizons, the future capacity landscape encountered at each period is lightly committed. Since high-priority patients are always scheduled before low-priority patients, they are assigned dates close to their arrival, and if sufficient capacity is unavailable within the booking horizon, they are queued at no cost, keeping their waiting costs low. Under such priority ordering, low-priority patients are left with access to only restricted residual capacity. As B increases, however, the provider can commit capacity further into the future at each period, shrinking the near-term capacity available for incoming high-priority arrivals and inducing longer wait times. Conversely, the expanded booking horizon grants

low-priority patients access to larger near-term capacity, allowing more of them to be scheduled closer to their arrival dates. We further observe that the performance of the policies does not change beyond $B = 20$. This happens because all patients (queued and realized) can be assigned within the current horizon, so the optimizer faces the same feasible set regardless of whether B increases further. For a higher utilization ratio, we would expect this to happen at longer booking horizons.

Finally, we set the baseline parameters for $B = 20$ and $N = 1$ for all subsequent numerical experiments. We pick $B = 20$ as it consistently yields the highest expected costs across all policies, making it one of the smallest booking horizons that induces the most stress to the system. Lastly, the choice of $N = 1$ is motivated by the best performance of the Reservation policy under frequent re-solving.

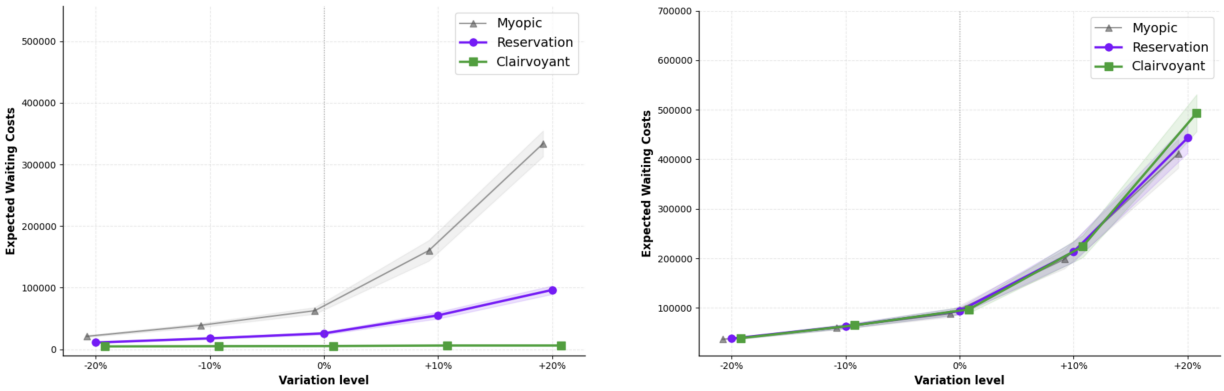
Capacity (c_t) and Service Time (s_p)



(a) High-Priority Expected Waiting Costs

(b) Low-Priority Expected Waiting Costs

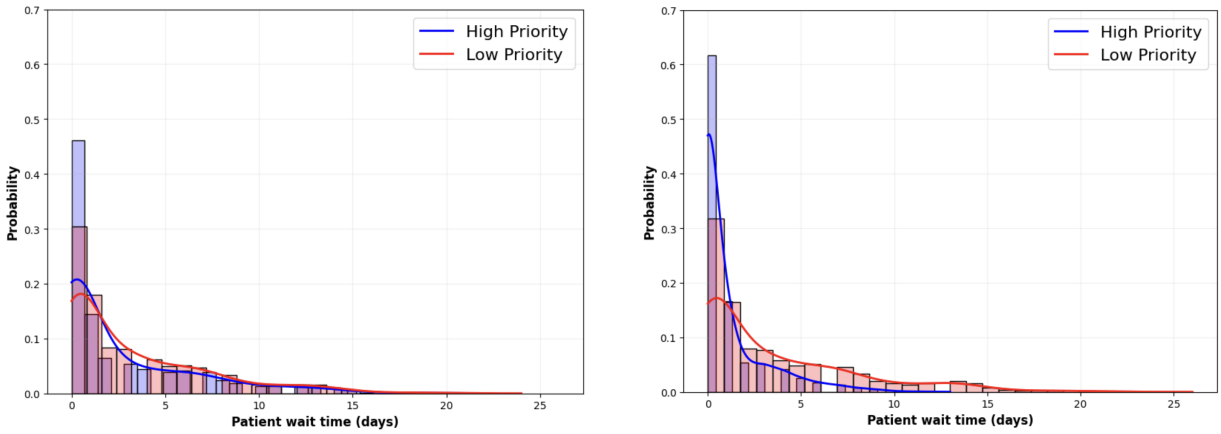
Figure 5.5: Sensitivity Analysis on Capacity



(a) High-Priority Expected Waiting Costs (b) Low-Priority Expected Waiting Costs

Figure 5.6: Sensitivity Analysis on Service Time (s_p)

We present the plots for both the sensitivity cases together because the variation in both affect capacity due to the shared knapsack constraint. From Figure 5.5, we can observe that the waiting costs decrease as the daily capacity increases whereas the costs increase as the service time increases. An increase in the total capacity available or decrease in the service time of the surgeries allow more patients to be scheduled as the ratio $\frac{\mathbf{c}_t}{s_p}$ increases. The opposite holds true when the capacity decreases or the service time increases as the value of the ratio decreases. When total capacity increases, fewer patients are unscheduled or delayed, so the total waiting cost per iteration is lower on average. Moreover, the difference in costs between Myopic and Reservation policies also shrinks because the value of the reservations reduces when the system becomes less capacity-constrained. However, when total capacity decreases, the system operates under high utilization and the future reservations become more valuable, causing the difference in costs between Myopic and Reservation policies to widen. In other words, insufficient capacity forces the patients to incur higher waiting costs and be scheduled in the later days, whereas sufficient capacity allows a larger proportion of patients be scheduled closer to their arrival day.



(a) Myopic (b) Reservation

Figure 5.7: Distribution of the wait time (days) under Myopic and Reservation policies

Table 5.4: Days within which 90% of patients received surgery

Priority	**Myopic**	**Reservation**	**Difference**
High Priority	8	4	−50%
Low Priority	9	10	+11%

5.2.2 Comparison of Patient Wait Time between Myopic and Reservation Policies

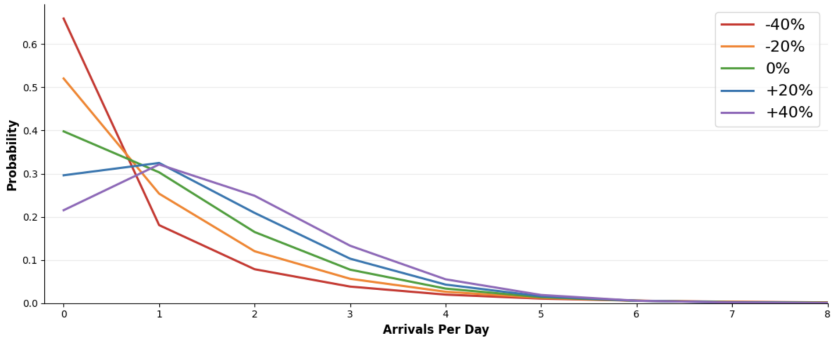
As mentioned earlier, each specialty offers a range of surgeries and each surgery has a range of wait time targets set by the provincial government. Specifically, the hospital is expected to schedule around 90% of the total patient volume within the specified target days. As each surgery within the combined group of specialties has a wide range of target days, we interpret the improvement in the 90^{th} percentile wait days of the Reservation policy over the Myopic policy. In Figure 5.7, we show the distribution of wait days for both the policies, where the wait days are computed by combining all 100 schedules obtained from the simulation with baseline parameters. The figure illustrates that the tail of the high-priority wait time distribution is substantially compressed under the Reservation policy relative to the Myopic policy. The low-priority distributions under both policies are broadly

similar in shape. Though the low-priority distribution for Reservation policy produces a slight rightward shift, the increase in low-priority wait time is significantly smaller than the improvement in high-priority wait time, as summarized in Table 5.4.

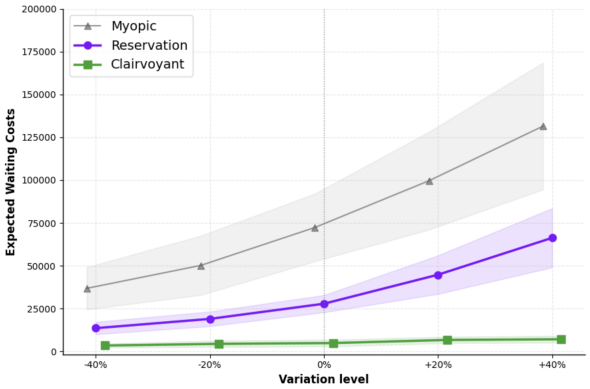
5.2.3 Out-of-Sample Performance Under Demand Shifts

In this section, we evaluate the out-of-sample performance of the algorithm and compare it with the other benchmarks. We study the robustness of the policy to shifts in the underlying demand distribution and examine the value of reservations when the true demand drifts away from certainty equivalent estimates. We split the results into four subsections, where we perturb the first and second moments of both the priorities. For each moment, we perform perturbations while maintaining the validity of the negative binomial distribution. Throughout the section, we refer to each perturbation as a variation level and plot the expected waiting costs against the moment perturbation of interest.

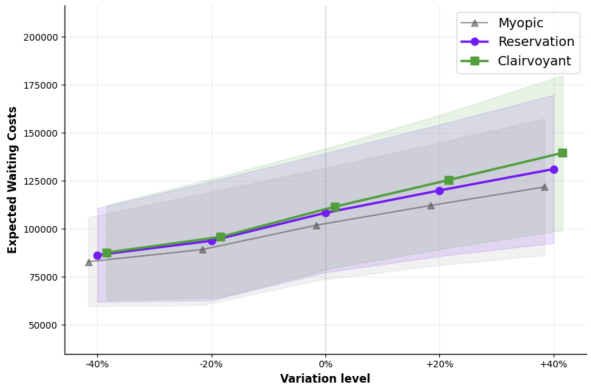
High-priority Arrival Mean



(a) Demand Scenario Distribution



(b) High-Priority Expected Waiting Costs



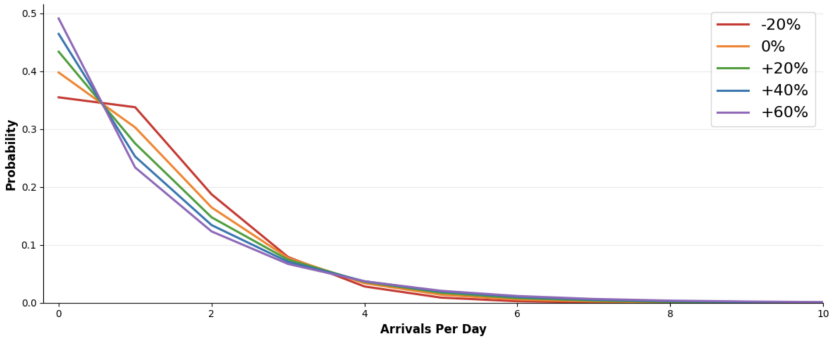
(c) Low-Priority Expected Waiting Costs

Figure 5.8: Out-of-Sample Performance Under High-Priority Arrival Mean Perturbations

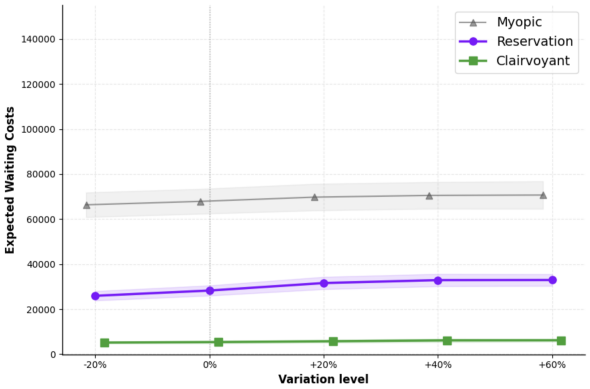
Figure 5.8 presents the sensitivity of expected waiting costs to perturbations in the high-priority arrival mean. Figure 5.8a presents the perturbations in arrival distributions as high-priority mean varies. From Figure 5.8b we observe that, as the high-priority arrival mean increases, the total expected waiting costs increases very mildly for Clairvoyant, but aggressively at extreme variations for Reservation and Myopic. The change in costs for Clairvoyant across the variation levels are not significant enough as there is sufficient capacity for high-priority patients to fill before low-priority patients. The Myopic policy's high-priority costs increase aggressively at the extreme variation because the low-priority

patients increasingly consume near-term future capacity to lock future high-priority patients out. We observe a similar shape in the case of Reservation policy because the quality of the future reservations deteriorates with large perturbations, restricting future high-priority patients from consuming near-term capacity, however not as much as Myopic. From Figure 5.8c we observe that as the low-priority arrival mean increases, expected waiting costs increase for Clairvoyant policy because there is a smaller capacity share remaining after allocating all high-priority patients. The low-priority costs increase for Myopic and Reservation policies because an increasing amount of high-priority patients compete with low-priority patients, and as high-priority patients are scheduled before low-priority patients, the latter naturally faces higher delay.

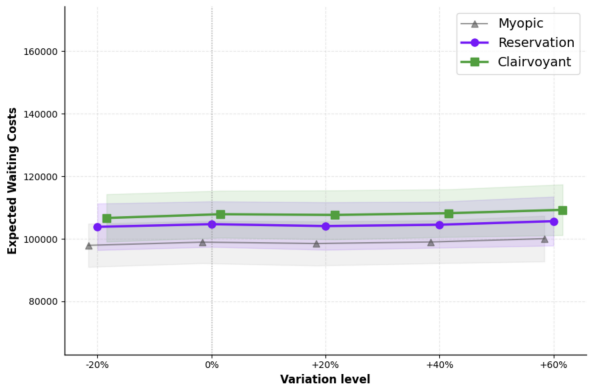
High-priority Arrival Variance



(a) Demand Scenario Distribution



(b) High-Priority Expected Waiting Costs



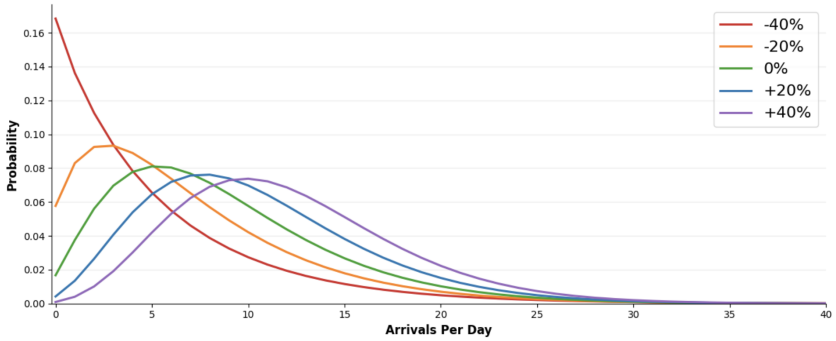
(c) Low-Priority Expected Waiting Costs

Figure 5.9: Out-of-Sample Performance Under High-Priority Arrival Variance Perturbations

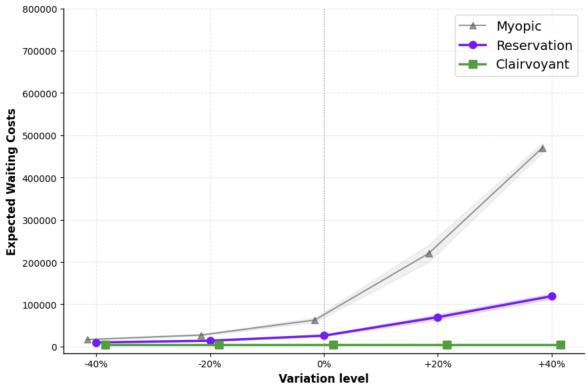
Figure 5.9 presents the sensitivity of expected waiting costs to perturbations in the high-priority arrival variance. Figure 5.9a presents the perturbations in arrival distributions as high-priority variance varies. From Figure 5.9b we observe that as the high-priority arrival variance increases, the total expected waiting costs increase very mildly for all the policies, of which Reservation policy shows the highest range. The biggest reason for relatively flat costs across variance perturbation is low baseline arrival variance and it is not significant enough to alter the capacity conditions that much. The Reservation policy's high-priority

costs, however, decreases between +20% to −20%, such that the decrease in costs is more aggressive than either benchmark policy. This is attributable to the fact that, as variance decreases, the realized demand concentrates more tightly around the mean, the forecast becomes increasingly accurate, and the protection levels produced by the re-solve program more precisely match the capacity that incoming high-priority patients will require. From Figure 5.9c we observe that as the low-priority arrival variance increase, expected waiting costs remain broadly flat across all three policies throughout the full variance range. This happens because high-priority variance perturbations do not alter the total volume of low-priority demand competing for OR capacity. Hence, the share of unprotected capacity available to low-priority patients remains unchanged in expectation, producing minimal impact on low-priority waiting costs under all three scheduling policies.

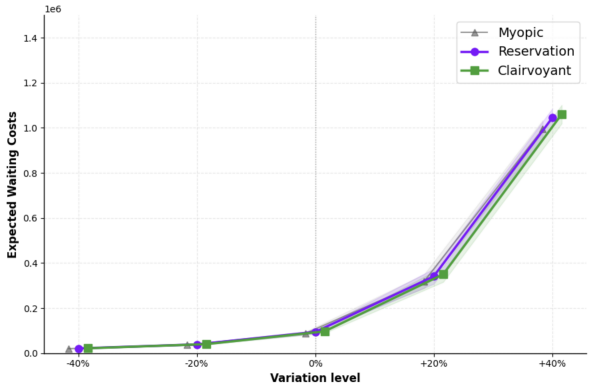
Low-priority Arrival Mean



(a) Demand Scenario Distribution



(b) High Priority Mean Waiting Costs



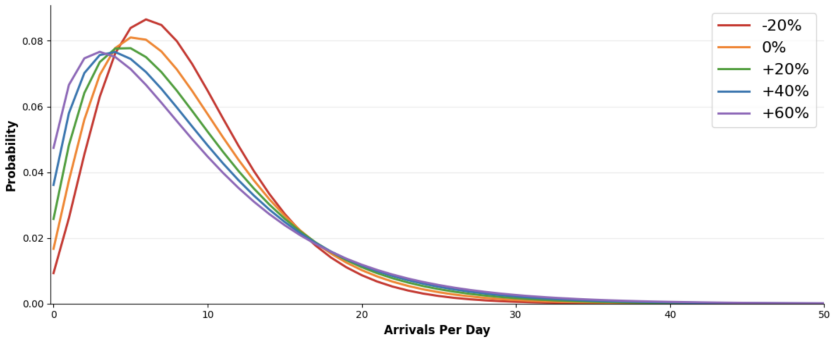
(c) Low Priority Mean Waiting Costs

Figure 5.10: Out-of-Sample Performance Under Low-Priority Arrival Mean Perturbations

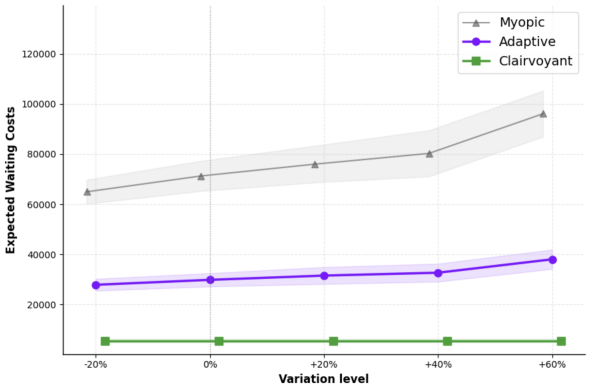
Figure 5.10 presents the sensitivity of expected waiting costs to perturbations in the low-priority arrival mean. Figure 5.10a presents the perturbations in arrival distributions as low-priority mean varies. From Figure 5.10b we observe that as the low-priority arrival mean increases, the Myopic and Reservation policies high-priority waiting costs increase non-linearly. This happens because, as the volume of low-priority demand grows, the patients consume a progressively larger share of near-term capacity under the Myopic policy, leaving insufficient capacity for future high-priority patients. This drives their wait times upward at an accelerating rate. The Reservation policy mitigates this by protecting

near-term capacity from low-priority patients through reservations. However as the demand increases, the future arrival forecasts become increasingly inaccurate, underestimating the protection levels for future high-priority patients. The Clairvoyant policy's high-priority costs remain flat across the full variation range because high-priority demand remains fixed across all variation levels and perfect foresight allows the offline integer program to protect capacity for high-priority patients in advance. For low-priority patients, Figure 5.10c shows that the waiting costs increase across all three policies. Overall, as the low-priority arrival mean grows, the system utilization ratio increases, driving a non-linear rise in expected waiting costs for both priority classes. It is important to note that for high-priority patients, the cost increase is driven by capacity restriction, whereas for low-priority patients, the cost increase is driven by the growth in demand volume itself.

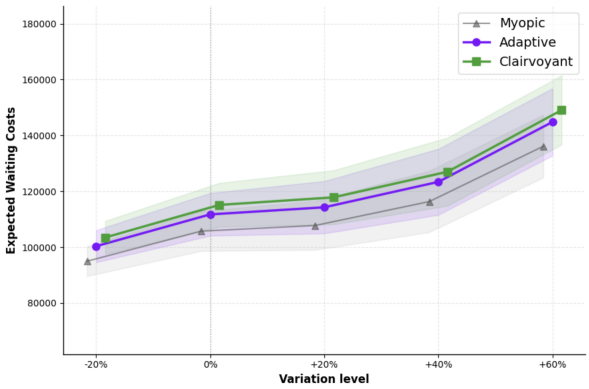
Low-priority Arrival Variance



(a) Demand Scenario Distribution



(b) High Priority Mean Waiting Costs



(c) Low Priority Mean Waiting Costs

Figure 5.11: Out-of-Sample Performance Under Low-Priority Arrival Variance Perturbations

Figure 5.11 presents the sensitivity of expected waiting costs to perturbations in the low-priority arrival variance. As the arrival variance increases, from Figure 5.11a, we observe that the tail gets heavier, suggesting higher probability of large demand spike. These demand surges consume a progressively larger share of near-term capacity, causing the high-priority waiting costs for Myopic and Reservation policies to increase in Figure 5.11b. The Reservation policy substantially mitigates this effect through reservations, the impact of which is most pronounced at extreme variance perturbations where the gap for high-priority

cost curves between Myopic and Reservation policies is the widest. The Clairvoyant policy's high-priority costs remain flat across the full variation range because perfect foresight allows the offline integer program to schedule high-priority demand in advance. For low-priority patients, Figure 5.11c shows that, waiting costs increase across all three policies as the arrival variance grows. Larger demand spikes due to high variability mean that on high-demand days the daily OR capacity cannot accommodate the full realized surge, causing a larger proportion of low-priority patients to be deferred to later periods. Given that the baseline low-priority arrival variance is already high relative to the mean, the system is particularly sensitive to upward perturbations in variance, explaining the increasing costs across variation levels.

5.2.4 Performance on Covariate-Based Demand Model

The baseline experiments presented in the preceding sections assume a stationary demand process, wherein arrivals for each priority class are drawn independently from a stationary negative binomial distribution. Real surgical demand often exhibits temporal structure such as day-of-week effects and seasonal trends. In this section, we test whether the Reservation policy performs differently on demand generated from a negative binomial regression model conditioned on calendar co-variates.

To address this, we fit separate negative binomial regression models for each priority level, regressing daily arrivals on day-of-week and month indicators. The high-priority regression yields a p-value of 0.120, with no individual covariate achieving significance at the $\alpha = 0.05$ level. This indicates that high-priority demand is empirically stationary with respect to these co-variates. The low-priority regression, by contrast, yields a highly significant overall fit with a p-value of 2.9×10^{-14} with positive day-of-week/month effects. Detailed results for the fit are presented in Table 5.5.

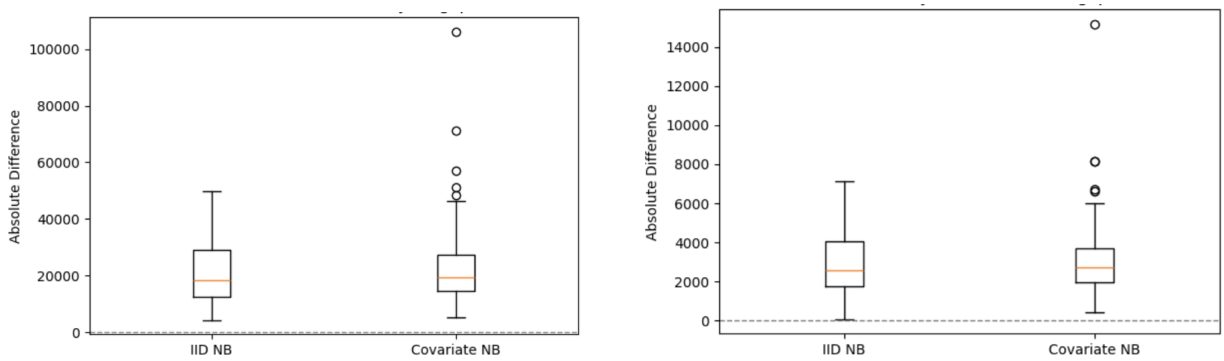
We use the fitted regression coefficients $\hat{\beta}$ to define an alternative demand generating process, wherein arrivals in period t are drawn from $\text{NB}(\exp(x_t'\hat{\beta}), \hat{r})$. And instead of using the unconditional sample mean $\bar{\mu}$ as the certainty equivalent estimate in earlier experiments, we use the conditional mean $\exp(x_t'\hat{\beta})$. Then, 100 independent demand paths are

Table 5.5: Negative binomial regression results for daily demand

	High Priority		*Low Priority*	
	Coef.	Std. Err.	Coef.	Std. Err.
Intercept	0.0390	0.166	2.1390***	0.097
Tue	0.1510	0.127	−0.0745	0.078
Wed	0.1249	0.128	0.0472	0.077
Thu	−0.1820	0.135	0.0149	0.078
Fri	−0.2107	0.136	−0.6438***	0.081
Feb	−0.2447	0.222	0.0683	0.124
Mar	0.0920	0.203	−0.0004	0.122
Apr	0.1257	0.205	0.1751	0.122
May	0.1082	0.202	0.0912	0.121
Jun	0.1933	0.202	0.1343	0.122
Jul	0.1168	0.204	0.1396	0.122
Aug	0.3199	0.196	0.0709	0.121
Sep	0.1488	0.204	0.2903**	0.121
Oct	0.1001	0.204	0.1509	0.121
Nov	0.0789	0.205	0.2668**	0.121
Dec	−0.0482	0.218	0.0040	0.128
Observations	773		773	
Log-Likelihood	−1,114.4		−2,361.0	
p-value	0.120		2.9×10^{-14}	
α	0.4344*** (0.079)		0.3574*** (0.027)	

Note: Reference categories are Monday (day-of-week) and January (month). Covariates are day-of-week and month indicators. High-priority demand exhibits no significant covariate structure. Low-priority demand exhibits a significant Friday effect and marginal seasonal effects in September and November. $^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$.

generated, and the same simulation mechanics applied as in the baseline experiments. To compare the performances, we measure iteration-level absolute cost difference for a demand path as the difference between the total waiting costs of Reservation and Clairvoyant policies. In essence, we compute the IID and Covariate NB policy cost difference and evaluate how the distributions of the absolute differences differ from each other. A visualization of the distribution for both demand models is presented in Figure 5.12.



(a) High priority

(b) Low priority

Figure 5.12: Distribution of Absolute Cost Differences between Reservation and Clairvoyant Policies under IID and Covariate-based Negative Binomial Demand

Since the demand generating process in both the experiments is different, we used the Mann–Whitney U test and Welch's two-sample t-test to test whether the distributions differed between the two demand models. We also computed the median and mean absolute cost difference between both demand models and report 95% confidence intervals. The results are presented in Table 5.6.

Neither priority class showed a statistically significant difference in distributions. For high-priority patients, the median absolute difference between covariate and stationary demand model is 1,218 with a 95% CI of $[-2{,}016,\ 4{,}788]$. For low-priority patients, the median absolute difference between both the models is 114 with 95% CI of $[-342,\ 690]$. Moreover, the statistical tests show that neither difference is statistically significant, such that Mann-Whitney $p = 0.421$ and Welch $p = 0.201$ for high-priority, and Mann-Whitney $p = 0.621$ and Welch $p = 0.321$ for low-priority. These results admit a clear interpretation.

Metric	High Priority	Low Priority
Median absolute difference between Covariate and IID		
Absolute Difference	1,218	114
95% CI	[−2,016, 4,788]	[−342, 690]
Mean absolute difference between Covariate and IID		
Absolute Difference	2,242.80	241.68
95% CI	[−1,113.86, 5,725.50]	[−215.90, 721.21]
Hypothesis tests (H_0: distributions are equal under IID and covariate demand)		
Mann-Whitney U	5,330	5,203
Mann-Whitney p-value	0.421	0.621
Welch t-statistic	1.282	0.995
Welch p-value	0.201	0.321
Reject H_0 ($\alpha = 0.05$)	No	No

Table 5.6: Statistical Tests: IID vs. Covariate-based NB Demand Model

For high-priority demand, the co-variates did not significantly improve model fit, hence the absence of a difference is expected. Although we observed statistically detectable day-of-week and month effects for low-priority demand, these effects were modest in magnitude and did not translate into a statistically significant change in the cost difference. Overall, our findings suggest that the performance of the Reservation policy is not affected when we add calendar variation to the demand model. This experiment could be further extended by considering more demand paths or testing it on arrival data that shows stronger calendar effects.

Chapter 6

Conclusion

In this thesis, we considered an advance patient scheduling problem where patients of two priority levels arrive randomly over a planning horizon. We formulated the problem as a stochastic dynamic program; however, solving it exactly suffers from the curse of dimensionality, and it becomes extremely challenging to build optimal scheduling policies. We tackle this by proposing a control algorithm, wherein we approximate the optimal policy by replacing the future arrival uncertainty with expected values and solving an approximation program that models the system dynamics. The solution from the program is used to instantiate policy parameters that act as reservations for high-priority patients and help shape future scheduling decisions. As the demand realizes, the existing reservations are dynamically updated until the next re-solve, where fresh reservations are computed over updated remaining capacities.

We validate the framework through a case study using arrival data from the hospital. We evaluate the performance of our proposed Reservation policy and compare it against two policy benchmarks – Myopic and Clairvoyant. Firstly, the Reservation policy registered a significant improvement in 90^{th} percentile high-priority wait time with a marginal increase in 90^{th} percentile low-priority wait time. Secondly, numerical experiments on operational and demand parameters across the three policies reveal a natural tradeoff between high- and low-priority waiting costs. The Clairvoyant policy, with perfect knowledge of future

arrivals, reserves capacity for high-priority patients in advance, minimizing their waiting costs at the expense of low-priority patients. Whereas, the Myopic policy schedules greedily without anticipating future high-priority demand, achieving lower low-priority waiting costs at the expense of high-priority patients. Our results show that the Reservation policy consistently outperforms the Myopic benchmark across all experiments. Out-of-sample experiments on demand volume demonstrate that the impact on waiting costs is most pronounced under perturbations to the low-priority mean arrival rate. An analysis of the re-solve interval shows that more frequent re-solving is helpful and allows the system to adapt to realized demand fluctuations and reduce waiting cost accumulation.

Several directions present natural extensions of this work. First, this work only considers a two-priority scheduling problem. Extending the model to an arbitrary number of priority levels would generalize the framework beyond the two-class setting. Second, incorporating overtime into the objective function could offer a more realistic cost structure faced by providers in practice. Third, the solve interval in the current framework is treated as a fixed parameter throughout the planning horizon. An interesting extension would be to determine optimal re-solve points based on realized system state, potentially reducing re-optimizations while preserving performance gains. Fourth, we tested the performance of the Reservation policy after incorporating calendar effects, such as day-of-week or seasonal trends, into the simulation. On the hospital data, calendar effects were modest and did not significantly change the absolute difference between Reservation and Clairvoyant costs. Future work could evaluate the same framework on datasets with stronger calendar seasonality.

References

Zahraa A. Abdalkareem, Amiza Amir, Mohammed Azmi Al-Betar, Phaklen Ekhan, and Abdelaziz I. Hammouri. Healthcare scheduling in optimization context: a review. *Health and Technology*, 11(3):445–469, 2021. ISSN 2190-7188.

Anders Reenberg Andersen, Thomas Jacob Riis Stidsen, and Line Blander Reinhardt. Simulation-based rolling horizon scheduling for operating theatres. *SN Operations Research Forum*, 1(2):9, 2020.

Roberto Aringhieri, Paolo Landa, Patrick Soriano, Elena Tànfani, and Angela Testi. A two level metaheuristic for the operating room scheduling and assignment problem. *Computers & Operations Research*, 54:21–34, February 2015. ISSN 0305-0548.

Davood Astaraky and Jonathan Patrick. A simulation based approximate dynamic programming approach to multi-class, multi-resource surgical scheduling. *European Journal of Operational Research*, 245(1):309–319, August 2015. ISSN 0377-2217.

Nur Ayvaz and Woonghee Tim Huh. Allocation of hospital capacity to multiple types of patients. *Journal of Revenue and Pricing Management*, 9(5):386–398, 2010.

Javiera Barrera, Rodrigo A. Carrasco, Susana Mondschein, Gianpiero Canessa, and David Rojas-Zalazar. Operating room scheduling under waiting time constraints: the Chilean GES plan. *Annals of Operations Research*, 286(1-2):501–527, 2020.

Corina Bello, Richard D. Urman, Lukas Andereggen, Dietrich Doll, and Markus M. Luedi. Operational and strategic decision making in the perioperative setting: Meeting bud-

getary challenges and quality of care goals. *Best Practice Research Clinical Anaesthesiology*, 36(2):265–273, 2022. ISSN 1521-6896.

Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 1. Athena Scientific, Nashua, NH, 4th edition, 2017. ISBN 978-1-886529-44-1.

Pornpawee Bumpensanti and He Wang. A re-solving heuristic with uniformly bounded loss for network revenue management. *Management Science*, 66(7):2993–3009, 2020.

Brecht Cardoen, Erik Demeulemeester, and Jeroen Beliën. Operating room planning and scheduling: A literature review. *European Journal of Operational Research*, 201(3):921–932, 2010.

Frank Chuang, Claus Danielson, and Francesco Borrelli. Optimality of certainty equivalence in expected value problems for uncertain linear systems. In *2014 IEEE 53rd Annual Conference on Decision and Control (CDC)*, pages 5555–5560. IEEE, 2014.

Dragos Florin Ciocan and Vivek Farias. Model predictive control for dynamic resource allocation. *Mathematics of Operations Research*, 37(3):501–525, 2012.

William L. Cooper. Asymptotic behavior of an allocation policy for revenue management. *Operations Research*, 50(4):720–727, 2002.

Yigal Gerchak, Diwakar Gupta, and Mordechai Henig. Reservation planning for elective surgery under uncertain demand for emergency surgery. *Management Science*, 42(3): 321–334, 1996.

Paul Glasserman and David D. Yao. Some guidelines and guarantees for common random numbers. 38:884–908, 1992. ISSN 0025-1909.

Lukas Glomb, Frauke Liers, and Florian Rösel. A rolling-horizon approach for multi-period optimization. *European Journal of Operational Research*, 300(1):189–206, 2022. ISSN 0377-2217.

Yasin Gocgun. Simulation-based approximate policy iteration for dynamic patient scheduling for radiation therapy. *Health Care Management Science*, 21(3):317–325, 2018.

Yasin Gocgun and Martin L. Puterman. Dynamic scheduling with due dates and time windows: an application to chemotherapy patient appointment booking. *Health Care Management Science*, 17(1):60–76, 2014.

Stefanus Jasin and Sunil Kumar. A re-solving heuristic with bounded revenue loss for network revenue management with customer choice. *Mathematics of Operations Research*, 37(2):313–345, 2012.

Esmaeil Keyvanshokooh, Cong Shi, and Mark P. Van Oyen. Online Advance Scheduling with Overtime: A Primal-Dual Approach. *Manufacturing & Service Operations Management*, 23(1):246–266, January 2021. ISSN 1523-4614.

Esmaeil Keyvanshokooh, Pooyan Kazemian, Mohammad Fattahi, and Mark P. Van Oyen. Coordinated and priority-based surgical care: an integrated distributionally robust stochastic optimization approach. *Production and Operations Management*, 31(4):1510–1535, 2022.

James M Magerlein and James B Martin. Surgical demand scheduling: a review. *Health Services Research*, 13(4):418–433, 1978.

Daiki Min and Yuehwern Yih. An elective surgery scheduling problem considering patient priority. *Computers & Operations Research*, 37(6):1091–1099, 2010.

Ontario Health. Surgery data standardization guide, version 2.0, Dec 2020.

Ontario Health. Wait times results, 2025.

Ontario Hospital Association. Quality-based procedures (qbp): What you need to know. Technical report, OHA, 2018.

Ontario Ministry of Health and Long-Term Care. Quality-based procedures: Overview and resources. Technical report, Government of Ontario, 2012.

Jonathan Patrick, Martin L. Puterman, and Maurice Queyranne. Dynamic multipriority patient scheduling for a diagnostic resource. *Operations Research*, 56(6):1507–1525, 2008. ISSN 0030-364X.

Michael L. Pinedo. *Scheduling: Theory, Algorithms, and Systems*. Springer Cham, 6 edition, 2022. ISBN 978-3-031-05920-9.

Warrren B Powell. Reinforcement learning and stochastic optimization. reinforcement learning and stochastic optimization: A unified framework for sequential decisions (2022), 2022.

Martin I Reiman and Qiong Wang. An asymptotically optimal policy for a quantity-based network revenue management problem. *Mathematics of Operations Research*, 33(2): 257–282, 2008.

Antoine Sauré, Mehmet A. Begen, and Jonathan Patrick. Dynamic multi-priority, multi-class patient scheduling with stochastic service times. *European Journal of Operational Research*, 280(1):254–265, January 2020. ISSN 0377-2217.

Nicola Secomandi. An analysis of the control-algorithm re-solving issue in inventory and revenue management. *Manufacturing Service Operations Management*, 10(3):468–483, Summer 2008.

Herbert A. Simon. Dynamic programming under uncertainty with a quadratic criterion function. *Econometrica*, 24(1):74–81, 1956. ISSN 0012-9682.

Clifford Stein, Van-Anh Truong, and Xinshang Wang. Advance Service Reservations with Heterogeneous Customers. *Management Science*, 66(7):2929–2950, July 2020. ISSN 0025-1909.

Kalyan Talluri and Garrett van Ryzin. An analysis of bid-price controls for network revenue management. *Management Science*, 44(11-part-1):1577–1593, 1998.

Angela Testi, Elena Tanfani, and Giancarlo Torre. A three-phase approach for operating theatre schedules. *Health Care Management Science*, 10(2):163–172, June 2007. ISSN 1572-9389.

Henri Theil. A note on certainty equivalence in dynamic planning. *Econometrica*, 25(2): 346–349, 1957.

Van-Anh Truong. Optimal Advance Scheduling. *Management Science*, 61(7):1584–1597, July 2015. ISSN 0025-1909.

Elena Tànfani and Angela Testi. A pre-assignment heuristic algorithm for the Master Surgical Schedule Problem (MSSP). *Annals of Operations Research*, 178(1):105–119, July 2010. ISSN 1572-9338.

Carla Van Riet and Erik Demeulemeester. Trade-offs in operating room planning for electives and emergencies: A review. *Operations Research for Health Care*, 7:52–69, December 2015. ISSN 2211-6923.

Xinshang Wang and Van-Anh Truong. Multi-priority online scheduling with cancellations. *Operations Research*, 66(1):104–122, 2018.

Xinshang Wang, Van-Anh Truong, and David Bank. Online Advance Admission Scheduling for Services with Customer Preferences, May 2018. arXiv:1805.10412 [math].

Kayla Wiebe, Simon Kelley, and Roxanne E. Kirsch. Revisiting the concept of urgency in surgical prioritization and addressing backlogs in elective surgery provision. *CMAJ*, 194 (29):E1037–E1039, 2022.

Christos Zacharias, Nan Liu, and Mehmet A. Begen. Dynamic Interday and Intraday Scheduling. *Operations Research*, 72(1):317–335, January 2024. ISSN 0030-364X.

Shuwan Zhu, Wenjuan Fan, Shanlin Yang, Jun Pei, and Panos M. Pardalos. Operating room planning and surgical case scheduling: a review of literature. *Journal of Combinatorial Optimization*, 37(3):757–805, April 2019. ISSN 1573-2886.

APPENDICES

Appendix A

Notation

A.1 Sets

Notation	Description
P	Set of patient priority classes
$\mathcal{K}$	Set of re-solve periods, $\mathcal{K} = \{1, N+1, 2N+1, \ldots\}$
$\mathcal{T}_k$	Set of interval periods associated with re-solve period $k \in \mathcal{K}$, $\mathcal{T}_k = \{k+1, \ldots, k+N-1\}$
$\mathcal{C}_t$	Feasible domain set for the capacity state $\mathbf{c}(t)$ at period t
$\mathcal{Q}_t^{(p)}$	Feasible domain set for the queue state $\mathbf{q}^{(p)}(t)$ of priority class $p \in P$ at period t
$\mathcal{Y}_t(\cdot)$	Feasible action set for the true (stochastic) system at period t, defined by (3.4)–(3.5)
$\tilde{\mathcal{Y}}_t(\cdot)$	Feasible action set for the certainty-equivalent re-solve program $\mathcal{RP}_k$ at period t, defined by (4.1)–(4.2)

Notation	Description
$\mathcal{Y}_\tau^{(1)}(\cdot)$	Feasible action set for the inner (high-priority) interval program $\mathcal{IP}_\tau^{(1)}$ at period τ
$\mathcal{Y}_\tau^{(2)}(\cdot)$	Feasible action set for the outer (low-priority) interval program at period τ

A.2 Decision Variables

Notation	Description
$x_{t_1,t_2}^{(p)}(t)$	Number of priority-p patients, arriving on day t_1, scheduled to surgery day t_2, under a decision made at period t
$\mathbf{x}^{(1)}(t)$	High-priority decision vector at period t
$\mathbf{x}^{(2)}(t)$	Low-priority decision vector at period t
$\mathbf{x}(t)$	Joint decision vector at period t, $\mathbf{x}(t) = (\mathbf{x}^{(1)}(t), \mathbf{x}^{(2)}(t))$
$\mathbf{X}$	Schedule matrix accumulating realized scheduling decisions, with columns indexed by $t \in [T]$
$\tilde{x}_{t_1,t_2}^{(p)}(t)$	Certainty-equivalent decision variable, produced by the re-solve program $\mathcal{RP}_k$
$\tilde{\mathbf{X}}$	Full solution matrix returned by $\mathcal{RP}_k$, with columns indexed by $t \in \{k, \ldots, T\}$

A.3 State Variables

Notation	Description
$c_{t_2}(t)$	Available service capacity on day t_2, as observed at the beginning of period t
$\mathbf{c}(t)$	Capacity state vector at period t
$q_{t_1}^{(p)}(t)$	Number of unscheduled priority-p patients who arrived on day t_1 and remain waiting at the beginning of period t
$\mathbf{q}^{(1)}(t)$	High-priority queue state vector at period t
$\mathbf{q}^{(2)}(t)$	Low-priority queue state vector at period t
$\tilde{c}_{t_2}(t)$	Certainty-equivalent capacity state on day t_2, as evolved under the re-solve program $\mathcal{RP}_k$
$\tilde{q}_{t_1}^{(p)}(t)$	Certainty-equivalent queue state for priority p, arrival day t_1, as evolved under $\mathcal{RP}_k$

A.4 Parameters

Notation	Description
T	Total number of days in the planning horizon
B	Booking horizon
N	Solve interval
s_p	Deterministic service duration (hours) for a patient of priority class $p \in P$
w_p	Per-period waiting cost incurred for each unscheduled patient of priority $p \in P$, with $w_1 > w_2$
$D_{t_1}^{(p)}$	Realized number of priority-p patients arriving on day t_1

Notation	Description
$\hat{D}_{t_1}^{(p)}$	Certainty-equivalent estimate of future random demand for priority-p and arrival day t_1
$\tilde{\mathbf{X}}'$	Active high-priority reservation profile carried forward into the interval periods, obtained from $\tilde{\mathbf{X}}$ via the update rule (4.7)

Appendix B

Fitting Demand Distributions for High- and Low-Priority Patients

Figure B.1 compares the fitted Poisson and negative binomial (NB) distributions against observed daily surgical arrivals for each priority class. In both cases, the Poisson distribution fits poorly due to overdispersion. This is corroborated by Table B.2, where the NB achieves substantially lower AIC and BIC values and the likelihood ratio test rejects the null at $p < 0.001$ for both priority classes. Accordingly, NB distributions are adopted to model daily surgical arrivals in the numerical experiments.

The parameters of the Negative Binomial distribution are generated as follows. The success probability q and dispersion parameter r are chosen such that

$$\mathbb{E}[D^{(p)}] = \mu_p = \frac{r(1-q)}{q}, \qquad \mathrm{Var}(D^{(p)}) = \sigma_p^2 = \frac{r(1-q)}{q^2}.$$

Dividing the variance equation by the mean equation yields

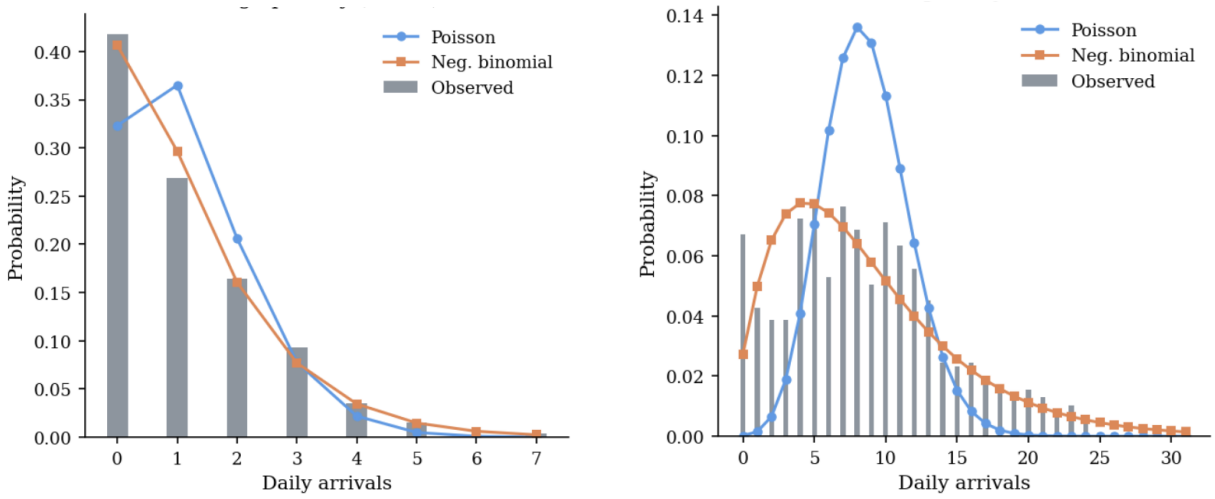
$$\frac{1}{q} = \frac{\sigma_p^2}{\mu_p} \quad \Longrightarrow \quad q = \frac{\mu_p}{\sigma_p^2}.$$

Substituting this expression into the mean equation gives

$$r = \frac{\mu_p^2}{\sigma_p^2 - \mu_p}.$$

Thus, for each priority class p, we parameterize the Negative Binomial distribution as

$$q = \frac{\mu_p}{\sigma_p^2}, \qquad r = \frac{\mu_p^2}{\sigma_p^2 - \mu_p}.$$



(a) High-priority demand fit

(b) Low-priority demand fit

Figure B.1: Fitted demand distributions for high- and low-priority patients

Priority level	**Mean**	**Variance**	**Dispersion index**
High priority	1.129	1.675	1.483
Low priority	8.655	33.327	3.851

Table B.1: Descriptive statistics of daily surgical arrivals.

Table B.2: Poisson versus negative binomial model comparison for surgical arrival distributions.

	AIC		**BIC**		**LRT**
Priority level	*Poisson*	*NB*	*Poisson*	*NB*	*p-value*
High priority	2320.49	2254.38	2325.14	2263.68	<0.001
Low priority	6079.30	4824.14	6083.95	4833.44	<0.001