

Data-driven Models for Inferring the Patient Scheduling Policies via Inverse Reinforcement Learning

by

Parham Moradi

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, 2023

© Parham Moradi 2023

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

In this work, we study multi-class patient scheduling with stochastic daily patient arrivals. Different classes of patients are characterized by different service times, waiting cost parameters, and rejection cost parameters. Our primary objective is to infer the policy used by the decision-makers, who schedule patients over a finite time horizon, based on their historical decisions. To achieve this, we first develop a mathematical model that captures the complexities of patient scheduling and is representative of the problem that decision-makers may consider to scheduling patients. Then, we utilize the Riccati and Hamiltonian approaches to estimate the cost parameters that have influenced the scheduling decisions made by the decision-maker. The Riccati approach begins by estimating the expert’s policy, which is then used to determine the cost parameters. Conversely, the Hamiltonian approach derives the cost parameters through the optimality conditions of a path trajectory without needing to estimate the expert’s policy. Using a simulation model, we demonstrate the efficiency and robustness of the proposed methods. Furthermore, we apply Riccati and Hamiltonian approaches to MRI data from two hospitals to estimate the cost parameters used in their scheduling decisions. Utilizing the estimated cost parameters, we analyze the root causes of the observed outcomes and examine the impact of these underlying factors on the scheduling process. Finally, through counterfactual analysis, we propose two alternative scheduling policies that reduce the total cost, even with the original cost parameters used by the decision-makers.

Acknowledgements

I would like to thank my supervisor, Prof. Hossein Abouee Mehrizi, for his guidance, relentless support, and patience. His insightful suggestions were crucial to the success of this work, and words cannot fully express my gratitude to him.

My special thanks to my friend and unofficial supervisor Prof. Mohamad Kazem Shirani Faradonbeh for his endless support, encouragement, special caring, and expertise. I would like to thank the readers of this thesis, Prof. Houra Mahmoudzadeh and Prof. Saeed Ghadimi, for their valuable comments and guidance along this way.

I would like to express my appreciation to all of my family. My parents, my sister, and my aunts were unwavering supportive throughout this journey. Setareh, my fiancée, has always been a source of true and unconditional love, and she has always been someone I can rely on. I couldn't be happier with her.

Table of Contents

Author’s Declaration	ii
Abstract	iii
Acknowledgements	iv
List of Figures	ix
List of Tables	x
1 Introduction	1
2 Literature Review	5
2.1 Inverse Reinforcement Learning	5
2.2 Advance Scheduling	8
3 Problem Formulation	10
3.1 Definitions of Variables	11

3.2	Cost Function	12
3.3	System Dynamics	14
3.4	Advance Scheduling with Noise in Observations	17
4	Riccati Approach	19
4.1	Estimating Action Matrices	22
4.2	Scalar Ambiguity in Cost Function	24
4.3	Riccati Gain Matrix	25
4.4	Estimation of Rejection Cost Parameters	26
4.4.1	State-Action Weight Matrix	26
4.4.2	Demand-Action Weight Matrix	28
4.5	Estimation of Waiting Cost Parameters	29
4.5.1	Time-Invariant Action Component	30
5	Hamiltonian Approach	33
6	Numerical Study	38
6.1	Performance of the Proposed Approaches	38
6.1.1	Sensitivity Analysis on Time Horizon	39
6.1.2	Sensitivity Analysis on Time Horizon with Noisy Dynamics	42
6.1.3	Sensitivity Analysis on Booking Horizon	43
6.1.4	Sensitivity Analysis on Booking Horizon with Noisy Dynamics	45

6.2 Case Study	46
6.3 Counterfactual Analysis	55
7 Concluding Remarks	59
References	61
APPENDICES	65
A Proofs	66
A.1 Proof of Lemma (1)	66
A.2 Proof of Proposition 4.1.1	69
A.3 Proof of Proposition 4.4.1	71
A.4 Proof of Lemma 2	75
A.5 Proof of Lemma 3	76
A.6 Proof of Proposition 4.4.2	77
A.7 Proof Proposition 4.5.1	79
A.8 Proof of Proposition 5.0.1	82
A.9 Detailed technical reformulation of the instantaneous cost function in (4.1)	84
A.10 Details on optimality of a path trajectory	87
B Matrix Representations	90
B.0.1 Section State-Action Weight Matrix	90

B.0.2	Section Demand-Action Weight Matrix	92
B.0.3	Hamiltonian Approach	93
C	More Details on the Hamiltonian Approach	98
D	More Details on the Riccati Approach	101

List of Figures

6.1	Estimation Error and the Time Horizon	41
6.2	Estimation Error and Time Horizon with Noisy Dynamics	43
6.3	Estimation Error Relative to the Booking Horizon	44
6.4	Estimation Error and Booking Horizon with Noisy Dynamics	46
6.5	Demand Percentages of Priority Levels at Hospital 1 in (A) and Hospital 2 in (B)	47
6.6	Daily Distribution of Patients Arrivals	47
6.7	Daily Distribution of Patients Receiving Services	49
6.8	Distribution of Waiting days	50

List of Tables

6.1	Average Waiting Days for Hospitals 1 and 2	51
6.2	Rejected Patients Based on Priorities in Hospital 1 and 2	52
6.3	Rejection and Waiting Costs in Hospitals 1 and 2	53
6.4	Comparison of Percentage Improvement in Total Costs for Hospitals 1 and 2 Using Ricatti and Hamiltonian Approaches	58

Chapter 1

Introduction

Due to the COVID-19 pandemic, more patients are experiencing extended waiting periods for medical appointments. In Ontario, over 100,000 people are waiting longer than the advised time for surgeries during the pandemic ¹. Additionally, Ontario is dealing with a big backlog of around one million surgeries because of the pandemic ². In response to this situation, healthcare providers are required to optimize patient scheduling to improve the system's efficiency and reduce the wait time. Patient scheduling is a challenging task considering the limited resources available and the unpredictable nature of daily patient arrivals. The objective of this study is to develop a data-driven model based on historical data to first understand how patients are currently being scheduled and then provide insights into how the procedure of patient scheduling can be improved so that the throughput of the system is increased.

Hospitals operate under constant pressure, striving to be as busy as possible without

¹[1], <https://globalnews.ca/news/9329715/ontario-long-waiters-pandemic-surgery/>

²[16], <https://bit.ly/ontario-surgery-health-care-backlog-medical-association>

wasting resources³. The stochastic nature of patient arrivals adds a layer of unpredictability to this challenge, further emphasizing the need for an efficient and responsive scheduling system.

According to Section 2.2, numerous studies focus on patient scheduling in hospitals, such as [4, 20] and [24]. Yet, there is a gap in understanding how decision-makers actually schedule patients in real-world settings. These studies, while addressing advance scheduling, often independently set optimization problems and assign some weights to different factors concerning the utilization of the system.

Inverse Reinforcement Learning is suitable for scenarios where the reward or cost function is unknown, but decision-maker’s demonstrations are available. Analysis of decision-maker behaviour can yield insights into their scheduling strategies. For instance, a decision-maker might believe they prioritize certain patient groups, but the reality could be different. We can quantify the daily waiting cost parameters for different patient groups to understand the level of prioritization for each group.

In this research, we study the problem of inverse advance scheduling, focusing on multi-class patients. Each day, a stochastic number of patients request service, termed as the daily demand. The service provider schedules these patients within a booking horizon, which we assume is based on a specific cost function considered in a finite horizon setting. While we can observe the decisions made by the service provider, the specific cost function guiding these decisions remains unknown. Our goal is to identify this cost function utilized by the decision maker, by analyzing their decisions and the corresponding daily demand.

In the first step, we carefully model the cost function that the decision-makers consider when scheduling patients. We assume that the objective function of the problem consists

³[17], <https://bit.ly/waiting-your-turn-wait-times-for-health-care-in-canada-2022>

of three main cost components. The idle time cost rises quadratically in both unused daily capacity and over-scheduled daily capacity. Waiting cost, on the other hand, grows linearly with the number of days patients wait to receive the service, varying for each class of patients. Lastly, rejection cost increases quadratically with the daily number of patients rejected, with each class having its own rejection cost coefficient.

We analyze the proposed inverse problem using two approaches, namely the Riccati approach and the Hamiltonian approach. The Riccati approach begins by estimating the expert’s policy and then uses this policy to determine the cost parameters. In contrast, the Hamiltonian approach employs a different strategy for solving the inverse problem. It estimates the cost parameters based on the optimality conditions of a path trajectory. We show that the Riccati and Hamiltonian approaches can be used in both noiseless and noisy environments to estimate cost parameters. We prove the uniqueness of cost parameters for a given set of actions and demands. Based on extensive numerical experiments, we demonstrate the effectiveness of the proposed approaches under various conditions where different parameters including the booking horizon, time horizon, and uncertainty in the system dynamics vary. As the time horizon approaches infinity, we observe that our cost parameter estimations approach the exact values.

Using patient-level data from two hospitals, we estimate the cost parameters utilized to schedule patients and observe a correlation between waiting cost parameters and average waiting day. Specifically, we observe that classes with fewer average waiting times typically have higher waiting cost parameters, indicating an inverse proportionality. Additionally, we analyze how the decision maker balances cost parameters, such as timely patient care prioritization versus minimizing overtime capacity. We also examine the scheduling priorities of the decision-makers and explore differences between the two hospitals’ policies. Additionally, we provide constructive feedback to hospitals, helping them improve their

performance and decrease the average waiting time for different classes of patients.

Moreover, in the counterfactual analysis section, we introduce two alternative scheduling policies, each showing cost advantages over the decision-maker’s policy. Under identical demand arrivals, our proposed methods outperform the decision-maker’s choice in terms of the total cost of the system. To calculate this total cost, we use the inferred cost parameters that the decision-maker employed to schedule patients. One of the proposed alternative scheduling policies is derived from the policy learned from the decision maker’s actions via the Riccati approach, and the other is based on the cost parameters estimated using the Hamiltonian approach. While the decision-maker occasionally deviates from the utilized cost parameters and alters prioritization levels, our approaches maintain consistency in these aspects, thereby reducing total costs. These proposed policies could replace the decision maker’s role, generating improved actions based on the inferred cost parameters of the decision maker, given the demand and the system’s state.

Chapter 2

Literature Review

In this study, with the aim of inverse reinforcement learning, we analyze the process of patient scheduling in a setting with a central decision-maker who schedules patients over a finite horizon. Thus, this work is related to the papers in the literature on Inverse Reinforcement Learning and Advance Scheduling, which are explored in sections [2.1](#) and [2.2](#).

2.1 Inverse Reinforcement Learning

According to [\[5\]](#), imitation learning is defined as the process of learning from the example set by an expert or a teacher. [\[5\]](#) describes Inverse Reinforcement Learning (IRL) as a particular form of imitation learning, which focuses on estimating the cost or reward function from the expert's demonstrations. Inverse optimal control (IOC), and inverse optimization are two related concepts to IRL. IOC and IRL have merged into one field, previously having distinct focuses with IOC centred on system stability and IRL concerned

with the optimality of actions, as explained in [5].

Both IRL and IOC are parts of the Inverse Optimization framework. Based on observed solutions or outcomes, Inverse Optimization addresses the task of inferring the parameters or objectives of an optimization problem. Inverse Optimization techniques are useful when the objective function or constraints of the problem are unknown, but its solutions or decisions are observable. The goal here is to find the underlying optimization problem that generates the observed solutions.

IRL is closely related to Reinforcement Learning (RL). According to [29], RL involves an agent developing a policy to interact with its environment, given a reward function. Specifying a reward function for an RL agent can be challenging. A solution to this is IRL, which can be viewed as an extension of RL in this environment, focused on learning a reward function from an expert’s demonstrations. [8] uses an external trainer to instruct new agents through inferring reward function.

When inferring an agent’s cost function, it may be necessary to consider the format of the function, which can vary from linear and quadratic to other types of functions. [3] assumes that the cost function can be expressed as a linear function of known features and focuses on finding the weights of each feature. [19] assumes a quadratic cost function and adapts the recovery matrix approach to fit the observed trajectory, thereby successfully recovering the cost function. [30] also models the cost function linearly but assumes the expert’s actions are stochastic. They use an algorithm to uncover the underlying reward via maximum entropy. Despite its usefulness for determining cost parameters, the method does not guarantee the uniqueness or existence of these parameters. In contrast, our study demonstrates that in the absence of noise, a unique solution up to a scalar ambiguity exists for the advance scheduling problem and can be successfully obtained.

A commonly explored scenario in various papers involves the system dynamics being represented as a linear combination of states and actions, accompanied by a cost function that is quadratic in nature. This framework is known as Linear Quadratic Regulators (LQR). [28] consider quadratic regulators. They make a strong assumption that the quadratic multiplier for expert actions is known. Under this assumption, they illustrate that cost parameters are unique and identifiable. Conversely, [27] extends the study of [28], relaxing the necessity of knowing the expert’s action quadratic multiplier and successfully determining a unique cost function up to a scalar ambiguity.

Our research problem is closely connected to the aforementioned papers; however, it differs in terms of its specific focus. We aim to apply the IOC and IRL methods to infer the cost parameters in an advance scheduling context. Using IOC approaches, we aim to demonstrate that each policy is uniquely generated by a specific set of cost parameters and estimate these parameters. In [15], another application of IRL is mentioned which is intent prediction and trajectory forecasting. Specifically, they consider a setting where they observe human behaviour and make predictions about their current and future actions. They assume that the actor aims to minimize a quadratic cost function by selecting an action from a predefined set, through which the actor’s intentions are identified.

[13] employs maximum entropy to address the problem of Inverse Optimal Control (IOC). They also propose an optimality condition that involves a setting where the cost function has zero gradients in terms of the action. We employ a comparable technique, known as the Hamiltonian approach, to identify the underlying cost function, as detailed in Section 5.

In Chapter 3 of [14], thorough research is presented on inverse optimal control, highlighting optimization-based methods that solve the inverse problem. Within these approaches,

the Hamiltonian method is employed extensively, providing numerous references to various studies that have applied the Hamiltonian approach in different contexts. These include scenarios involving feedback-law-based approaches and truncated-sequence methods. [2] offers a comprehensive survey on inverse optimal control and inverse reinforcement learning, discussing the Hamiltonian approach as one of the primary methods for solving inverse optimal control problems.

2.2 Advance Scheduling

Our study concentrates on inter-day scheduling, differing from intra-day scheduling. In intra-day scheduling, patients receive appointments within the same day, with the primary objective of minimizing service wait times, as detailed in [9, 12], and [21]. In contrast, our focus on inter-day scheduling involves dynamically scheduling patients for future days.

This study investigates the use of inverse reinforcement learning in advance scheduling. The advance scheduling setting involves notifying patients of their appointments once they request the service. It contrasts with the allocation scheduling setting, in which service providers maintain a queue and add service requests to the queue as the requests for the service arrive. Under the allocation policy, patients are notified of their appointments on the scheduled day.

[25] is based on the advance scheduling setting with two main classes of patients, urgent and regular. Regular patients can wait for their service, whereas urgent ones require immediate service. Our study aligns with this categorization, classifying patients by service type and urgency. However, in this study, decision-makers have the option to schedule multiple classes of patients for future time periods. The literature on advance scheduling generally assumes stochastic demand. [20] and [24] utilize Markov decision processes as an

analytical tool. Additionally, [4] explores an asymptotic regime and develops a diffusion-optimal policy.

There are some applications of inverse reinforcement learning in different areas such as economics e.g. [2, 18] and autonomous driving e.g. [22, 26]. However, in the context of advance scheduling, to the best of our knowledge, no paper in the literature applies inverse reinforcement learning to understand how patients are being scheduled as their requests are received by the service provider.

Chapter 3

Problem Formulation

The main focus of this study is on advance multi-class patient scheduling. It begins with the assumption that service providers do not make arbitrary choices; rather, they adhere to some form of algorithmic thinking when allocating time slots to patients. This behavior is carefully modeled in [3.2](#) and based on the proposed optimization model we estimate the best weights that fit the decision-maker's algorithms to schedule patients. Essentially, our main objective is to infer the intentions of the decision-makers when they schedule patients. This allows us to offer constructive feedback on patient scheduling. We will use the decision maker, who schedules the patients, and the expert interchangeably since in this problem it is assumed the decision maker is an expert who is minimizing some cost function and our primary aim is to estimate the weights of their cost function.

3.1 Definitions of Variables

Our research aims to understand the decision-making behavior of service providers in a patient scheduling setting. We model this behavior by introducing an objective function, which decision-makers aim to minimize when making scheduling decisions. To estimate the parameters of this cost function, we employ Inverse Reinforcement Learning (IRL), which is an approach for inferring the underlying objective function based on observed decisions.

In the context of advance patient scheduling, applying IRL involves identifying the cost parameters governing scheduling decisions. These decisions are defined as the actions taken by the decision maker denoted by u . The variable $u_t^{i,n}$ denotes the number of patients of class i scheduled for day $t + n$ (receiving their service on day $t + n$), based on the action taken on day t .

On day t , patients can be scheduled for consecutive days $t, t + 1, t + 2, \dots, t + N - 1$. Here N defines the length of the booking window which is the time window within which patients can be scheduled. This period ranges from day 0 to $N - 1$ following the day of scheduling. This also imposes a constraint on the dimensions of the action vector u . Patients are categorized into I distinct classes due to their service type (e.g., brain, heart, and leg) and their emergency level (e.g., 1, 2, 3, and 4). The service providers are granted the daily capacity of $C_{\max}$ hours.

In the process of scheduling patients, decision-makers take into account the required duration of service for each class of patient, commonly referred to as service time. The service time vector is denoted by M where element M_i represents the estimated service duration for a patient belonging to class i . A stochastic number of patients from class i request services on the day t , indicated by $D_{i,t}$. We consider a finite horizon setting with T periods.

3.2 Cost Function

The cost function is carefully designed to reflect the complexity of scheduling patients. As previously stated, the goal is to provide service to patients as soon as possible. However, the process is complicated due to different levels of urgency among distinct classes of patients. Our approach involves breaking down the cost parameters associated with decision-makers into three primary components: waiting cost, rejection cost, and idle time cost.

The waiting time refers to the delay between patient arrival and their scheduled appointment. Patients' satisfaction and service quality will generally decline the longer they wait for service. The cost function is constructed such that the longer a patient has to wait for their service, the higher the incurred waiting cost is. We assume the system incurs a daily waiting cost of W_i for each patient of class i waiting for service. These waiting cost weights are distinct for each class, thereby allowing us to capture the different levels of urgency among them. Specifically, $W_i > W_j$ indicates that class i patients have a higher level of urgency compared to class j patients. The vector representing the cost of waiting is denoted as W . It is composed of elements W_i , where i ranges from 1 to I , inclusive. Specifically, for each $i \in [1, I]$, W_i specifies the waiting cost associated with the element. The following equation represents the i -th class waiting cost on day t incurred by the system:

$$W_i \sum_{n=0}^{N-1} n u_t^{i,n}. \quad (3.1)$$

The cumulative waiting cost for class i on day t can be derived from (3.1). This is calculated by multiplying n , the total days of waiting, with the number of patients required to wait for n days until receiving service.

To calculate the total waiting cost over the time horizon, we need to sum this up for different patient classes and across different days. Thus, the total waiting cost can be

expressed as follows:

$$\sum_{t=0}^{T-1} \sum_{i=1}^I W_i \sum_{n=0}^{N-1} n u_t^{i,n}. \quad (3.2)$$

This waiting cost definition and its linear growth with the number of days waiting for the services is aligned with the work of [10].

Rejection costs are incurred when the service provider declines patients rather than scheduling them. The rejection cost vector, denoted as ρ , where each element ρ_i corresponds to the weight associated with class i . We assume that the rejection cost follows a quadratic growth pattern with the daily number of rejected patients of a specific class. Each day, the total number of rejected patients of class i can be found by subtracting its daily demand and the total number of scheduled patients of class i on the same day within the booking window. Thus, the total rejection cost accumulated over the time horizon is given by:

$$\sum_{t=0}^{T-1} \sum_{i=1}^I \rho_i \left(D_{i,t} - \sum_{n=0}^{N-1} u_t^{i,n} \right)^2. \quad (3.3)$$

For the sake of simplifying future calculations, we define a $I \times I$ matrix, P , called the rejection cost matrix, where The only non-zero values are found on its diagonal, $P_{i,i} = \rho_i$.

Idle time cost penalizes inefficient scheduling, which can manifest in two ways. Firstly, overscheduling which captures scheduling more patients than the maximum daily capacity can accommodate. Secondly, the days when resources remain underutilized, creating idle periods. We define S , as the idle time cost coefficient. The cost associated with idle time on any given day exhibits a quadratic growth with the difference between the hospital's capacity and the total time already scheduled for that day. More specifically, the total idle cost has the following form:

$$\sum_{t=0}^{T-1} S \left(C_{\max} - \sum_{j=j_{\min}}^t \sum_{i=1}^I M_i u_{i,t-j}^j \right)^2, \quad (3.4)$$

where,

$$j_{\min} = \max(1, t - N + 1). \quad (3.5)$$

We next derive the total cost of the system as the sum of the total rejection, waiting, and total idle time costs. Thus, it can be stated as follows:

$$\begin{aligned} \mathcal{J}(x_{0:(T-1)}, D_{0:(T-1)}, u_{0:(T-1)}) &= \sum_{t=0}^{T-1} \sum_{i=1}^I \rho_i (D_{i,t} - \sum_{n=0}^{N-1} u_t^{i,n})^2 \\ &\quad + \sum_{t=0}^{T-1} \sum_{i=1}^I W_i \sum_{n=0}^{N-1} n u_t^{i,n} \\ &\quad + \sum_{t=0}^{T-1} S \left(C_{\max} - \sum_{j=j_{\min}}^t \sum_{i=1}^I M_i u_j^{i,t-j} \right)^2. \end{aligned} \quad (3.6)$$

Then, the instantaneous cost for each day can be defined as:

$$g(x_t, D_t, u_t) = \sum_{i=1}^I \rho_i (D_{i,t} - \sum_{n=0}^{N-1} u_t^{i,n})^2 + \sum_{i=1}^I W_i \sum_{n=0}^{N-1} n u_t^{i,n} + S (C_{\max} - \sum_{j=j_{\min}}^t \sum_{i=1}^I M_i u_j^{i,t-j})^2. \quad (3.7)$$

Thus, the total cost of the system can be written as,

$$\mathcal{J}(x_{0:(T-1)}, D_{0:(T-1)}, u_{0:(T-1)}) = \sum_{t=0}^{T-1} g(x_t, D_t, u_t). \quad (3.8)$$

3.3 System Dynamics

In this section, we discuss the dynamics of the system. Let x_t represent the state of the system on day t . It is necessary to consider the effect of the action taken by the expert on day t , as it alters the scheduled capacities of days $t, t+1, \dots, t+N-1$. The state of the system on any given day is a function of the previous day's state and the actions taken on that day:

$$x_{t+1} = Ax_t + Bu_t, \quad \forall t \in [0, T-1], \quad (3.9)$$

where A is an $N \times N$ upper shift matrix and B is

$$B \triangleq A(\mathbb{I}_N \otimes M^\top). \quad (3.10)$$

The multiplication of B with u_t illustrates the effect of the action at time t (u_t) on the state at time $t + 1$ (x_{t+1}). In particular, x_{t+1} includes the previously allocated time for the days between $t + 1$ and $t + N - 1$.

Given the dynamics of the system in (3.9) and the cost function the expert is trying to minimize defined in (3.8), the optimization problem faced by the expert is as follows:

$$\begin{aligned} \mathcal{J}(x_0, D_{0:(T-1)}) &= \min_{u_0, u_1, \dots, u_T} \sum_{t=0}^{T-1} g(x_t, D_t, u_t), \\ \text{subject to } x_{t+1} &= Ax_t + Bu_t, \quad \forall t \in [0, T-1]. \end{aligned} \quad (3.11)$$

When the decision-maker is scheduling patients, the state x_t and demand D_t for future dates are unknown. Therefore, every day, the decision-maker is confronted with a cost-to-go function defined as $\bar{J}$. The objective is not just to minimize the instantaneous cost, $g(x_t, D_t, u_t)$, but also to consider the expected cost of the future dates. The cost-to-go function can be written as follows:

$$\begin{aligned} \bar{\mathcal{J}}(x_t, D_t) &= \min_{u_t \geq 0} g(x_t, D_t, u_t) + \mathbb{E} [\bar{\mathcal{J}}(x_{t+1}, D_t) | x_t], \\ \text{s.t. } x_{t+1} &= Ax_t + Bu_t, \quad \forall t \in [0, T-1]. \end{aligned} \quad (3.12)$$

It's worth noting that the main focus of this study is not on how decision-makers solve the optimization problem demonstrated in (3.12), as this aspect has been explored in various advance scheduling studies, e.g., [4], [25], [20], and [24]. Rather, the focus is, given the historical scheduling decisions made, to characterize the criteria considered to make the decisions.

Ensuring that $u_t \geq 0$ poses a significant challenge and makes solving the problem very difficult. Thus, for inverse advance scheduling, we consider the unconstrained problem to estimate the forward problem and refer to it as the unconstrained scheduling problem or modified problem. Consequently, in the modified problem, the cost-to-go function becomes:

$$\begin{aligned} \bar{\mathcal{J}}(x_t, D_t) = \min_{u_t} & \quad g(x_t, D_t, u_t) + \mathbb{E} [\bar{\mathcal{J}}(Ax_t + Bu_t, D_t)|x_t], \\ \text{s.t.} \quad & x_{t+1} = Ax_t + Bu_t, \quad \forall t \in [0, T-1]. \end{aligned} \quad (3.13)$$

On some days, a negative number of patients may be scheduled for a specific day in (3.13). The solution to this issue is to redefine the cost function and add a new term that prevents and penalizes such scenarios. This term is called the smoothness coefficient term and is denoted by δ . Then, the cost-to-go function becomes:

$$\begin{aligned} \bar{\mathcal{J}}(x_t, D_t) = \min_{u_t} & \quad g(x_t, D_t, u_t) + \mathbb{E} [\bar{\mathcal{J}}(Ax_t + Bu_t, D_t)|x_t] + \delta u_t^\top u_t, \\ \text{s.t.} \quad & x_{t+1} = Ax_t + Bu_t, \quad \forall t \in [0, T-1]. \end{aligned} \quad (3.14)$$

The term $\delta u_t^\top u_t$ penalizes the objective when some decision variables get negative values, encouraging smoother actions during the booking window. If the value of δ is set too high, it forces actions to be nearly equal throughout the booking window, ensuring non-negativity.

The cost function previously defined in (3.14) represents the forward problem. As a next step, we will discuss the inverse problem. We assume that an expert has already solved the cost-to-go function (3.14) and generated a sequence of actions, $u_{0:(T-1)}$, based on daily demand, $D_{0:(T-1)}$, and the state of that day, $x_{0:(T-1)}$. Our objective in this study is to estimate the cost parameters they considered for generating this data. In mathematical

terms, the inverse problem is defined as follows:

$$\min_{S, \rho, W, \hat{u}_{0:(T-1)}, \hat{x}_{0:(T-1)}} 0 \quad (3.15a)$$

s.t.

$$\arg \min_{\bar{u}_t} \left(g(\hat{x}_t, D_t, \bar{u}_t) + \mathbb{E} [\bar{\mathcal{J}}(A\hat{x}_t + B\bar{u}_t, D_t) | \hat{x}_t] + \delta \bar{u}_t^\top \bar{u}_t \right) = u_t, \quad \forall t \in [0, T-1], \quad (3.15b)$$

$$\hat{x}_{t+1} = A\hat{x}_t + B\hat{u}_t, \quad \forall t \in [0, T-1], \quad (3.15c)$$

$$\hat{x}_t = x_t, \quad \forall t \in [0, T-1], \quad (3.15d)$$

$$\hat{u}_t = u_t, \quad \forall t \in [0, T-1]. \quad (3.15e)$$

The above optimization problem, (3.15), describes the inverse problem. We assume that there is an expert who generated the set of states, $x_{0:(T-1)}$ and actions, $u_{0:(T-1)}$. In the absence of noise, the objective value may take some value such as 0 but the important part is that our equations must hold and we state them as the constraints of the problem.

The first set of constraints in (3.15b) indicates that the expert selects optimal actions that minimize the cost-to-go function. The second set ensures that the system dynamics are maintained, consistent with the dynamics outlined in (3.3). The third and fourth sets of constraints ensure that the states and actions derived from the inverse problem precisely match the observed states and actions. In essence, this implies that there is no noise in our observations of the system states or expert actions.

3.4 Advance Scheduling with Noise in Observations

Note that the expert's actions might be suboptimal, causing actions to deviate from those obtained based on solving an optimization problem using the cost parameters. In this case,

the optimality constraint (3.15b) is violated. To cover such deviations, we include a noise term, ϵ_i , in the optimal feedback:

$$u_i = Vx_i + W_D D_i + W_E + \epsilon_i, \quad \forall i \in [0, T-1]. \quad (3.16)$$

Uncertainty in the system dynamics then results in violation of constraints (3.15c). Considering uncertainty in the system dynamics, we get,

$$x_{t+1} = Ax_t + Bu_t + \zeta_t, \quad \forall t \in [0, T-1], \quad (3.17)$$

where ζ_t accounts for the deviation from the ideal state transition, due to missing data or other inconsistencies in the scheduling process.

Sometimes the expert makes optimal decisions but our observation of the system's dynamics is imprecise, e.g., certain patients' service times deviate from the estimated service times. This leads to inconsistencies in our observations of states, as described in constraint (3.15d), and actions, as described in constraint (3.15e). To account for this, we introduce a noise term ζ_t into the system state. The observed or estimated state can then be expressed as:

$$\bar{x}_t = x_t + \hat{\zeta}_t. \quad (3.18)$$

The same logic is applied to the actions u_t . In an ideal scenario, these actions would be optimal. However, due to the presence of noise, the observed actions may deviate from the theoretically optimal or expected ones. Thus, we introduce a noise term $\bar{\zeta}_t$ to express the observed or estimated action as follows:

$$\bar{u}_t = u_t + \bar{\zeta}_t. \quad (3.19)$$

Chapter 4

Riccati Approach

The Riccati approach is beneficial in the analysis of systems with quadratic cost functions in terms of state and action, especially when the system dynamics are linear. A notable benefit of Riccati equations is to find a linear function where the action is linearly dependent on the state. Riccati's approach helps us in two ways. Initially, it helps us estimate the parameters of the cost function in (3.15e) (the inverse problem). In addition, it solves the unconstrained problem outlined in (3.14) (the forward problem).

Our model defined in (4.2) is different from the standard Riccati equation in the literature (see e.g. Chapter 4 of [7]). First, the cost function in (4.1) includes cross-term multipliers between state and action denoted by matrix G , resulting from the idle time cost in (3.4), which are not present in the cost function defined in Chapter 4 of [7]. Second, our model includes stochastic demand term denoted by D_t , which is absent in the standard Riccati cost function. In Lemma 1 we show that the optimal action, u_t , is a linear function of the state, x_t , and the daily demand, D_t in 1. For detailed technical information on the previously established aspects of the Riccati approach in the literature and our application

of it, please refer to the appendix [D](#).

In [\(4.1\)](#), we rewrite the instantaneous cost function as a quadratic function of state, action, and demand. For detailed technical proof of this reformulation, refer to the appendix in [A.9](#).

$$\begin{aligned} g(x_t, D_t, u_t) &= L^\top (\mathbb{I}_N \otimes W^\top) u_t + \left(D_t - (\mathbb{1}_N^\top \otimes \mathbb{I}_I) u_t \right)^\top P (D_t - (\mathbb{1}_N^\top \otimes \mathbb{I}_I) u_t) \\ &+ (C_{\max} \mathbb{1}_N - x_t - (\mathbb{I}_N \otimes M^\top) u_t)^\top S e_1 e_1^\top (C_{\max} \mathbb{1}_N - x_t - (\mathbb{I}_N \otimes M^\top) u_t). \end{aligned} \quad (4.1)$$

The instantaneous cost function can be rewritten in the matrix form, introducing new variables to aid in calculations. We define H_t as the linear cost action vector. Matrix F is the linear cost state vector. Matrix G serves as the cost action-state matrix. The quadratic cost state matrix is denoted by Q , while R is the quadratic cost action matrix. The instantaneous cost is represented as a scalar. The matrix form of the instantaneous cost is given by:

$$g(x_t, D_t, u_t) = C_t + 2H_t u_t + 2F x_t + 2x_t^\top G u_t + x_t^\top Q x_t + u_t^\top R u_t, \quad (4.2)$$

where,

$$C_t = D_t^\top P D_t + S C_{\max}^2, \quad (4.3a)$$

$$H_t = \frac{1}{2} L^\top (\mathbb{I}_N \otimes W^\top) - D_t^\top P (\mathbb{1}_N^\top \otimes \mathbb{I}_I) - C_{\max} S e_1 e_1^\top (\mathbb{I}_N \otimes M^\top), \quad (4.3b)$$

$$F = -C_{\max} S e_1 e_1^\top, \quad (4.3c)$$

$$G = (S e_1 e_1^\top) (\mathbb{I}_N \otimes M^\top), \quad (4.3d)$$

$$Q = S e_1 e_1^\top, \quad (4.3e)$$

$$R = (\mathbb{1}_N \otimes \mathbb{I}_I) P (\mathbb{1}_N^\top \otimes \mathbb{I}_I) + (\mathbb{I}_N \otimes M^\top)^\top S e_1 e_1^\top (\mathbb{I}_N \otimes M^\top) + \delta \mathbb{I}_{NI}. \quad (4.3f)$$

While both the matrix form and the normal form are equivalent, the matrix form tends to be more suitable for mathematical calculations. Note that the only time-varying cost matrix is the linear cost action matrix, H_t , and within the definition of this matrix stated in (4.3b), the only time-varying component is the demand. Based on the matrix presentation in (4.2), the following lemma is derived.

Lemma 1 *If the system incurs the daily cost (instantaneous cost) described in (4.2), the optimal action of the expert on the day t , u_t , is a linear function of the state, x_t , and demand, D_t . Specifically,*

$$u_t = Vx_t + W_D D_t + W_E, \quad \forall t \in [0, T-1], \quad (4.4)$$

where,

$$V = -(B^\top KB + \mathcal{R})^{-1}(A^\top KB + G)^\top, \quad (4.5a)$$

$$W_E = \alpha \left(B^\top E^\top + \left(\frac{1}{2} L^\top (\mathbb{I}_N \otimes W^\top) - C_{\max} S e_1^\top (\mathbb{I}_N \otimes M^\top) \right)^\top \right), \quad (4.5b)$$

$$W_D = -\alpha (\mathbb{1}_N^\top \otimes \mathbb{I}_I)^\top P^\top, \quad (4.5c)$$

$$K = Q + A^\top KA - (A^\top KB + G)(B^\top KB + \mathcal{R})^{-1}(A^\top KB + G)^\top, \quad (4.5d)$$

$$E = (F + \mathbb{E}[H_t]V)(\mathbb{I}_N - A - BV)^{-1}, \quad (4.5e)$$

$$\alpha = -(B^\top KB + \mathcal{R})^{-1}. \quad (4.5f)$$

Proof. A.1.

The significance of Lemma 1 lies in the ability to express the expert's optimal actions as a linear function of the state, with the multiplier denoted as V , and the demand, with the

multiplier, denoted as W_D , along with a constant factor referred to as W_E . The matrices V , W_D , and W_E are referred to as the action matrices. Starting from this point onward, we shall refer to V as the *state-action weight matrix*, W_D as the *demand-action weight matrix*, and W_E as the *time-invariant action component*.

By estimating the values of V , W_D , and W_E , we can learn the expert's policy. As a result, given any state and demand, we can derive the optimal action of the expert. Note that matrices V and W_D , as well as the vector W_E , are functions that depend on the cost parameters and the service times.

4.1 Estimating Action Matrices

This section provides an estimation of the action matrices as the first step in estimating the cost parameters. Using the provided data, we find the expert's actions, states, and demands throughout the time horizon, which are represented by the vectors $u_{0:(T-1)}$, $x_{0:(T-1)}$, and $D_{0:(T-1)}$. Note that action matrices are estimated using Lemma 1. Then, the closed-loop equation given in (1) can be restated as follows:

$$u_t = Vx_t + W_D D_t + W_E, \quad \forall t \in [0, T-1]. \quad (4.6)$$

Although we know all the states, $x_{0:(T-1)}$, actions, $u_{0:(T-1)}$, and demands, $D_{0:(T-1)}$, the action matrices that the expert used are unknown. However, the matrices V and W_D , as well as the vector W_E , can be estimated using Equations (4.6). We estimate the action matrices by combining all equations in (4.6) and using the following theorem. Theorem 4.1.1 focuses on estimating the action matrices V , W_D , and W_E . In order to accomplish

this, it is essential to introduce some new variables as detailed below:

$$\begin{aligned}
W &= \begin{bmatrix} V & W_D & W_E \end{bmatrix}, \\
U &= \begin{bmatrix} u_1 & u_2 & \cdots & u_T \end{bmatrix}, \\
Y &= \begin{bmatrix} X \\ D \\ \mathbf{1}_T \end{bmatrix}, \\
X &= \begin{bmatrix} x_1 & x_2 & \cdots & x_T \end{bmatrix}, \\
D &= \begin{bmatrix} D_1 & D_2 & \cdots & D_T \end{bmatrix}.
\end{aligned} \tag{4.7}$$

In (4.7), the matrices U , X , and D represent the concatenated sequences of expert actions $u_{0:(T-1)}$, system states $x_{0:(T-1)}$, and daily demands $D_{0:(T-1)}$.

The matrix Y is constructed by stacking X , D , and a row of ones ($\mathbf{1}_T$), facilitating the representation of the system inputs in a compact form. The matrix W , is composed of V , W_D , and W_E , the action matrices.

Proposition 4.1.1 *Let $\{u_{0:(T-1)}\}$, $\{x_{0:(T-1)}\}$, $\{D_{0:(T-1)}\}$ represent the sequences of expert actions, states, and demands respectively, observed over a discrete time horizon T . If the matrix Y has full column rank, the action matrices V , W_D , and W_E can be uniquely determined by solving the subsequent optimization problem:*

$$\underset{W}{\text{minimize}} \quad \|U - WY\|_F.$$

Proof. [A.2](#).

In Theorem [4.1.1](#), the estimation of action matrices is enhanced by the length of path trajectory. In other words, a longer path trajectory reduces the estimation error in W .

The reason is that more equations lead to larger sizes for both U and Y , enhancing the ability of least squares optimization to find more accurate values for the action matrices. A reduction in the estimation error in W has a cascading effect on the estimation errors in V , W_D , and W_E , thanks to their interdependencies.

Once the action matrices are estimated, it is necessary to determine whether certain cost parameters contributed to these matrices. In the following sections, using the recovered action matrices, we estimate a unique set of cost parameters up to a scalar ambiguity under some conditions.

4.2 Scalar Ambiguity in Cost Function

In Theorem 4.1.1, the action matrices are estimated. However, the cost parameters remain unknown. These parameters include the rejection cost parameters (ρ), idle time cost (S), and waiting cost parameters (W), all of which should be estimated. In matrix form stated in (4.2), note that each matrix is unknown at the beginning. However, these matrices can be found step by step and subsequently, the cost parameters can be estimated. To estimate the cost parameters, the following lemma is crucial.

Lemma 2 *The action matrices, V, W_D , and W_E , defined in Lemma 1 remain unchanged when all cost parameters are multiplied by a specific non-zero number.*

Proof. A.4.

Based on Lemma 2, we conclude that the multiplication of all cost parameters by $\frac{1}{S}$, does not alter the action matrices. Therefore, without loss of generality, we assume that $S = 1$. Furthermore, this ensures the uniqueness of the remaining cost parameters by resolving

the scalar ambiguity. Note that, Matrices Q , representing the state quadratic multiplier, G , the state-action cross multiplier, and F , the state linear multiplier, are independent of both the rejection cost vector ρ and the waiting cost vector W . Consequently, assuming $S = 1$, they can be considered as known matrices.

4.3 Riccati Gain Matrix

Given the known matrices Q , F , and G found in Section 4.2, along with the action matrices V , W_D , and W_E estimated in Theorem 4.1.1, we aim to find matrix K . By applying Riccati equations as detailed in Lemma 1, the matrix K can be estimated. As shown in (4.5d), matrix K is uniquely identifiable. Notably, this solution lies within the class of positive semi-definite symmetric matrices. We first revisit Lemma 1 that defines the matrix K , as follows:

$$K = Q + A^\top K A - (A^\top K B + G)(B^\top K B + \mathcal{R})^{-1}(A^\top K B + G)^\top. \quad (4.8)$$

Lemma 3 *Consider the Riccati Gain Matrix K , which is defined in (4.8). Then, K can be determined through the following vectorized formulation:*

$$\text{vec}(K) = (\mathbb{I}_{N \times N} - (A^\top \otimes A^\top) - (\theta^\top \otimes A^\top))^{-1}(\text{vec}(Q) + \text{vec}(GV)), \quad (4.9)$$

where, θ is defined as:

$$\theta \triangleq BV, \quad (4.10)$$

Proof. A.5.

Now, we can estimate K simply by converting $\text{vec}(K)$ back to the matrix form K .

4.4 Estimation of Rejection Cost Parameters

Rejection cost parameters can be estimated through action matrices using two distinct ways. The first approach involves using the state-action weight matrix, denoted as V , which is further discussed in Section 4.4.1. In the second approach, the demand-action weight matrix, denoted as W_D , is leveraged, as discussed in the subsequent Section 4.4.2.

4.4.1 State-Action Weight Matrix

In this section, rejection cost parameters are estimated using the equation that defines the state-action weight matrix, V , as stated in Lemma 1:

$$V = -(B^\top KB + \mathcal{R})^{-1}(A^\top KB + G)^\top, \quad (4.11a)$$

$$(B^\top KB + \mathcal{R})V = -(A^\top KB + G)^\top. \quad (4.11b)$$

The two relations provided in (4.11a) and (4.11b) are equivalent. This equivalence is achieved by multiplying both sides of (4.11a) by $(B^\top KB + \mathcal{R})$.

Utilizing (4.11b), our objective is to estimate the rejection cost parameters and the smoothness coefficient term. In order to achieve this, we define variables $\phi_{ij}, c_{ij}, C_1, C_2$, and C_3 as detailed below:

$$\begin{aligned} \phi_{ij} &\triangleq ((PC_1^\top)V)_{ij} = \rho_i \sum_{k=0}^N V_{KI+i,j}, \quad \forall i \in [1, I], \forall j \in [1, N], \\ c_{ij} &\triangleq ((C_1^\top C_1)^{-1}C_1^\top V)_{ij}, \quad \forall i \in [1, I], \forall j \in [1, N], \\ C_1 &\triangleq (\mathbb{1}_N \otimes \mathbb{I}_I), \\ C_3 &\triangleq (C_1^\top C_1)^{-1}C_1^\top C_2, \\ C_2 &\triangleq -((\mathbb{I}_N \otimes M^\top)^\top e_1 e_1^\top (\mathbb{I}_N \otimes M^\top))V - B^\top K(\theta + A) - G^\top. \end{aligned} \quad (4.12)$$

Note that P denotes the rejection cost matrix, M represents the service time vector, K is the Riccati Gain Matrix as identified in Lemma 3, and G is the cost action-state matrix defined in Lemma 1.

The variables defined in (4.12) are then employed in Theorem 4.4.1 to construct an optimization problem for determining the cost parameters.

Proposition 4.4.1 *By applying the Riccati approach and state-action weight matrix equations, the optimization problem defined below aims to minimize the error between the observed and estimated values of the Riccati equations, leading to the estimation of the rejection cost parameters $\rho_{1:I}$ for each class and the smoothness coefficient term δ .*

$$\begin{aligned} \min_{\rho_1, \rho_2, \rho_3, \dots, \rho_I, \delta} & \sum_{i=1}^I \sum_{j=1}^N (\phi_{ij} + \delta c_{ij} - C_{3i,j})^2 \\ \text{s.t.} \quad & \phi_{ij} = \rho_i \sum_{k=0}^N V_{KI+i,j}, \quad \forall i \in [1, I], \quad \forall j \in [1, N]. \end{aligned} \tag{4.13}$$

Proof. A.3.

Note that (4.13) can be used in situations where there is noise in the equations. However, in the absence of noise, (4.13) transforms into the following equation, as detailed in Appendix 4.4.1:

$$\phi_{ij} + \delta c_{ij} = C_{3i,j}, \quad \forall i \in [1, I], \quad \forall j \in [1, N]. \tag{4.14}$$

Equation (4.14) presents a set of $N \times I$ equations with $I + 1$ unknowns, including the rejection cost parameters for each class and the smoothness coefficient term. In noise-free scenarios, solving these equations enables the determination of $\rho_1, \rho_2, \rho_3, \dots, \rho_I$, and δ . Notably, (4.14) contains redundant equations, so in the presence of noisy data, Theorem 4.4.1

recommends an optimization-based approach to estimate the rejection cost parameters. Despite potential deviations in some equations due to noise, using all available equations improves the estimation of these parameters.

4.4.2 Demand-Action Weight Matrix

This section explores an alternative method to estimate the rejection cost parameters and the smoothness coefficient. Instead of employing the state-action weight matrix discussed in 4.4.1, we utilize the definition of the demand-action weight matrix, W_D stated in (4.5c). To assess consistency and accuracy, we can compare the results of these two approaches when dealing with noisy data. We restate the definition of W_D as follows:

$$W_D = -\alpha(\mathbb{1}_N^\top \otimes \mathbb{I}_I)^\top P^\top. \quad (4.15)$$

By employing the definition of α stated in (4.5f), we obtain the following equation:

$$(B^\top KB + \mathcal{R})W_D = (\mathbb{1}_N^\top \otimes \mathbb{I}_I)^\top P^\top. \quad (4.16)$$

Utilizing (4.16), our objective is to estimate the rejection cost parameters and the smoothness coefficient term. In order to achieve this, we define variables Θ , $\bar{\Theta}$, C_4 , and γ as detailed below:

$$\begin{aligned} \Theta &\triangleq ((C_1^\top C_1)^{-1} C_1^\top W_D), \\ \bar{\Theta} &\triangleq \mathbb{I}_I - C_1^\top W_D, \\ C_4 &\triangleq (C_1^\top C_1)^{-1} C_1^\top \gamma, \\ \gamma &\triangleq (B^\top KB)W_D - ((\mathbb{I}_N \otimes M^\top)^\top e_1 e_1^\top (\mathbb{I}_N \otimes M^\top))W_D. \end{aligned} \quad (4.17)$$

Proposition 4.4.2 *By applying the Riccati approach and demand-action weight matrix equations, the optimization problem defined below aims to minimize the error between the observed and estimated values of the Riccati equations, leading to the estimation of the rejection cost parameters $\rho_{1:I}$ for each class and the smoothness coefficient term δ .*

$$\min_{\rho_1, \rho_2, \rho_3, \dots, \rho_I, \delta} \sum_{i=1}^I \sum_{j=1}^I (\rho_i \bar{\Theta}_{ij} + \delta \Theta_{ij} - C_{4i,j})^2, \quad \forall i \in [1, I], \quad \forall j \in [1, I]. \quad (4.18)$$

Proof. [A.6](#).

Similar to the previous section [4.4.1](#), we can argue that [\(4.18\)](#) can be used in situations where there is noise in the equations. However, in the absence of noise, [\(4.18\)](#) transforms into the following equation:

$$\rho_i \bar{\Theta}_{ij} + \delta \Theta_{ij} = C_{4i,j}, \quad \forall i \in [1, I], \quad \forall j \in [1, I]. \quad (4.19)$$

As detailed in Appendix [4.4.2](#), [\(4.19\)](#) is proven to hold in scenarios free of noise. Equation [\(4.19\)](#) presents a set of $I \times I$ equations with $I + 1$ unknowns, including the rejection cost parameters for each class and the smoothness coefficient term. In noise-free scenarios, solving these equations enables the determination of $\rho_1, \rho_2, \rho_3, \dots, \rho_I$, and δ . By utilizing the minimization problem in [\(4.19\)](#), which aims to satisfy the demand-action weight matrix equations, we can estimate the rejection cost parameters, ρ_i , as well as the smoothness coefficient, δ .

4.5 Estimation of Waiting Cost Parameters

In Sections [4.4.1](#) and [4.4.2](#), we estimated the rejection cost parameters ρ and δ , allowing us to determine R , which depends on ρ , δ , and service time vector, M , as defined in [\(4.3f\)](#).

Below, we restate the equation defining R :

$$R = (\mathbb{I}_N \otimes \mathbb{I}_I)P(\mathbb{1}_N^\top \otimes \mathbb{I}_I) + (\mathbb{I}_N \otimes M^\top)^\top e_1 e_1^\top (\mathbb{I}_N \otimes M^\top) + \delta \mathbb{I}_{NI}. \quad (4.20)$$

As a result of (4.20), R is determined and used to identify waiting cost parameters in Section 4.5.1.

4.5.1 Time-Invariant Action Component

The time-invariant action component, W_E , defined in (4.5b), can be written as follows:

$$W_E = \alpha(B^\top E^\top + (\frac{1}{2}L^\top(\mathbb{I}_N \otimes W^\top) - C_{\max}e_1^\top(\mathbb{I}_N \otimes M^\top))^\top). \quad (4.21)$$

Multiplying both sides of (4.21) by α^{-1} , given in (4.5f), we get:

$$-(B^\top KB + R)W_E = B^\top E^\top + (\frac{1}{2}L^\top(\mathbb{I}_N \otimes W^\top) - C_{\max}e_1^\top(\mathbb{I}_N \otimes M^\top))^\top. \quad (4.22)$$

Utilizing (4.22), our objective is to estimate the waiting cost parameters. In order to achieve this, we define variables C_5, C_6, C_7 , and C_8 as detailed below:

$$C_7 \triangleq C_5 - \left(C_6 F^\top - C_6 V^\top (\mathbb{E}[D_t^\top])^\top P(\mathbb{1}_N^\top \otimes \mathbb{I}_I) + C_{\max}e_1^\top(\mathbb{I}_N \otimes M^\top)^\top + (C_{\max}e_1^\top(\mathbb{I}_N \otimes M^\top))^\top \right), \quad (4.23a)$$

$$C_8 \triangleq (C_6 V^\top + \mathbb{I}_{NI}), \quad (4.23b)$$

$$C_6 \triangleq B^\top (\mathbb{I}_N - A - BV)^{-T}, \quad (4.23c)$$

$$C_5 \triangleq B^\top E^\top + \left(\frac{1}{2}L^\top(\mathbb{I}_N \otimes W^\top) - C_{\max}e_1^\top(\mathbb{I}_N \otimes M^\top) \right)^\top. \quad (4.23d)$$

Proposition 4.5.1 *For each class i , let W_i be the waiting cost parameter. By applying the Riccati approach and its equations, the optimization problem defined below aims to*

minimize the error between the observed and estimated values of the Riccati equations, leading to the estimation of the waiting cost parameters.

$$\min_{W_1, W_2, W_3, \dots, W_I} \sum_{k=1}^{NI} \left((C_7)_k - \sum_{i=1}^{NI} \frac{n}{2} (C_8)_{k,i} W_j \right)^2 . \quad (4.24)$$

where,

$$j = \text{rem}(i, I) + 1, \quad (4.25)$$

$$n = \lfloor \frac{i}{I} \rfloor. \quad (4.26)$$

Proof. [A.7](#).

In Theorem [4.5.1](#), we can see [\(4.24\)](#) can be used in situations where there is noise in the equations. However, in the absence of noise, [\(4.24\)](#) transforms into the following equation:

$$(C_7)_k = \sum_{i=1}^{NI} \frac{n}{2} (C_8)_{k,i} W_j , \quad \forall k \in [1, I \times N], \quad (4.27)$$

where,

$$\begin{aligned} j &= \text{rem}(i, I) + 1, \\ n &= \lfloor \frac{i}{I} \rfloor. \end{aligned} \quad (4.28)$$

In [\(4.27\)](#), we encounter a set of $N \times I$ equations with I unknowns, which include the waiting cost parameters. Nonetheless, through [\(4.24\)](#), we can determine these waiting cost parameters in both noisy and noiseless environments.

We can now evaluate the Riccati approach's efficacy for our problem setting. Time efficiency is one of the major advantages of this method. Solving the forward problem

discussed in (3.14), requires daily calculations of optimal actions. The Riccati approach can find the optimal actions quickly. The reason behind that is the optimal actions are expressed as linear functions of the state and demand, as indicated in Lemma 1.

An additional benefit of the Riccati method is its capability to replicate the policy of experts. This method finds action matrices, which capture the expert’s decision-making process. This is especially useful if our objective is not estimating the cost parameters but replicating the action matrices. For example, if we want to replicate how well-managed hospitals schedule their resources, the Riccati method is effective. It allows us to not just copy successful strategies, but also still gives us the option to estimate the cost parameters.

Note that, this approach has some limitations. While a greater time horizon, T , can reduce the estimation error in V , W_D , and W_E in (4.1.1), this does not guarantee a straightforward, proportional reduction in the estimation error for the rejection costs and the waiting cost parameters. Therefore, the relationship between the number of equations and the accuracy of estimation is not direct.

Further, the Riccati approach requires extensive parameter learning. To put this into perspective, our primary learning targets are the rejection and waiting cost parameters, which include $2 \times I$ parameters. However, the Riccati method requires us to learn $N \times I \times (N + I + 1)$ parameters for the action matrices. Additionally, we must also identify K , with dimensions $N \times N$. This complexity could make alternative methods more attractive when the focus is to obtain cost parameters. In the following section, we will explore another approach that aims to directly estimate the cost parameters, thereby potentially mitigating some of these limitations.

Chapter 5

Hamiltonian Approach

In this section, we aim to directly estimate the cost parameters defined in (3.15) without finding the action matrices defined in Lemma 1. We explore one of the most traditional methods in both optimal control theory and optimization algorithms, known as the Hamiltonian Approach. In contrast to the Riccati approach, which is tailored just for quadratic cost functions, the Hamiltonian Approach is more versatile. Scalability is one of the most notable characteristics of the Hamiltonian Approach. While both the Hamiltonian and the Riccati approaches achieve reduced estimation errors with extended trajectories, the Hamiltonian approach accomplishes this objective more efficiently.

In [27], the Hamiltonian approach is utilized with the cost function being a quadratic function of the states and actions of the problem. A unique set of cost parameters can be identified, subject to scalar ambiguity under certain assumptions. However, their model excludes cross and linear terms as well as stochastic variables like daily demand in our study in the cost function. Additionally, their model does not account for uncertainty in the observations. For detailed technical information on the previously established aspects

of the Hamiltonian approach in the literature and our application of it, please refer to the appendix C.

We assume that the expert is addressing an optimization problem, referred to as the forward problem, as previously discussed in (3.14):

$$\min_{u_0, u_1, \dots, u_T} J(x_0, D_{0:(T-1)}) = \sum_{t=1}^T g(x_t, D_t, u_t), \quad (5.1a)$$

$$\text{s.t. } x_{t+1} = Ax_t + Bu_t, \quad \forall t \in [0 : (T-1)]. \quad (5.1b)$$

Equation (5.1b) defines the dynamic of the system. To estimate the cost parameters using the inverse problem, we first relax the constraints described in (5.1b) using Lagrangian relaxation. As a result, the forward problem becomes:

$$\mathcal{L}(X, \lambda) \triangleq \min_{u_0, u_1, \dots, u_T} \sum_{k=0}^{T-1} g(x_k, u_k) + \sum_{k=0}^{T-1} \lambda_k^\top (Ax_k + Bu_k - x_{k+1}). \quad (5.2a)$$

Here, g represents the per-stage weighting cost (instantaneous cost) in the objective function as defined in (4.2). The Lagrangian function, $\mathcal{L}(X, \lambda)$, is defined to help us solve the inverse problem by analyzing the forward problem. Using Pontryagin's maximum principle, discussed in Section 3.3 in [11], they prove that a necessary condition for the sequences $x_{0:(T-1)}$ and $u_{0:(T-1)}$ to be a minimizer of the cost function is the existence of a sequence of vectors $\lambda_{(-1):(T-2)}$ called co-states. Given the presence of an expert who has minimized the cost function by solving the optimization problem outlined in (3.14), we can conclude that the sequences $x_{0:(T-1)}$ and $u_{0:(T-1)}$ are the optimal solutions.

Utilizing the definition of $g(x_k, D_k, u_k)$ stated in (4.2), the Hamiltonian function is defined as follows:

$$\begin{aligned} \mathbb{H}(x_k, u_k, \lambda_k) &= \mathbb{H}_k \triangleq g(x_k, u_k) + \lambda_k^\top (Ax_k + Bu_k) \\ &= C_k + 2H_k u_k + 2F x_k + 2x_k^\top G u_k + x_k^\top Q x_k + u_k^\top R u_k \\ &\quad + \lambda_k^\top (Ax_k + Bu_k), \quad \forall k \in [0 : (T-1)]. \end{aligned} \quad (5.3)$$

Consequently, we can express the Lagrangian function in terms of the Hamiltonian function and the Lagrangian multipliers, as follows:

$$\begin{aligned}
\mathcal{L}(X, \lambda) &= \sum_{k=0}^{T-1} g(x_k, u_k) + \sum_{k=0}^{T-1} \lambda_k^\top (Ax_k + Bu_k - x_{k+1}) \\
&= \sum_{k=0}^{T-1} (g(x_k, u_k) + \lambda_k^\top (Ax_k + Bu_k)) - \sum_{k=0}^{T-1} \lambda_k^\top x_{k+1} \\
&= \sum_{k=0}^{T-1} \mathbb{H}_k - \sum_{k=0}^{T-1} \lambda_k^\top x_{k+1}.
\end{aligned} \tag{5.4}$$

To address the inverse problem, it is helpful to determine the values of the co-states, which are derived from the Lagrangian function. In Section 3.3 of the study presented in [11], using Pontryagin's maximum principle, it is established that within the Hamiltonian approach, a path trajectory's optimality is assured when four specific conditions are met. For the detailed mathematical formulations, please refer to A.10. Below, we provide a summary of the outcomes associated with each condition:

1. **System Dynamics:** This condition guarantees that the state of the system on any particular day depends on the state of the previous day and the actions taken on that day. Our system meets this condition, as the system dynamics, detailed in (3.9), are likewise a linear function of the previous day's state and the actions executed on that day.
2. **Adjoint Equations:** This represents the relationship between co-states and the Hamiltonian function, expressed as:

$$\lambda_{k-1} = \frac{\partial \mathbb{H}_k}{\partial x_k}, \quad \forall k \in [0, (T-1)]. \tag{5.5}$$

3. **Boundary Condition:** This condition ensures that the length of the path trajectory is finite.

4. Hamiltonian Condition: This condition guarantees that the actions taken by the expert are optimal.

Similar to what was discussed in Section 4.2 and based on Lemma 2, without loss of generality, we assume that the idle time coefficient multiplier, S , is equal to 1. As a result, Q , G , and F can be considered as known matrices, as outlined in Section 2, since they are only dependent on idle time coefficient multiplier.

Co-state variable λ_{k-1} can be found by as discussed below. We can express $\mathbb{H}_k$ as:

$$\begin{aligned}\mathbb{H}_k &= g(x_k, D_k, u_k) + \lambda_k^\top (Ax_k + Bu_k) \\ &= C_k + 2H_k u_k + 2Fx_k + 2x_k^\top Gu_k + x_k^\top Qx_k + u_k^\top Ru_k + \lambda_k^\top (Ax_k + Bu_k).\end{aligned}\tag{5.6}$$

Based on the definition of $\mathbb{H}_k$ and the adjoint equations, Condition 2, our objective is to identify the co-state variables, λ . This can be expressed as:

$$\begin{aligned}\lambda_{k-1} &= \frac{\partial \mathbb{H}_k}{\partial x_k} \\ &= \frac{\partial (C_k + 2H_k u_k + 2Fx_k + 2x_k^\top Gu_k + x_k^\top Qx_k + u_k^\top Ru_k + \lambda_k^\top (Ax_k + Bu_k))}{\partial x_k} \\ &= 2F^\top + 2Gu_k + 2Qx_k + A^\top \lambda_k, \quad \forall k \in [0, T-1].\end{aligned}\tag{5.7}$$

Only two variables are unknown in this equation: λ_{k-1} and λ_k . Thus, we can write λ_{k-1} as a function of λ_k . Subsequently, we employ (5.7) and recursively determine the co-state variables, λ s using the co-state terminal, $\lambda_T = 0$, based on Condition 3.

In Proposition ??, we outline an optimization problem designed to estimate all the cost parameters. To facilitate the estimation process using the Proposition 5.0.1, it is helpful to define $\bar{\kappa}_k$, for simplification of calculations, as detailed below:

$$\begin{aligned}\bar{\kappa}_k &\triangleq -2G^\top x_k - B^\top \lambda_k + 2C_{\max} e_1^\top (\mathbb{I}_N \otimes M^\top) \\ &\quad - 2(\mathbb{I}_N \otimes M^\top)^\top e_1 e_1^\top (\mathbb{I}_N \otimes M^\top) u_k, \quad \forall k \in [0, T-1].\end{aligned}\tag{5.8}$$

Proposition 5.0.1 *We propose the optimization problem formulated below that estimates the cost parameters $W_{1:I}, \rho_{1:I}$ and δ :*

$$\begin{aligned} \min_{\rho_{1:I}, W_{1:I}, \delta} & \sum_{k=0}^T \sum_{i=0}^{NI-1} (\bar{\kappa}_k^i - \kappa_k^i)^2 \\ \text{s.t. } & \kappa_k^i = nW_j - 2(D_k^j - u_k^j) + \delta u_k^{j,n}, \quad \forall k \in [0, T-1], \quad \forall i \in [0, (NI-1)]. \end{aligned} \quad (5.9)$$

where,

$$j = \text{rem}(i, I) + 1, \quad (5.10a)$$

$$n = \lfloor \frac{i}{I} \rfloor. \quad (5.10b)$$

Proof. [A.8](#).

By applying least squares, we estimate the best cost parameter that fits to (5.9). Extending the time horizon will introduce additional terms to (5.9). As each term represents an individual equation, these extra terms are important for the estimation of cost parameters, particularly in scenarios with disturbances in certain equations. Specifically, in the absence of noise in the system, (5.9) turns into the following set of equations:

$$\bar{\kappa}_k^i = \kappa_k^i \quad \forall k \in [0 : T-1], \quad \forall i \in [0 : (NI-1)] \quad (5.11)$$

where,

$$\forall i \in [0, (NI-1)], \quad (5.12a)$$

$$j = \text{rem}(i, I) + 1, \quad (5.12b)$$

$$n = \lfloor \frac{i}{I} \rfloor. \quad (5.12c)$$

In (5.11), we encounter a set of $T \times N \times I$ equations and $2I + 1$ unknowns, which include the cost parameters. This system of linear equations can be solved to determine the cost parameters.

Chapter 6

Numerical Study

6.1 Performance of the Proposed Approaches

In this research, Propositions 4.4.1, 4.4.2, and 5.0.1 (Riccati and Hamiltonian approaches) estimate rejection cost parameters and the smoothness coefficient term. Moreover, Propositions 4.5.1 and 5.9 can be used to estimate waiting cost parameters. However, the robustness and effectiveness of our methods under noisy environments have not yet been tested.

We created a simulation model featuring an expert making decisions based on daily demand. The simulator stores information about the expert's actions, system states, and daily demand while information about their cost function is discarded. The simulation model consists of some input parameters, including service times (M), booking horizon (N), maximum service provider's capacity ($C_{\max}$), and time horizon (T), which define the service provider's operating conditions. Simulation inputs also include cost parameters rejection costs (ρ), waiting costs (W), and smoothness coefficient term, δ . Poisson distribution

characterizes the daily demand as the final input.

Upon receiving these inputs, the simulation model generates the expert's optimal actions over the time horizon, utilizing the Riccati approach discussed in Section 4. It also generates the linear cost state vector, F , the linear cost action vector, H_t , cost action-state matrix, G , quadratic cost state matrix, Q , and lastly quadratic cost action matrix, R which are defined in (4.2). Based on these cost matrices, it then generates the action matrices, namely the State-Action Weight Matrix V , the Demand-Action Weight Matrix W_D , and the Action Offset Vector W_E defined in Lemma 1. With this sequence of operations, the forward problem defined in (3.14) is solved, resulting in the generation of the expert's actions. Following the expert's actions, states are generated according to the system dynamics stated in (3.3).

The second part of the implementation covers the inverse problem defined in (3.15), where given the states ($x_{0:(T-1)}$), actions ($u_{0:(T-1)}$), demands ($D_{0:(T-1)}$), we estimate the cost parameters used by the expert. The cost parameters are retrieved using the Riccati approach and Hamiltonian approach discussed in Section 4 and 5. We will demonstrate the robustness of these methods in the presence of observation noise and system dynamics noise discussed in Section 3.4. Additionally, varying conditions such as changes in the time horizon and booking horizon might affect the estimation error in cost parameters. The results will illustrate the correlation between the estimation error and noise, time horizon T , and booking horizon N which are discussed in the following sections.

6.1.1 Sensitivity Analysis on Time Horizon

One of the key strengths of the Riccati and the Hamiltonian approach lies in their scalability and the capacity to minimize estimation errors as the dataset expands. In this context, the

analysis demonstrates that extending the time horizon directly correlates with a reduction in estimation errors. The proposed methods have a significant impact on reliability and robustness, making them ideal for applications that require high accuracy and large data collection.

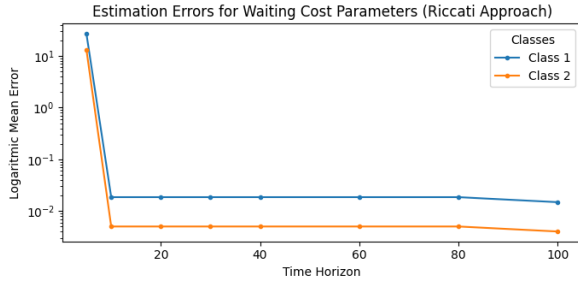
Initially, we focus only on a noise-free environment. Specifically, we assume that there is no noise either in the system dynamics or in the observations, and all executed actions are optimal. We use this as a baseline test to validate the effectiveness of the proposed methods.

In the Riccati approach, the estimation error in the optimization problem, given in Proposition 4.1.1, decreases with the increase in data length. Consequently, we attain a lower estimation error in action matrices- the State-Action Weight Matrix V , the Demand-Action Weight Matrix W_D , and the Action Offset Vector W_E . This leads to a reduced estimation error in the rejection cost parameters found in (4.13), (4.18), and the waiting cost parameters found in (4.24).

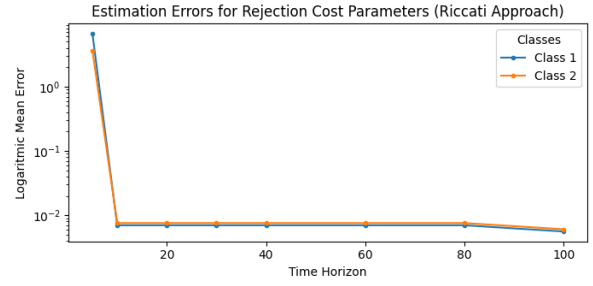
In the Hamiltonian approach, the estimation error in the optimization problem presented in 5.9, diminishes with the extension of time horizon. As a result, the rejection and waiting cost parameters are estimated with greater accuracy, due to the increased number of equations in the optimization problem. In (5.11), the number of equations is equal to $T \times N \times I$, meaning that an expansion of the time horizon directly leads to more equations.

In Figure 6.1, the findings provide estimation errors in the Hamiltonian approach. Figure 6.1 illustrates the estimation errors involved in the Riccati approach as the time horizon is extended.

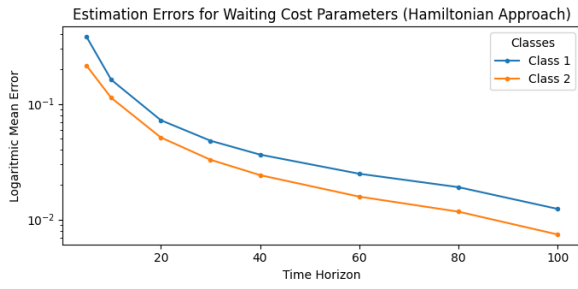
These figures indicate that when the time horizon is small and is around 10 days up to 40 days, the Riccati approach has less estimation error compared to the Hamiltonian



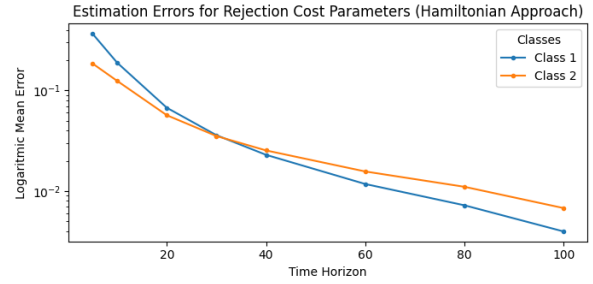
(a) Estimation error for Riccati Waiting Cost



(c) Estimation error for Riccati Rejection Cost



(b) Estimation error for Hamiltonian Waiting Cost



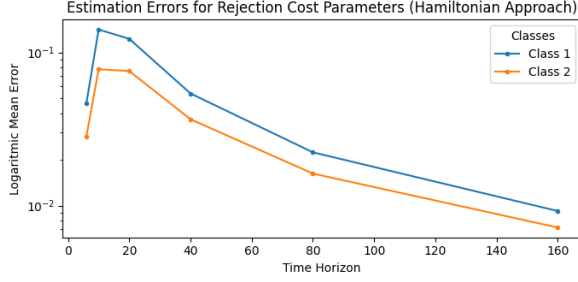
(d) Estimation error for Hamiltonian Rejection Cost

Figure 6.1: Estimation Error and the Time Horizon

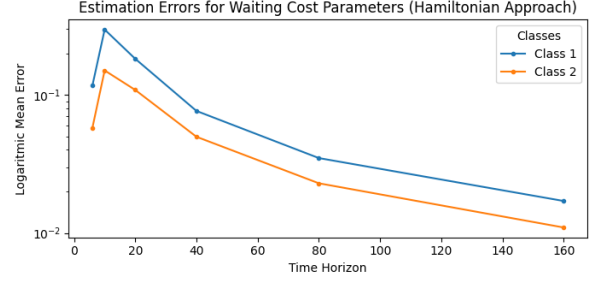
approach. The reason is that the action matrices, estimated in Proposition 4.1.1, are converging faster than the method of estimating cost parameters with the Hamiltonian approach discussed in Proposition 5.0.1. This is one advantage of the Riccati approach compared to the Hamiltonian approach. However, with an increasing time horizon, the Hamiltonian approach demonstrates a greater reduction in estimation error compared to the Riccati approach. Thus, from these two figures, we can infer that the Riccati approach performs better in settings with shorter time horizons, whereas the Hamiltonian approach is more suitable for longer time horizons.

6.1.2 Sensitivity Analysis on Time Horizon with Noisy Dynamics

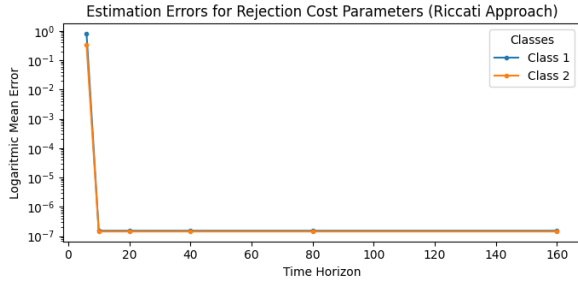
In Figure 6.2, it is demonstrated that our methods are robust against noise in the system dynamics, even when the noise has a standard deviation of 100, as the cost parameters are accurately identified and the estimation error does not change significantly compared to the Figure 6.1. This figure shows the estimation error in determining the rejection and waiting cost parameters with the extension of the time horizon. Similar to Figure 6.1, the estimation error in the Hamiltonian approach and the Riccati approach decreases with the extension of the time horizon.



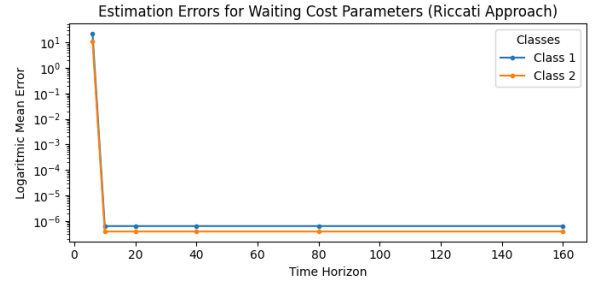
(a) Flowchart for Hamiltonian Rejection



(b) Flowchart for Hamiltonian Waiting



(c) Caption for Third Image



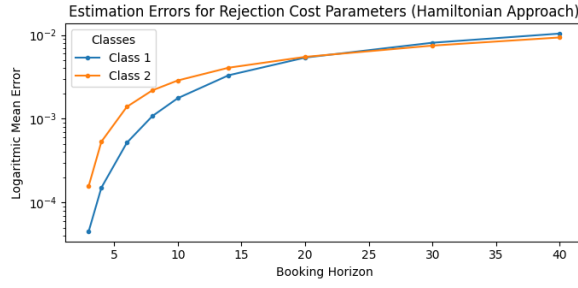
(d) Caption for Fourth Image

Figure 6.2: Estimation Error and Time Horizon with Noisy Dynamics

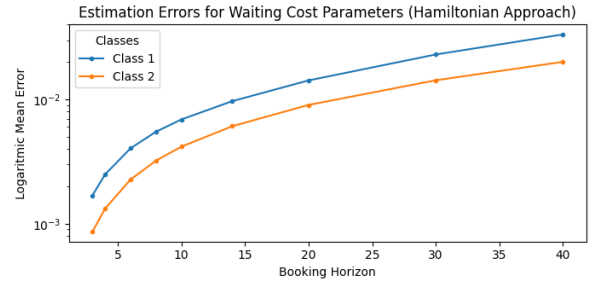
6.1.3 Sensitivity Analysis on Booking Horizon

In this section, we examine the performance of the proposed approaches when there is noise in the actions, states, or dynamics of the system. Figure 6.3 illustrates the relationship between the estimation error and the booking horizon where there is no noise in the equations.

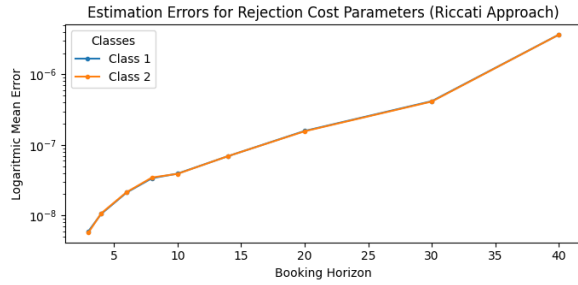
In the Hamiltonian approach, the number of unknowns rises with an increase in the booking horizon, which leads to larger estimation errors. All co-state values must be determined before estimating the cost parameters. Based on the second condition of the optimality of a path trajectory discussed in 2, the size of the co-state vectors equals N .



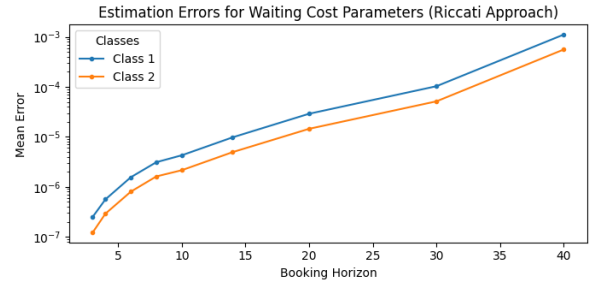
(a) Estimation error for Hamiltonian Rejection Cost



(b) Estimation error for Hamiltonian Waiting Cost



(c) Estimation error for Riccati Rejection Cost



(d) Estimation error for Riccati Waiting Cost

Figure 6.3: Estimation Error Relative to the Booking Horizon

Consequently, as shown in Figure 6.3, the estimation error in the Hamiltonian approach increases with the rise in N . Additionally, in the Riccati approach, the number of unknowns grows as the booking horizon extends, leading to a larger estimation error. The Riccati approach is more suitable for scenarios with large booking horizons than the Hamiltonian approach.

6.1.4 Sensitivity Analysis on Booking Horizon with Noisy Dynamics

In Figure 6.4, it is demonstrated that our methods are robust against noise in the system dynamics, even when the noise has a standard deviation of 100, as the cost parameters are accurately identified. This figure also shows the estimation error in determining the rejection and waiting cost parameters. Similar to Figure 6.3, the estimation error in the Hamiltonian approach increases with the expansion of the booking horizon. As shown in Figure 6.4, although the Riccati approach exhibits a similar effect, it is more effective for longer booking horizons. For creating all the graphs in the sensitivity analysis, we utilized Python along with the NumPy package. To solve the forward problem, we employed the Riccati forward approach as a simulator. For addressing the inverse problem, we continued to use the least squares method from the NumPy package to solve the optimization problems as defined in Propositions 4.4.1, 4.4.2, 4.5.1, and 5.0.1.

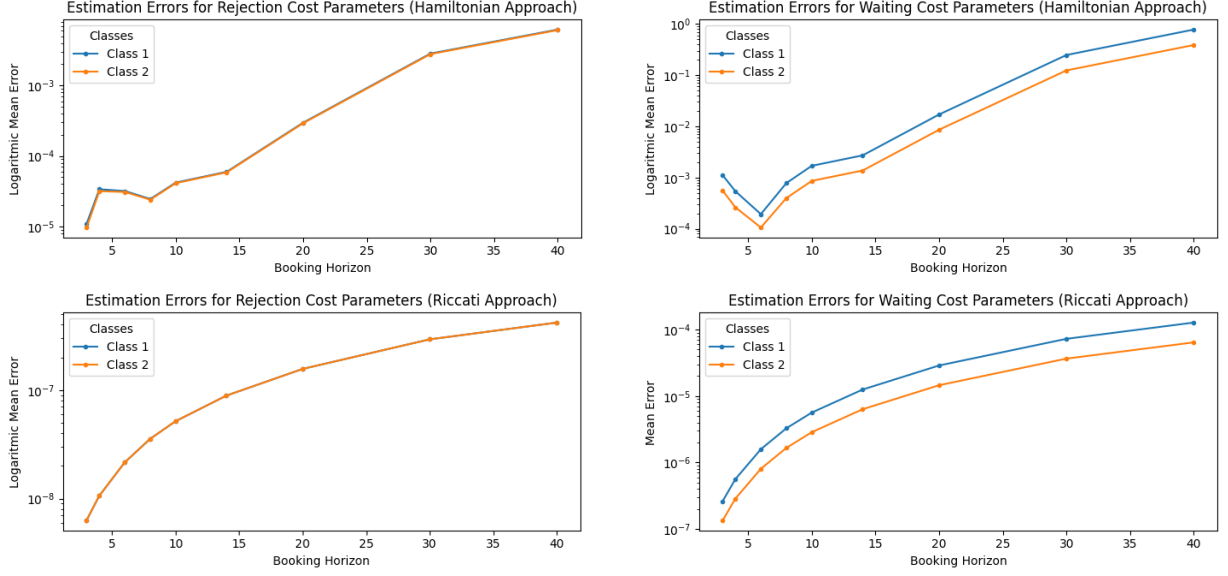


Figure 6.4: Estimation Error and Booking Horizon with Noisy Dynamics

6.2 Case Study

We have access to historical data of two hospitals over two years. For each patient, the data specifies the arrival date, priority level, and service date.

Patients are grouped into 4 different classes based on their priority level. The priority levels are assumed to play a significant role in patient scheduling, with priority 1 incurring the highest cost, followed by priority 2, 3, and 4. Patients' daily arrivals are stochastic. Figure 6.5 indicates the demand percentages across priority levels for the two hospitals.

We can calculate the daily demand based on the patient's arrival date. For each day, the daily demand vector is constructed such that D_t^i represents the number of patients of class i that arrived on day t . In Figure 6.6, the daily distribution of patient arrivals at Hospitals 1 and 2 are illustrated.

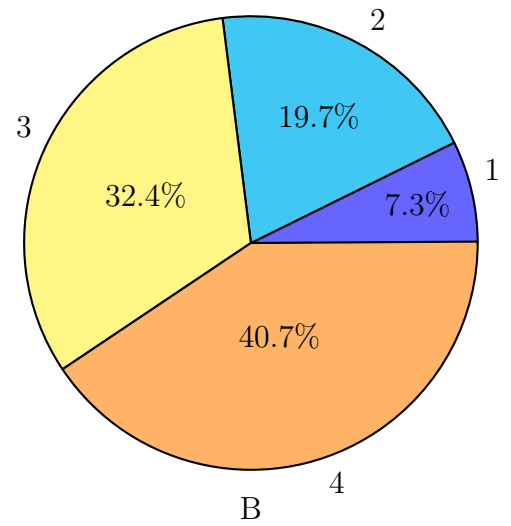
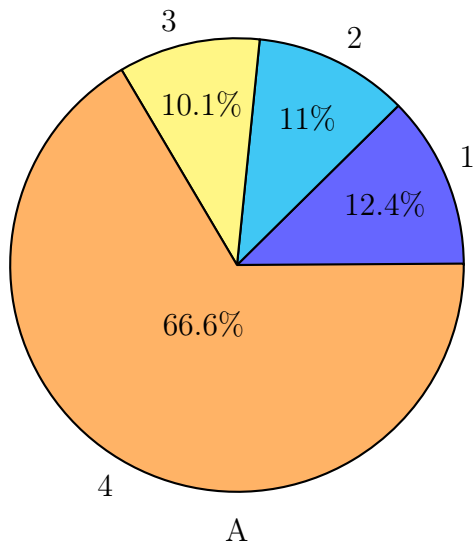
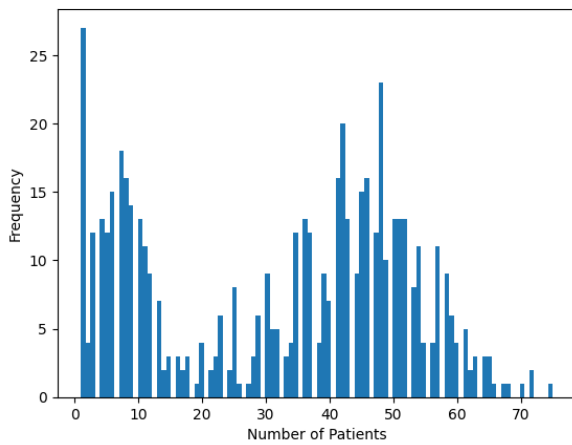
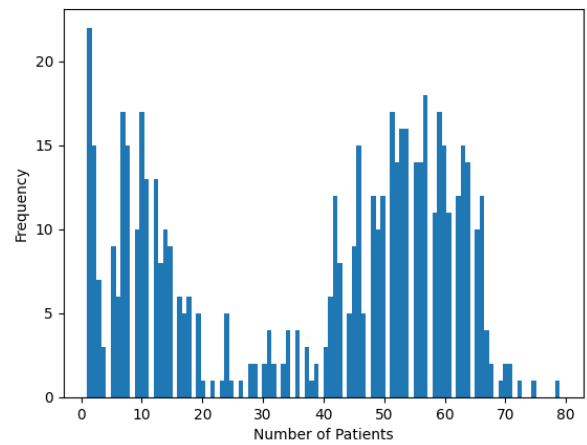


Figure 6.5: Demand Percentages of Priority Levels at Hospital 1 in (A) and Hospital 2 in (B)



Hospital 1



Hospital 2

Figure 6.6: Daily Distribution of Patients Arrivals

The next step will be to determine the expert's actions. The objective is to construct u_t over the time horizon based on the available data. For a specific day t , this involves identifying patients of class i who arrived on day t and are scheduled for day $t + n$. This subset is denoted as $u_t^{i,n}$.

The vector M of size I is available, where the M_i represents the estimated service time for the i -th class of patients. Given M , we can find matrix B , as previously defined in (3.9) to obtain the effect of the action taken on each day:

$$B \triangleq A(\mathbb{I}_N \otimes M^\top). \quad (6.1)$$

Based on the expert's actions, we can also find the system's states. An initial state is represented by the following equation:

$$x_0 = [0, 0, 0, \dots, 0]^\top \in \mathbb{R}^N. \quad (6.2)$$

The expression for subsequent states is the same as previously stated in (3.9), which accounts for the impact of both current and preceding actions, as follows:

$$x_{t+1} = Ax_t + Bu_t, \quad \forall t \in [0, T - 1]. \quad (6.3)$$

Using the data, we estimate $C_{\max}$, indicative of a hospital's maximum capacity. Although scheduled capacities vary across the time horizon, $C_{\max}$ is a constant, aligned with the hospital's service capacity. As previously discussed, we assume operating at levels either below or above $C_{\max}$ incurs system idle time costs, as described in (3.4). The hospital's maximum daily capacity is approximated as a value close to the average number of operational hours.

The booking horizon is defined as the smallest number for which about 65% of the patients' waiting days fall below. This number signifies the threshold beyond which patients

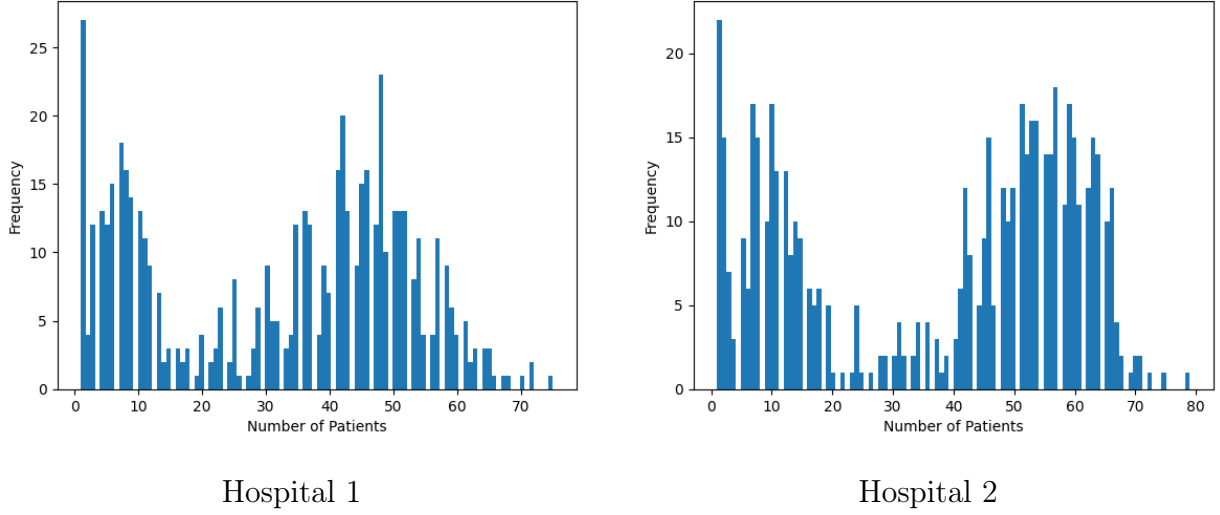


Figure 6.7: Daily Distribution of Patients Receiving Services

are considered to be rejected. However, in a real-world scenario, these patients are not actually rejected since every patient must be allocated an appointment. Instead, we consider a substantial delay in receiving the service to be perceived as a form of rejection. Since some patients do not receive immediate service and are scheduled outside the window, we label them non-urgent patients or rejected patients.

The selection of the thresholds is based on two considerations: firstly, the majority of patients do not experience rejection; secondly, this threshold strikes a balance between the costs associated with rejection and waiting. Furthermore, it distinguishes urgent patients from the rest of the patients by setting a threshold. The time horizon, T , is 500 for both hospitals, corresponding to the 500 days of patient arrivals observed in Hospitals 1 and 2. Figure 6.8 illustrates the distribution of waiting times for patients at two hospitals.

By using the recovered states $x_{0:(T-1)}$, actions $u_{0:(T-1)}$, daily demands $D_{0:(T-1)}$, service times M , and maximum capacity $C_{\max}$, the Hamiltonian approach is employed to estimate the cost parameters associated with the expert's behaviour. We utilize Proposition 5.0.1

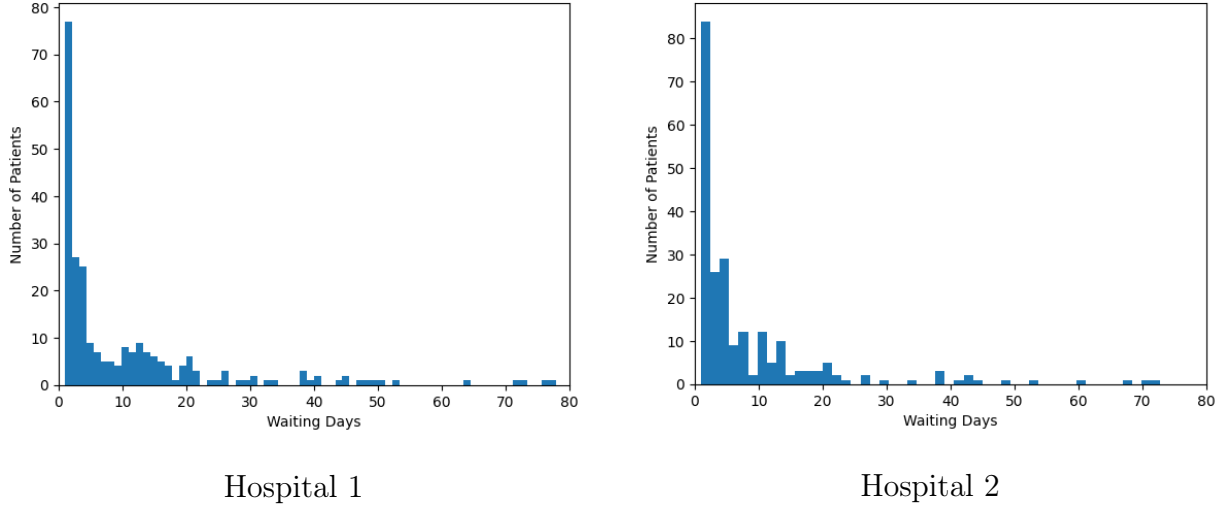


Figure 6.8: Distribution of Waiting days

to identify cost parameters using the Hamiltonian approach. The minimization problem defined in (5.9) allows us to derive values for rejection cost parameters, denoted as ρ , waiting cost parameters represented as W , and the smoothness coefficient term, denoted as δ .

In our implementations, we calculate the average waiting days for different priority levels, yielding Table 6.1. Table 6.1 indicates that patients with priority 1 generally experience shorter waiting times than those with other priorities. This implies that priority 1 patients are assigned higher cost parameters, leading to their prioritization in treatment. Consequently, this prioritization results in a reduced average waiting time for priority 1 patients.

Table 6.2 shows the number of rejected patients. We anticipate that classes with a lower percentage of rejected patients correlate with higher rejection cost parameters. Consider a scenario where priority 1 has higher rejection costs than priority 2. Decision-makers are more likely to schedule these patients within the booking horizon than to reject them

Hospital	Priority Level	Average Waiting Days
1	1	0.5
1	2	0.9
1	3	4.5
1	4	31.3
2	1	0.7
2	2	1.9
2	3	15.9
2	4	43.0

Table 6.1: Average Waiting Days for Hospitals 1 and 2

because of the higher rejection cost parameter. However, in rare cases where the percentage of rejected patients is exceptionally low, this statement may not be entirely accurate. Overtime costs and increased patient arrivals might cause more rejections from class 1 on certain days, so this anticipation is not mathematically precise. However, this approach remains a useful tool for generally anticipating rejection cost parameters.

Based on Table 6.2, at Hospital 1, the rejected cost parameter for classes 1 and 2 cannot be precisely estimated and our implementation, utilizing the Python NumPy solver, indicated a significantly high rejection cost for classes 1 and 2. This arises from the absence of a significant number of patients being rejected from these two classes. In such cases, we estimate this cost effectively towards infinity. Based on the results provided in Table 6.2, class 2 presented a large rejection cost parameter that resulted in no patient being rejected. Similarly, the high rejection cost parameter in class 1 led to a minimal number of rejections, i.e., only seven out of 1561 patients.

Hospital	Priority Level	Number of Rejected Patients	% of Rejected Patients	Total Number of Patients
1	1	2	0.1%	2215
1	2	0	0%	1958
1	3	35	1.9%	1801
1	4	6979	58.8%	11886
2	1	7	0.4%	1561
2	2	36	0.9%	4219
2	3	1746	25%	6936
2	4	6659	76%	8719
Total Patients in Hospital 1			17860	
Total Patients in Hospital 2			21435	

Table 6.2: Rejected Patients Based on Priorities in Hospital 1 and 2

Based on the data provided in Tables 6.2 and 6.3, the behaviour of the decision maker at Hospital 2 aligns well with our model. The lower percentage of patient rejections from priority 1 is a result of higher rejection costs for this class, as the inferred cost parameters suggest. Notably, the rejection cost for class 1 is higher than those for other classes. This trend continues with classes 2, 3, and 4, in the same order as the percentages of rejected patients, thereby validating our model’s accuracy in explaining the decision maker’s intentions.

Turning to the waiting cost analysis, Table 6.1 presents the average waiting days, and Table 6.3 illustrates the recovered cost parameters. In both Hospitals 1 and 2, we observe that the average number of waiting days is inversely proportional to the class priority, with higher priority classes experiencing fewer waiting days. This trend is consistent with the recovered cost analysis, where higher priority classes indicate greater waiting cost parameters, suggesting that delays in servicing these patients are more costly.

At Hospital 1, the negative waiting cost values for priorities 3 and 4 suggest a decision-

Hospital	Priority Level	Rejection Cost	Waiting Cost
Hospital 1	1	+inf	0.04
Hospital 1	2	+inf	0.02
Hospital 1	3	1.00	-0.08
Hospital 1	4	0.06	-0.62
Hospital 2	1	0.58	0.229
Hospital 2	2	0.51	0.209
Hospital 2	3	0.23	0.084
Hospital 2	4	-0.02	0.015

Table 6.3: Rejection and Waiting Costs in Hospitals 1 and 2

making preference to delay services for these classes. However, the positive rejection costs, denoted as ρ_3 and ρ_4 , imply an incentive in scheduling them beyond the booking horizon. While a small positive waiting cost might be justifiable, especially when compared with higher-priority patients, a negative value indicates a preference to schedule these classes as late as possible. This is particularly evident for class 4, where the large negative value suggests that patients are scheduled later even when capacity exists, potentially leading to a backlog and reduced service quality.

In contrast, Hospital 2 exhibits a negative rejection cost parameter for class 4, ρ_4 , as shown in Table 6.3. A negative value in the cost function can have detrimental effects on the system. We recommend a small positive value instead. The positive waiting cost for class 4, W_4 , suggests a preference to schedule non-rejected patients as soon as possible, however with a lower priority than other classes. This implies that the average waiting days for non-rejected patients from class 4 should not be excessively high, as this would result

in substantial waiting costs for the system. Regarding class 4 patients who are rejected, the negative value in the rejection cost suggests that the decision maker is inclined to reject patients from this class as much as possible. Nevertheless, the idle time cost, with a large coefficient of 1, forces the scheduling of some of the patients to avoid resource waste, especially given that the rejection cost parameter for this class is -0.03. Our analysis also reveals that non-rejected class 4 patients have an average waiting time of 10.4 days, significantly lower than the 43-day average shown in 6.1. This aligns precisely with our expectations based on the recovered cost parameters: a relatively small number of waiting days for non-rejected patients and a significantly larger number for those who are rejected. The decision-making policy for class 4 patients can be summarized as follows: the decision-maker tends to reject them as often as feasible, scheduling them beyond the booking window. However, in instances where there is available capacity in the upcoming days, some class 4 patients are scheduled for these slots to prevent resource wastage. This accounts for the significant difference between an average of 10.4 days for patient of class 4 scheduled within the booking window compared to the average of 43 days for class 4 patients. This finding underscores the effectiveness of our inverse method in interpreting patient scheduling accurately.

Comparing Hospitals 1 and 2 based on Table 6.1, we find that Hospital 1 has a higher percentage of patients of class 1. Hospital 2 demonstrates a clearer distinction between classes based on priority, as indicated by the waiting cost parameters being comparable to the idle time coefficient, which is equal to 1. This is further evidenced by more pronounced differences in waiting times across classes. The policy of Hospital 2 suggests a preference for adhering to schedules based on waiting and rejection costs rather than filling empty appointment slots. At Hospital 1, as shown in Table 6.3, the waiting cost parameters for classes 1 and 2 are relatively small positive, while for class 3 it is relatively small

negative. Given the results from Section 4.2, the idle time coefficient is equal to 1, which is substantially larger than the waiting cost parameters of classes 1, 2, and 3. It implies that Hospital 1 places a higher emphasis on minimizing idle time rather than scheduling based on class priority. This indicates that the decision maker is more focused on efficiently utilizing resources to prevent both resource wastage and over-scheduling of patients. The small waiting cost parameters for classes 1 and 2 imply a strategy of scheduling these classes sooner rather than later while avoiding overscheduling. The large negative waiting cost for class 4 reflects the longer average waiting times for the scheduled patients of this class. Notably, the waiting cost for class 4 exhibits a negative value. This implies that experts have an incentive to schedule these patients as late as possible. However, their rejection costs remain positive. This suggests that experts are inclined towards scheduling rather than rejecting class 4 patients but still aim to delay their scheduling as much as possible. This strategic scheduling grants them flexibility on subsequent dates to accommodate more urgent patients efficiently.

As evidence from Table 6.1 shows, the average waiting days for priority 4 patients surpass the expected value of 28 days for class 4. This departure from expectations warrants scrutiny and potential adjustments in the scheduling process.

6.3 Counterfactual Analysis

We have estimated the optimal cost parameters via the least squares method, accurately reflecting the behaviour of experts in Hospitals 1 and 2 in Section 6.2. This leads to questions about the consistency of experts in their decision-making across the time horizon, and if their priorities and associated cost parameters remain unchanged over various days. Although experts often prioritize patients from class 1 over class 4, the degree of this

priority can fluctuate, varying at times or even shifting priorities throughout the time horizon. Consistency in decision-making over the time horizon is vital to maximize the efficiency of the system.

In this section, we examine the impact of inconsistency in scheduling decisions on the performance of the system. We assume that the cost parameters estimated in Table 6.3 are the optimal ones used by the expert. An alternative method we employ is as follows: With these optimal cost parameters, we use the Riccati forward approach to solve the unconstrained forward problem detailed in (3.12). Under the same daily demand, we analyze what might have happened if the experts strictly followed their cost functions. By generating an alternative set of actions based on identical daily demands observed in the data, we derive the alternative actions. This alternative trajectory is then compared with the actual path taken by decision-makers in scheduling patients based on the total cost the system incurred.

We consider the same $C_{\max}$ (maximum capacity) and service times as those used by the experts. Considering that we cannot predict future demands in the real world, our alternative approach is based on the assumption that we know only the expected value of demand, which can be determined with sufficient information. In this analysis, rejection cost parameters, waiting cost parameters, and delta are similar to the expert's recovered cost parameters. Based on these values, we obtain action matrices and then get the optimal actions. To determine the action matrices, we apply Lemma 1 to acquire the matrices V , W_D , and the vector W_E , all of which are functions of the cost parameters. Additionally, with these action matrices, along with the daily demands and states of the system, we derive an alternative set of actions.

It is worth noting that our forward approach, rooted in the Riccati method, is unconstrained, potentially resulting in negative values. In the alternative path trajectory,

negative values are adjusted to zero, aligning it with practical real-world scheduling scenarios. Nevertheless, our approach has yielded significant improvements for both Hospitals 1 and 2, leading to a substantial reduction in total costs. The results are discussed in Table 6.4.

To develop an alternative policy, we employ the results of our inverse approaches. Firstly, we apply the findings from the Hamiltonian approach, and secondly, we use the outcomes of the Riccati approach. In assessing the Hamiltonian alternative policy, we compare the total costs resulting from experts' actions with those created from the alternative actions. Table 6.4 indicates using the alternative approach led to a significant cost reduction: approximately 15% for Hospital 1 and 62% for Hospital 2, using the Hamiltonian approach for the inverse problem and the Riccati forward approach. It is worth noting that the total cost is found based on (3.6).

To derive an additional alternative policy, we use the results obtained from the Riccati inverse approach. While we do not estimate the cost parameters using the Riccati inverse approach, it is still feasible to apply it in estimating the action matrices V , W_D , and W_E using Proposition 4.1.1. By learning the expert's policy represented by these matrices, even without directly determining the cost parameters, we can predict the expert's optimal actions based on the state and demand, as outlined in Lemma 1. Given that obtaining the optimal action requires only the pair of matrices V , W_D , and W_E , we can employ the following equation, as already stated in Lemma 1, to determine the optimal action:

$$u_t = Vx_t + W_D D_t + W_E, \quad \forall t \in [0 : T - 1] \quad (6.4)$$

We note that both alternative path trajectories are generated based on (6.4). In the Hamiltonian approach, we utilize the estimated cost parameters to generate the action matrices. Conversely, in the Riccati approach, we bypass learning the cost parameters

and directly learn the action matrices using Proposition 4.1.1. Using these matrices, we then derive the optimal action. Employing the Riccati inverse approach results in a 29% reduction in total cost compared to the expert’s actions, and specifically for Hospital 1, the Riccati inverse approach yields a 32% decrease in total cost.

The table below presents a comparison of the percentage improvement in total costs, as computed based on (3.6), for Hospitals 1 and 2 using the Ricatti and Hamiltonian approaches.

Hospital	Ricatti Approach	Hamiltonian Approach
Hospital 1	32 %	15%
Hospital 2	29%	62%

Table 6.4: Comparison of Percentage Improvement in Total Costs for Hospitals 1 and 2 Using Ricatti and Hamiltonian Approaches

Chapter 7

Concluding Remarks

In recent years, advance scheduling has received increased attention, particularly post-pandemic. With limited resources and an increasing patient load, scheduling efficiency is more important than ever. Although many studies have proposed various scheduling patterns and policies, there is a lack of investigation into the underlying costs that influence the decision-making of service providers. Our study addresses this gap. Our approach enables comprehensive analysis of current scheduling processes and offers constructive feedback for efficiency enhancement.

Initially, we formulated a multi-class patient scheduling problem. Both the Hamiltonian and Riccati approaches enable the estimation of a unique set of cost parameters utilized by decision-makers, applicable in both noisy and noiseless environments as detailed in Chapters 4 and 5. Utilizing the Riccati Approach, we create a simulation model mimicking a decision-maker’s strategy for scheduling patients. This model serves to validate our method’s robustness against common disturbances, variations in data size, and extensions of the booking horizon, as detailed in Section 6. Following this, we apply our method to

real-world data, estimating the cost parameters using both the Hamiltonian and Riccati approaches in Section 6.2. Using these estimated parameters, we analyze the root causes of the observed outcomes, assess their impact on scheduling, and suggest recommendations for enhancing efficiency. Finally, through Counterfactual Analysis detailed in section 6.3, we propose alternative scheduling policies that reduce total costs, even with the original cost parameters used by the decision-makers.

Future research could involve applying our methodology to various hospitals to recover their distinct cost parameters. Analyzing the characteristics and scheduling patterns of different decision-makers will offer valuable insights and is likely to garner considerable attention. Identifying the most effective decision-makers and replacing the ineffective ones is one particularly exciting prospect. Additionally, forthcoming work in advance scheduling will focus on bridging the gap between the forward and inverse approaches. We aim to further enhance system efficiency by suggesting various alternative policies derived from the recovered underlying cost function. Additionally, utilizing the Hamiltonian approach, we plan to explore modifications to the cost function, experimenting with forms beyond the quadratic cost function.

References

- [1] 100k-plus waiting longer than recommended, 2023. Accessed: Date.
- [2] Nematollah Ab Azar, Aref Shahmansoorian, and Mohsen Davoudi. From inverse optimal control to inverse reinforcement learning: A historical review. *Annual Reviews in Control*, 50:119–138, 2020.
- [3] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1, 2004.
- [4] Hossein Abouee Mehrizi, Mohamad Kazem Shirani Faradonbeh, and Mohamad Sadegh Shirani Faradonbeh. Asymptotic analysis of multi-class advance patient scheduling. *Available at SSRN 4055715*, 2022.
- [5] Stephen Adams, Tyler Cody, and Peter A Beling. A survey of inverse reinforcement learning. *Artificial Intelligence Review*, 55(6):4307–4346, 2022.
- [6] Saurabh Arora and Prashant Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, 297:103500, 2021.

- [7] Dimitri Bertsekas. *Dynamic programming and optimal control: Volume I*, volume 4. Athena scientific, 2012.
- [8] Francisco Cruz, Sven Magg, Cornelius Weber, and Stefan Wermter. Training agents with interactive reinforcement learning and contextual affordances. *IEEE Transactions on Cognitive and Developmental Systems*, 8(4):271–284, 2016.
- [9] Brian Denton and Diwakar Gupta. A sequential bounding approach for optimal appointment scheduling. *IIE transactions*, 35(11):1003–1016, 2003.
- [10] Brian Denton and Diwakar Gupta. A sequential bounding approach for optimal appointment scheduling. *IIE transactions*, 35(11):1003–1016, 2003.
- [11] Graham Goodwin, María M Seron, and José A De Doná. *Constrained control and estimation: an optimisation approach*. Springer Science & Business Media, 2006.
- [12] Qingxia Kong, Shan Li, Nan Liu, Chung-Piaw Teo, and Zhenzhen Yan. Appointment scheduling under time-dependent patient no-show behavior. *Management Science*, 66(8):3480–3500, 2020.
- [13] Florian Köpf, Jairo Inga, Simon Rothfuß, Michael Flad, and Sören Hohmann. Inverse reinforcement learning for identification in linear-quadratic dynamic games. *IFAC-PapersOnLine*, 50(1):14902–14908, 2017.
- [14] Timothy L Molloy, Jairo Inga Charaja, Sören Hohmann, and Tristan Perez. *Inverse optimal control and inverse noncooperative dynamic game theory*. Springer, 2022.
- [15] Mathew Monfort, Anqi Liu, and Brian Ziebart. Intent prediction and trajectory forecasting via predictive inverse linear-quadratic regulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

- [16] Author Name. Ontario has a pandemic backlog of 1 million surgeries. one group has a prescription for change, 5 2022.
- [17] Author Name. Title of the study. <https://www.fraserinstitute.org/studies/waiting-your-turn-wait-times-for-health-care-in-canada-2022>, 2022. Accessed: YYYY-MM-DD.
- [18] Mehmet Fatih Ozkan and Yao Ma. Inverse reinforcement learning based driver behavior analysis and fuel economy assessment. In *Dynamic Systems and Control Conference*, volume 84270, page V001T02A003. American Society of Mechanical Engineers, 2020.
- [19] Mahsa Parsapour and Dana Kulić. Recovery-matrix inverse optimal control for deterministic feedforward-feedback controllers. In *2021 American control conference (ACC)*, pages 4765–4770. IEEE, 2021.
- [20] Jonathan Patrick, Martin L Puterman, and Maurice Queyranne. Dynamic multipriority patient scheduling for a diagnostic resource. *Operations research*, 56(6):1507–1525, 2008.
- [21] Jin Qi. Mitigating delays and unfairness in appointment systems. *Management Science*, 63(2):566–583, 2017.
- [22] Sascha Rosbach, Vinit James, Simon Großjohann, Silviu Homoceanu, and Stefan Roth. Driving with style: Inverse reinforcement learning in general-purpose planning for automated driving. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2658–2665. IEEE, 2019.
- [23] Navyata Sanghvi, Shinnosuke Usami, Mohit Sharma, Joachim Groeger, and Kris Kitani. Inverse reinforcement learning with explicit policy estimates, 2021.

- [24] Clifford Stein, Van-Anh Truong, and Xinshang Wang. Advance service reservations with heterogeneous customers. *Management Science*, 66(7):2929–2950, 2020.
- [25] Van-Anh Truong. Optimal advance scheduling. *Management Science*, 61(7):1584–1597, 2015.
- [26] Zheng Wu, Liting Sun, Wei Zhan, Chenyu Yang, and Masayoshi Tomizuka. Efficient sampling-based maximum entropy inverse reinforcement learning with application to autonomous driving. *IEEE Robotics and Automation Letters*, 5(4):5355–5362, 2020.
- [27] Chengpu Yu, Yao Li, Hao Fang, and Jie Chen. System identification approach for inverse optimal control of finite-horizon linear quadratic regulators. *Automatica*, 129:109636, 2021.
- [28] Han Zhang, Jack Umenberger, and Xiaoming Hu. Inverse optimal control for discrete-time finite-horizon linear quadratic regulators. *Automatica*, 110:108593, 2019.
- [29] Shao Zhifei and Er Meng Joo. A survey of inverse reinforcement learning techniques. *International Journal of Intelligent Computing and Cybernetics*, 5(3):293–311, 2012.
- [30] Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3*, AAAI’08, page 1433–1438. AAAI Press, 2008.

APPENDICES

Appendix A

Proofs

A.1 Proof of Lemma (1)

In this section, we prove Lemma (1) using the cost-to-go function (3.14) . Following are the steps involved in deriving the optimal action for each stage:

$$\begin{aligned}
u_t^* &= \arg \min_{u_t} \left(g(x_t, D_t, u_t) + \mathbb{E}[\bar{J}(Ax_t + Bu_t, D_t)|x_t] \right) \\
&= \arg \min_{u_t} \left(C_t + 2H_t u_t + 2Fx_t + 2x_t^\top Gu_t + x_t^\top Qx_t + u_t^\top Ru_t + \mathbb{E}[\bar{J}(Ax_t + Bu_t, D_t)|x_t] \right) \\
&= \arg \min_{u_t} \left(C_t + 2H_t u_t + 2Fx_t + 2x_t^\top Gu_t + x_t^\top Qx_t + u_t^\top Ru_t + x_{t+1}^\top Kx_{t+1} + 2Ex_{t+1} \right. \\
&\quad \left. + C_{t+1} - (EB + H_{t+1}) (B^\top KB + R)^{-1} (EB + H_{t+1})^\top \right) \\
&= \arg \min_{u_t} \left(C_t + 2H_t u_t + 2Fx_t + 2x_t^\top Gu_t + x_t^\top Qx_t + u_t^\top Ru_t \right. \\
&\quad \left. + (Ax_t + Bu_t)^\top K(Ax_t + Bu_t) + 2E(Ax_t + Bu_t) + C_{t+1} \right. \\
&\quad \left. - (EB + H_{t+1}) (B^\top KB + R)^{-1} (EB + H_{t+1})^\top \right)
\end{aligned} \tag{A.1}$$

where,

$$K = Q + A^\top KA - (A^\top KB + G)(B^\top KB + \mathcal{R})^{-1}(A^\top KB + G)^\top, \tag{A.2a}$$

$$E = (F + \mathbb{E}[H_t]V)(\mathbb{I}_N - A - BV)^{-1}. \tag{A.2b}$$

To find the optimal action, u_t^* , we compute the derivative of $g(x_t, D_t, u_t) + \mathbb{E}[\bar{J}(Ax_t + Bu_t, D_t)|x_t]$ with respect to u_t and set it to zero. The derivative result is as follows:

$$2H_t^\top + 2G^\top x_t + 2Ru_t^* + 2B^\top KBu_t^* + 2B^\top KAx_t + 2B^\top E^\top = 0, \tag{A.3a}$$

$$-(R + B^\top KB)u_t^* = G^\top x_t + B^\top KAx_t + B^\top E^\top + H_t^\top \tag{A.3b}$$

$$u_t^* = -(R + B^\top KB)^{-1} \left((B^\top KA + G^\top)x_t + (EB + H)^\top \right) \tag{A.3c}$$

The state-action weight matrix is defined and denoted as V according to the following:

$$V \triangleq (R + B^\top KB)^{-1}(A^\top KB + G)^\top. \tag{A.4}$$

Additionally, to simplify our analysis, the auxiliary variable α is defined as follows:

$$\alpha \triangleq -(B^\top KB + \mathcal{R})^{-1}. \quad (\text{A.5})$$

Now, our objective is to simplify equation (A.3c) and express the optimal action, u_t^* , in terms of the state, x_t , and demand, D_t . We use the state-action weight matrix definition from (A.4) and the definition of α as described in (A.5). The following presents the reformulated form of (A.3c):

$$u_t^* = Vx_t - (R + B^\top KB)^{-1} \left((EB + H_t)^\top \right). \quad (\text{A.6})$$

By applying the definition of H_t as specified in (4.3b), we can further simplify (A.6) as follows:

$$u_t^* = Vx_t + \alpha \left(EB + \frac{1}{2} L^\top (\mathbb{I}_N \otimes W^\top) - D_t^\top P (\mathbb{1}_N^\top \otimes \mathbb{I}_I) - C_{\max} S e_1^\top (\mathbb{I}_N \otimes M^\top) \right)^\top \quad (\text{A.7a})$$

$$u_t^* = Vx_t + \alpha \left(B^\top E^\top + \left(\frac{1}{2} L^\top (\mathbb{I}_N \otimes W^\top) - C_{\max} S e_1^\top (\mathbb{I}_N \otimes M^\top) \right)^\top - \right. \quad (\text{A.7b})$$

$$\left. (\mathbb{1}_N^\top \otimes \mathbb{I}_I)^\top P^\top D_t \right)$$

$$u_t^* = Vx_t - \alpha (\mathbb{1}_N^\top \otimes \mathbb{I}_I)^\top P^\top D$$

$$+ \alpha \left(B^\top E^\top + \left(\frac{1}{2} L^\top (\mathbb{I}_N \otimes W^\top) - C_{\max} S e_1^\top (\mathbb{I}_N \otimes M^\top) \right)^\top \right) \quad (\text{A.7c})$$

New variables can now be introduced. Firstly, the demand-action weight matrix is defined as W_D in equation (A.8b). Secondly, the time-invariant action component is denoted as W_E , containing all remaining variables, as detailed in equation (A.8a).

$$W_E = \alpha \left(B^\top E^\top + \left(\frac{1}{2} L^\top (\mathbb{I}_N \otimes W^\top) - C_{\max} S e_1^\top (\mathbb{I}_N \otimes M^\top) \right)^\top \right), \quad (\text{A.8a})$$

$$W_D = -\alpha (\mathbb{1}_N^\top \otimes \mathbb{I}_I)^\top P^\top, \quad (\text{A.8b})$$

The dimensions of the matrices are as follows: V is $NI \times N$, W_D is $NI \times I$, and W_E is $NI \times 1$. Ultimately, the outcome of Lemma 1 is derived, presenting the optimal action as a function of the current state and the daily demand, complemented by a constant action component. The optimal action is expressed as follows:

$$u_i = Vx_i + W_D D_i + W_E, \quad \forall i \in [0, T-1]. \quad (\text{A.9})$$

A.2 Proof of Proposition 4.1.1

We define three matrices for the process of combining equations which include the output matrix U , the matrix of multipliers W , and the input matrix Y . The input matrix consists of the system inputs such as states and demands of the system. The output matrix consists of the expert's actions and represents the concatenated actions $u_{0:(T-1)}$. For defining matrix Y , we first define matrices X and D representing the concatenated states $x_{0:(T-1)}$, and concatenated daily demands $D_{0:(T-1)}$. Each column corresponds to actions, states, or demands on a specific day. These matrices are defined as follows:

$$U \triangleq \begin{bmatrix} u_1 & u_2 & u_3 & \cdots & u_T \end{bmatrix}, \quad (\text{A.10a})$$

$$X \triangleq \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_T \end{bmatrix}, \quad (\text{A.10b})$$

$$D \triangleq \begin{bmatrix} D_1 & D_2 & D_3 & \cdots & D_T \end{bmatrix}. \quad (\text{A.10c})$$

Now, we define the matrix Y , in which we stack the concatenated state, X , and the concatenated demand, D , and an additional row of ones. This arrangement facilitates the

rewriting of all the equations outlined in (4.6). Matrix Y is defined as follows:

$$Y = \begin{pmatrix} X \\ D \\ \mathbb{1}_T \end{pmatrix}. \quad (\text{A.11})$$

The matrix W is defined as the concatenation of the action matrices, as follows:

$$W = \begin{pmatrix} V & W_D & W_E \end{pmatrix}. \quad (\text{A.12})$$

Equations (4.6) can be reformulated using the definitions of Y in (A.11) and W in (A.12), as follows:

$$U = W Y = \begin{pmatrix} V & W_D & W_E \end{pmatrix} \begin{pmatrix} X \\ D \\ \mathbb{1}_T \end{pmatrix}. \quad (\text{A.13})$$

(A.13) is just the compact form of equations in (4.6). In the presence of noise in the optimal action, as discussed in (3.19), we observe some disturbances in (A.13). Consequently, if not all actions are optimal, (A.13) changes as follows:

$$U = WY + \bar{\zeta}. \quad (\text{A.14})$$

In (3.19), matrices Y and U are known. However, W is unknown and can be estimated by the following equation:

$$\min_W \|U - WY\|_F. \quad (\text{A.15})$$

Assuming the Y is of full rank we can find the action matrices uniquely. This assumption reflects the richness of the data. By applying the least-squares approach, we can solve (A.15) and estimate the W .

A.3 Proof of Proposition 4.4.1

By reorganizing (4.11b), we isolate RV , which represents the only unknown component of (4.11b). We can then state the following equation:

$$B^\top KBV + RV = -(A^\top KB + G)^\top, \quad (\text{A.16a})$$

$$RV = -B^\top KBV - B^\top KBA - G^\top, \quad (\text{A.16b})$$

$$RV = -B^\top K(BV + A) - G^\top, \quad (\text{A.16c})$$

$$RV = -G^\top - B^\top K(\theta + A). \quad (\text{A.16d})$$

In (A.16d), matrix K is known as given in Lemma 3. Consequently, all matrices V , K , G , and θ , are known. However, R is a function of rejection cost parameters which is unknown. As a next step, we use the specific structure of R defined in (4.3f) to rewrite (A.16d) as follows:

$$((\mathbb{1}_N \otimes \mathbb{I}_I)P(\mathbb{1}_N^\top \otimes \mathbb{I}_I) + (\mathbb{I}_N \otimes M^\top)^\top e_1 e_1^\top (\mathbb{I}_N \otimes M^\top) + \delta \mathbb{I}_{NI})V \quad (\text{A.17a})$$

$$= -B^\top K(\theta + A) - G^\top,$$

$$((\mathbb{1}_N \otimes \mathbb{I}_I)P(\mathbb{1}_N^\top \otimes \mathbb{I}_I) + \delta \mathbb{I}_{NI})V = \quad (\text{A.17b})$$

$$- ((\mathbb{I}_N \otimes M^\top)^\top e_1 e_1^\top (\mathbb{I}_N \otimes M^\top))V - B^\top K(\theta + A) - G^\top. \quad (\text{A.17c})$$

In (A.17c), the rejection cost matrix, P , and smoothness coefficient term δ should be estimated. Additionally, the vector M , which denotes service times, is known. Based on (A.17c), we find a unique solution for the rejection cost matrix, P , and δ . Hence, if there is no noise in the observations of the expert's actions or the system dynamics, the values of P and δ will be recovered uniquely. In the presence of noise, we propose an optimization-based approach to estimate these parameters by leveraging the redundancy of Equations

(A.17c).

To simplify the process, we introduce auxiliary matrices C_1 and C_2 for determining P and δ in (A.17c):

$$C_1 \triangleq (\mathbb{1}_N \otimes \mathbb{I}_I), \quad (\text{A.18a})$$

$$C_2 \triangleq -((\mathbb{I}_N \otimes M^\top)^\top e_1 e_1^\top (\mathbb{I}_N \otimes M^\top))V - B^\top K(\theta + A) - G^\top. \quad (\text{A.18b})$$

In (A.17c), the right-hand side of the equation is equal to C_2 and is known.

The vector $(\mathbb{1}_N)$ is a vector of ones, where N denotes the booking horizon. Additionally, $(\mathbb{I}_I)$ represents an $I \times I$ identity matrix.

The matrix C_1 is obtained through the Kronecker product of $(\mathbb{1}_N)$ and $(\mathbb{I}_I)$ as indicated in (A.18a). Now, we rewrite (A.17c) using the definition of C_1 , and C_2 :

$$\begin{aligned} (C_1 P C_1^\top + \delta \mathbb{I}_{NI})V &= C_2, \\ (C_1 P C_1^\top)V + \delta V &= C_2. \end{aligned} \quad (\text{A.19})$$

The matrix C_1 must have full column rank to simplify the equation and estimate matrix P .

Matrix C_1 has full column rank. Thus, matrix $C_1^\top C_1$ is invertible. We multiply both sides of (A.19) with $C_1^\top$. Subsequently, by multiplying both sides of (A.19) with $(C_1^\top C_1)^{-1}$, we can simplify the term $(C_1 P C_1^\top)V$ to $(P C_1^\top)V$. The simplified version of (A.19) is as follows:

$$(C_1^\top C_1)(P C_1^\top) V + \delta C_1^\top V = C_1^\top C_2, \quad (\text{A.20a})$$

$$(P C_1^\top) V + \delta (C_1^\top C_1)^{-1} C_1^\top V = (C_1^\top C_1)^{-1} C_1^\top C_2. \quad (\text{A.20b})$$

To simplify the right-hand side of (A.20b), we introduce two new known auxiliary

variables called C and C_3 :

$$C_3 \triangleq (C_1^\top C_1)^{-1} C_1^\top C_2, \quad (\text{A.21a})$$

$$C \triangleq (C_1^\top C_1)^{-1} C_1^\top V, \quad (\text{A.21b})$$

where c_{ij} , the elements of C , are given by:

$$c_{ij} = ((C_1^\top C_1)^{-1} C_1^\top V)_{ij}, \quad \forall i \in [1, I], \quad \forall j \in [1, N]. \quad (\text{A.22})$$

Variable ϕ is defined to quantify the first term, $(PC_1^\top) V$, in (A.20b):

$$\phi \triangleq (PC_1^\top) V = P \left[\underbrace{\mathbb{I}_I \quad \mathbb{I}_I \quad \cdots \quad \mathbb{I}_I}_{N \text{ times}} \right] V = \left[\underbrace{P \quad P \quad \cdots \quad P}_{N \text{ times}} \right] V, \quad (\text{A.23})$$

where ϕ_{ij} , the elements of ϕ , are given by:

$$\phi_{ij} = ((PC_1^\top) V)_{ij}, \quad \forall i \in [1, I], \quad \forall j \in [1, N]. \quad (\text{A.24})$$

The visualized version of (A.23) can be found in (B.3c). The element ϕ_{ij} determined using (A.23) can be expressed in the following form:

$$\phi_{ij} = \rho_i \sum_{k=0}^N V_{KI+i,j}, \quad \forall i \in [1, I], \quad \forall j \in [1, N]. \quad (\text{A.25a})$$

Combining the auxiliary variables, (A.20b) transforms into the following equation:

$$(PC_1^\top) V + \delta(C_1^\top C_1)^{-1} C_1^\top V = \phi + \delta C = C_3. \quad (\text{A.26})$$

We know that (A.26) is valid for every element in the matrix C_3 . Therefore,

$$\phi_{ij} + \delta c_{ij} = C_{3i,j}, \quad \forall i \in [1, I], \quad \forall j \in [1, N], \quad (\text{A.27a})$$

$$\rho_i \sum_{k=0}^N V_{KI+i,j} + \delta c_{ij} = C_{3i,j}, \quad \forall i \in [1, I], \quad \forall j \in [1, N]. \quad (\text{A.27b})$$

For technical details and to view the visual representation of the matrices of (A.26), please refer to the appendix, as detailed in (B.2).

In (A.27b), a set of equations is given consisting of $N \times I$ equations and $I + 1$ unknowns. The unknowns include the rejection cost parameters for each class and the smoothness coefficient term. In the absence of noise, by solving the set of equations, we can determine the values of $\rho_1, \rho_2, \rho_3, \dots, \rho_I$, and δ . Note that there are redundant equations in (4.14), thus, in the presence of noisy data, an optimization-based approach can be employed to estimate the cost parameters. Although not all equations may hold due to the noise, utilizing all available equations allows for better estimation of rejection cost parameters. We formulate a norm-2 optimization problem aimed at minimizing the disturbances in our equations, ultimately reducing the difference between the obtained and expected values as follows:

$$\begin{aligned} \min_{\rho_1, \rho_2, \rho_3, \dots, \rho_I, \delta} & \sum_{i=1}^I \sum_{j=1}^N (\phi_{ij} + \delta c_{ij} - C_{3i,j})^2 \\ \text{s.t.} \quad & \phi_{ij} = \rho_i \sum_{k=0}^N V_{KI+i,j}, \quad \forall i \in [1, I], \quad \forall j \in [1, N]. \end{aligned} \tag{A.28}$$

The matrix form associated with the optimization problem (A.28) can be found in the appendix, see (B.4). By utilizing this minimization problem, we can estimate the rejection cost parameters, ρ_i as well as the smoothness coefficient, δ and assuming the V is of full rank we can find the rejection cost parameters uniquely.

A.4 Proof of Lemma 2

Multiplying all cost parameters by a specific non-zero scalar, μ , also scales the cost matrices Q, F, H_t, R , and G by μ , as these matrices are linear functions of the cost parameters. Multiplying these matrices by μ results in the Riccati Gain matrix K and matrix E being scaled by μ as well. We then examine how this affects the action matrices V, W_D , and W_E . To achieve this, we revisit the definition of these action matrices as stated in Lemma 1:

$$V = -(B^\top KB + \mathcal{R})^{-1}(A^\top KB + G)^\top, \quad (\text{A.29a})$$

$$W_E = -(B^\top KB + \mathcal{R})^{-1} \left(B^\top E^\top + \left(\frac{1}{2} L^\top (\mathbb{I}_N \otimes W^\top) - C_{\max} S e_1^\top (\mathbb{I}_N \otimes M^\top) \right)^\top \right), \quad (\text{A.29b})$$

$$W_D = (B^\top KB + \mathcal{R})^{-1} (\mathbb{1}_N^\top \otimes \mathbb{I}_I)^\top P^\top. \quad (\text{A.29c})$$

For matrix V , the first term $-(B^\top KB + \mathcal{R})^{-1}$ is scaled by $\frac{1}{\mu}$, and the second term $(A^\top KB + G)^\top$ is scaled by μ , as both matrices K and G are scaled by μ . Consequently, V remains unchanged, since $\frac{1}{\mu} \times \mu = 1$.

For the vector W_E , a similar argument applies. The term $-(B^\top KB + \mathcal{R})^{-1}$ is scaled by $\frac{1}{\mu}$. The second term, involving various parameters, each scaled by μ , will have their summation also scaled by μ . Therefore, this second term is scaled by μ , resulting in vector W_E remaining unchanged, as the multiplicative effect of $\frac{1}{\mu} \times \mu$ equals 1.

Regarding the matrix W_D , the term $-(B^\top KB + \mathcal{R})^{-1}$ is scaled by $\frac{1}{\mu}$, while the term P is scaled by μ . As a result, W_D also remains unchanged, since the scaling factors $\frac{1}{\mu} \times \mu$ cancel out to 1.

A.5 Proof of Lemma 3

The term $(B^\top KB + \mathcal{R})^{-1}(A^\top KB + G)^\top$ from Lemma 1 is equivalent to the state-action weight matrix, V . This value has already been estimated using least-squares, as outlined in (4.1.1). For simplicity, the auxiliary variable θ is defined as a known matrix. Equation (4.8) can be further simplified as a step to find K :

$$K = Q + A^\top KA + (A^\top KB + G)V, \quad (\text{A.30a})$$

$$\theta \triangleq BV, \quad (\text{A.30b})$$

$$K = Q + A^\top KA + A^\top K\theta + GV. \quad (\text{A.30c})$$

In the aforementioned (A.30c), the only unknown matrix is K . Thus, K can be obtained using this equation. By finding the vectorized value of both sides of (A.30c), we get:

$$\text{vec}(K) = \text{vec}(Q) + \text{vec}(A^\top KA) + \text{vec}(A^\top K\theta) + \text{vec}(GV). \quad (\text{A.31})$$

[?] proves the following equation that reformulates $\text{vec}(ABC)$:

$$\text{vec}(ABC) = (C^\top \otimes A) \text{vec}(B). \quad (\text{A.32})$$

Utilizing (A.32), in (A.31), we replace $\text{vec}(A^\top KA)$ with $(A^\top \otimes A^\top) \text{vec}(K)$, and apply a similar transformation to $\text{vec}(A^\top K\theta)$:

$$\text{vec}(K) = \text{vec}(Q) + (A^\top \otimes A^\top) \text{vec}(K) + (\theta^\top \otimes A^\top) \text{vec}(K) + \text{vec}(GV), \quad (\text{A.33a})$$

$$(\mathbb{I}_{N \times N} - (A^\top \otimes A^\top) - (\theta^\top \otimes A^\top)) \text{vec}(K) = \text{vec}(Q) + \text{vec}(GV). \quad (\text{A.33b})$$

On the left-hand side of (A.33b), the only unknown vector is $\text{vec}(K)$, while its multiplier, $(\mathbb{I}_{N \times N} - (A^\top \otimes A^\top) - (\theta^\top \otimes A^\top))$, is known as all terms inside it are known. Further simplification of this equation allows us to solve for the matrix K :

$$\text{vec}(K) = (\mathbb{I}_{N \times N} - (A^\top \otimes A^\top) - (\theta^\top \otimes A^\top))^{-1}(\text{vec}(Q) + \text{vec}(GV)). \quad (\text{A.34})$$

The inverse of $(I - ((A+O)^\top \otimes A^\top))$ exists, as evidenced by the presence of an expert. This is because the existence of an expert implies a unique solution K for the Riccati equation outlined in 1.

A.6 Proof of Proposition 4.4.2

By observing the visualization of P in (B.1), we can conclude that this matrix is symmetric since only non-zero elements are on its diagonal. Hence, $P^\top = P$. Using the definition of C_1 in (4.12), (4.16) can be simplified by following:

$$(B^\top KB + \mathcal{R})W_D = C_1 P, \quad (\text{A.35a})$$

$$-RW_D + C_1 P = (B^\top KB)W_D. \quad (\text{A.35b})$$

Equation (4.3f) describes the special structure of R as follows:

$$R = (\mathbb{1}_N \otimes \mathbb{I}_I)P(\mathbb{1}_N^\top \otimes \mathbb{I}_I) + (\mathbb{I}_N \otimes M^\top)^\top e_1 e_1^\top (\mathbb{I}_N \otimes M^\top) + \delta \mathbb{I}_{NI}. \quad (\text{A.36})$$

After using the special structure of R , we can derive the following result:

$$-(C_1 P C_1^\top + (\mathbb{I}_N \otimes M^\top)^\top e_1 e_1^\top (\mathbb{I}_N \otimes M^\top) + \delta \mathbb{I}_{NI})W_D + C_1 P = (B^\top KB)W_D, \quad (\text{A.37a})$$

$$-C_1 P C_1^\top W_D - \delta W_D + C_1 P = (B^\top KB)W_D - ((\mathbb{I}_N \otimes M^\top)^\top e_1 e_1^\top (\mathbb{I}_N \otimes M^\top))W_D. \quad (\text{A.37b})$$

For the sake of simplicity, the auxiliary variable γ covers all known terms on the right-hand side of the (A.37b):

$$\gamma \triangleq (B^\top KB)W_D - ((\mathbb{I}_N \otimes M^\top)^\top e_1 e_1^\top (\mathbb{I}_N \otimes M^\top))W_D, \quad (\text{A.38a})$$

$$C_1 P(\mathbb{I}_I - C_1^\top W_D) + \delta W_D = \gamma. \quad (\text{A.38b})$$

Next, by first multiplying both sides of (A.38b) by $(C_1)^\top$ and then by $(C_1^\top C_1)^{-1}$, we can further simplify (A.38b) as follows:

$$P(\mathbb{I}_I - C_1^\top W_D) + \delta(C_1^\top C_1)^{-1}C_1^\top W_D = (C_1^\top C_1)^{-1}C_1^\top \gamma. \quad (\text{A.39})$$

In (A.39), the unknown variables are P and δ , while all other variables are known. Our objective is to estimate these two unknown variables using (A.39).

We introduce a known auxiliary variable Θ that assists us in solving the equation (A.39):

$$\Theta \triangleq (C_1^\top C_1)^{-1}C_1^\top W_D, \quad (\text{A.40})$$

where Θ_{ij} , the elements of Θ , are given by:

$$\Theta_{ij} = ((C_1^\top C_1)^{-1}C_1^\top W_D)_{ij}, \quad \forall i \in [1, I], \quad \forall j \in [1, I]. \quad (\text{A.41})$$

Now, we can proceed to rewrite the (A.39) as a step of determining the rejection cost parameters and delta:

$$C_4 \triangleq P(\mathbb{I}_I - C_1^\top W_D) + \delta(C_1^\top C_1)^{-1}C_1^\top W_D = (C_1^\top C_1)^{-1}C_1^\top \gamma. \quad (\text{A.42})$$

We define a new known matrix called $\bar{\Theta}$ to simplify the calculations, as follows:

$$\bar{\Theta} \triangleq \mathbb{I}_I - C_1^\top W_D. \quad (\text{A.43})$$

Utilizing the mathematical formulations presented in the appendix (see (B.4)), we reformulate (A.43) as follows:

$$\rho_i \bar{\Theta}_{ij} + \delta \Theta_{ij} = C_{4ij}, \quad \forall i \in [1, I], \quad \forall j \in [1, I]. \quad (\text{A.44})$$

In (A.44), a set of equations is given consisting of $I \times I$ equations and $I + 1$ unknowns. Using the set of equations, we can determine the values for the rejection cost parameters,

ρ , and smoothness coefficient term, δ and assuming the W_D is of full rank we can find the rejection cost parameters uniquely. Even in cases where there is noise present in the dynamics of the system or in the observations from the expert, we can still employ the optimization approach. Similar to Section 4.4.1, we construct a norm-2 optimization problem to minimize disturbances in our equations. This approach is designed to reduce the difference between the obtained and expected values as follows:

$$\min_{\rho_1, \rho_2, \rho_3, \dots, \rho_I, \delta} \sum_{i=1}^I \sum_{j=1}^I (\rho_i \bar{\Theta}_{ij} + \delta \Theta_{ij} - C_{4i,j})^2, \quad \forall i \in [1, I], \quad \forall j \in [1, I] \quad . \quad (\text{A.45})$$

The matrix representation of the optimization problem referenced in (A.55) can be found in the appendix, see (B.6a).

A.7 Proof Proposition 4.5.1

It is worth noting that the matrix K is estimated in (A.34), R is determined in (4.20), and the time-invariant action component W_E is estimated in (4.1.1). Consequently, the left-hand side of (4.22) can be estimated. To simplify the right-hand side of (4.22), we introduce the auxiliary variable C_5 :

$$C_5 \triangleq B^\top E^\top + \left(\frac{1}{2} L^\top (\mathbb{I}_N \otimes W^\top) - C_{\max} e_1^\top (\mathbb{I}_N \otimes M^\top)\right)^\top. \quad (\text{A.46})$$

Next, we aim to find W using C_5 and the special structure of E as given in (4.5b). Using definition of C_5 and E , we can rewrite (A.46) as follows:

$$\begin{aligned} C_5 &= B^\top (\mathbb{I}_N - A - BV)^{-T} (F + \mathbb{E}[H_t]V)^\top + \\ &\quad \left(\frac{1}{2} L^\top (\mathbb{I}_N \otimes W^\top) - C_{\max} e_1^\top (\mathbb{I}_N \otimes M^\top)\right)^\top. \end{aligned} \quad (\text{A.47})$$

To further simplify (A.47), we introduce variable C_6 , determined by the state-action weight matrix, V :

$$C_6 \triangleq B^\top (\mathbb{I}_N - A - BV)^{-T}. \quad (\text{A.48})$$

In (A.47), $\mathbb{E}[H_t]$ represents the expected value of H_t , and the definition of H_t is given in (4.3b). As a result, $\mathbb{E}[H_t]$ can be written as follows:

$$\mathbb{E}[H_t] = \mathbb{E} \left[\frac{1}{2} L^\top (\mathbb{I}_N \otimes W^\top) - D_t^\top P (\mathbb{1}_N^\top \otimes \mathbb{I}_I) - C_{\max} e_1^\top (\mathbb{I}_N \otimes M^\top) \right]. \quad (\text{A.49})$$

All variables, with the exception of the demand D_t , remain constant over time, thus their expected values are equivalent to their actual values. By calculating the expected value of the demand, denoted as $\mathbb{E}[D_t]$, we obtain a vector representing the expected demand for different classes of patients. We use the mean of the daily demand as an estimation of the expected demand. As a result, $\mathbb{E}[H_t]$ can be expressed as follows:

$$\mathbb{E}[H_t] = \frac{1}{2} L^\top (\mathbb{I}_N \otimes W^\top) - \mathbb{E}[D_t^\top] P (\mathbb{1}_N^\top \otimes \mathbb{I}_I) - C_{\max} e_1^\top (\mathbb{I}_N \otimes M^\top). \quad (\text{A.50})$$

In (A.50), our objective is to estimate W , the waiting cost vector, which is the only unknown variable. Note that L is a vector containing values from $0, 1, 2, \dots, N-1$.

To simplify (A.47), we use the expanded format $\mathbb{E}(H_t)$ derived in (A.47):

$$\begin{aligned} C_5 = & C_6 F^\top + C_6 V^\top \left(\frac{1}{2} L^\top (\mathbb{I}_N \otimes W^\top) - \mathbb{E}(D_t)^\top P (\mathbb{1}_N^\top \otimes \mathbb{I}_I) - C_{\max} e_1^\top (\mathbb{I}_N \otimes M^\top) \right)^\top + \\ & \left(\frac{1}{2} L^\top (\mathbb{I}_N \otimes W^\top) - C_{\max} e_1^\top (\mathbb{I}_N \otimes M^\top) \right)^\top. \end{aligned} \quad (\text{A.51})$$

In the next step, we shift the known variables, to the left side of (A.51) and denote it as C_7 :

$$\begin{aligned} C_7 \triangleq & C_5 - (C_6 F^\top - C_6 V^\top (\mathbb{E}[D_t^\top]^\top P (\mathbb{1}_N^\top \otimes \mathbb{I}_I) + C_{\max} e_1^\top (\mathbb{I}_N \otimes M^\top)))^\top \\ & + (C_{\max} e_1^\top (\mathbb{I}_N \otimes M^\top))^\top. \end{aligned} \quad (\text{A.52})$$

To estimate the waiting cost parameters, we employ the definition of C_7 given in (A.52) and reformulate equation (A.51) as follows:

$$C_7 = C_6 V^\top \left(\frac{1}{2} L^\top (\mathbb{I}_N \otimes W^\top) \right)^\top + \left(\frac{1}{2} L^\top (\mathbb{I}_N \otimes W^\top) \right)^\top, \quad (\text{A.53a})$$

$$C_7 = (C_6 V^\top + \mathbb{I}_{NI}) \left(\frac{1}{2} L^\top (\mathbb{I}_N \otimes W^\top) \right)^\top, \quad C_8 \triangleq (C_6 V^\top + \mathbb{I}_{NI}) \quad (\text{A.53b})$$

$$C_7 = C_8 \left(\frac{1}{2} L^\top (\mathbb{I}_N \otimes W^\top) \right)^\top. \quad (\text{A.53c})$$

We can rewrite (A.53c) as follows:

$$\begin{aligned} j &= \text{rem}(i, I) + 1, \\ n &= \lfloor \frac{i}{I} \rfloor, \\ (C_7)_k &= \sum_{i=1}^{NI} \frac{n}{2} (C_8)_{k,i} W_j, \quad \forall k \in [1, I \times N]. \end{aligned} \quad (\text{A.54})$$

In the absence of noise in our equations we can find the waiting cost parameters discussed in (A.54). There are $N \times I$ equations and I of unknowns which include the waiting cost parameters and assuming the C_8 is of full rank we can find the waiting cost parameters uniquely. However, in the presence of noise, we construct a norm-2 optimization problem to minimize disturbances in our equations in (A.54). This approach is designed to reduce the difference between the obtained and expected values as follows:

$$\min_{W_1, W_2, W_3, \dots, W_I} \sum_{k=1}^{NI} \left((C_7)_k - \sum_{i=1}^{NI} \frac{n}{2} (C_8)_{k,i} W_j \right)^2. \quad (\text{A.55})$$

where,

$$j = \text{rem}(i, I) + 1, \quad (\text{A.56})$$

$$n = \lfloor \frac{i}{I} \rfloor. \quad (\text{A.57})$$

A.8 Proof of Proposition 5.0.1

We apply the outcome of the Hamiltonian condition (see Condition 4) and compute the derivatives of $\mathbb{H}_k$ with respect to u_k :

$$\begin{aligned} \frac{\partial \mathbb{H}_k}{\partial u_k} &= 0, \\ &= \frac{\partial(C_k + 2H_k u_k + 2F x_k + 2x_k^\top G u_k + x_k^\top Q x_k + u_k^\top R u_k + \lambda_k^\top (A x_k + B u_k))}{\partial u_k}, \quad (\text{A.58}) \\ &= 2H_k^\top + 2G^\top x_k + 2R u_k + B^\top \lambda_k = 0, \quad \forall k \in [0, T-1]. \end{aligned}$$

In the above equation, x_k and u_k are known since we observe the states and actions of the expert. Only two variables are unknown: vector H_k and matrix R . We aim to find rejection cost parameters as well as the waiting cost parameters using vector H_k and matrix R . (A.58) can be simplified in the following equations:

$$2H_k^\top + 2R u_k = -2G^\top x_k - B^\top \lambda_k, \quad \forall k \in [0, T-1]. \quad (\text{A.59})$$

R , H_k given by (4.3b) and (4.3f) are restated in the following equations:

$$H_k = \frac{1}{2} L^\top (\mathbb{I}_N \otimes W^\top) - D_k^\top P (\mathbf{1}_N^\top \otimes \mathbb{I}_I) - C_{\max} e_1^\top (\mathbb{I}_N \otimes M^\top), \quad (\text{A.60a})$$

$$R = (\mathbf{1}_N \otimes \mathbb{I}_I) P (\mathbf{1}_N^\top \otimes \mathbb{I}_I) + (\mathbb{I}_N \otimes M^\top)^\top e_1 e_1^\top (\mathbb{I}_N \otimes M^\top) + \delta \mathbb{I}_{NI}. \quad (\text{A.60b})$$

In H_k , there are three components. The initial two components are unknown as they rely on the waiting cost vector W and the rejection cost matrix P . However, the final component is known and relies on the known service time vector, denoted as M .

In R , we have three components. the first term depends on the rejection cost matrix (P) and the third term depends on δ which is unknown. However, the second term is known since it only depends on M for different classes of patients.

For finding the cost parameters, P and W and δ , in (A.59), we shift the known parameters to the right-hand side of (A.59) and the unknowns to the left-hand side. We introduce the auxiliary variable $\bar{\kappa}_k$ to include all known parameters on the right-hand side of (A.59):

$$\bar{\kappa}_k \triangleq -2G^\top x_k - B^\top \lambda_k + 2C_{\max} e_1^\top (\mathbb{I}_N \otimes M^\top) - 2(\mathbb{I}_N \otimes M^\top)^\top e_1 e_1^\top (\mathbb{I}_N \otimes M^\top) u_k, \quad \forall k \in [0, T-1]. \quad (\text{A.61})$$

We also introduce the auxiliary variable κ_k to include all unknown parameters on the left-hand side of (A.59):

$$\kappa_k \triangleq (I_N \otimes W)L - 2(\mathbb{1}_N \otimes I_I)PD_k + 2(\mathbb{1}_N \otimes I_I)P(\mathbb{1}_N^\top \otimes I_I)u_k + 2\delta u_k, \quad \forall k \in [0, T-1], \quad (\text{A.62})$$

where,

$$j = \text{rem}(i, I) + 1, \quad (\text{A.63a})$$

$$n = \lfloor \frac{i}{I} \rfloor, \quad (\text{A.63b})$$

$$\kappa_k^i = nW_j - 2(D_k^j - u_k^j) + \delta u_k^{j,n}, \quad \forall k \in [0, T-1], \quad \forall i \in [0, (NI-1)]. \quad (\text{A.63c})$$

The process of obtaining κ_k involves several steps. Initially, we find the vector $(I_N \otimes W)$ in (B.7), followed by the computation of the vector $2(\mathbb{1}_N \otimes I_I)PD_k$ as shown in (B.9). Subsequently, we determine the expression $2(\mathbb{1}_N \otimes I_I)P(\mathbb{1}_N^\top \otimes I_I)u_k$ in (B.10). Finally, the summation of these components, resulting in κ_k , is detailed in (B.12a), where a comprehensive visual representation and calculations are provided.

The variables denoted as κ_k in (A.62) are unknown. However, we have estimated values, denoted as $\bar{\kappa}_k$, which are useful in dealing with disturbances in the system. Our objective here is to estimate values for rejection and waiting costs that closely align with our estimation $\bar{\kappa}$. To put it simply, we aim to minimize the difference between the actual value κ and our estimation $\bar{\kappa}$ through mathematical optimization, as expressed by the following equation:

$$\min_{\rho_{1:I}, W_{1:I}, \delta} \sum_{k=0}^T \sum_{i=0}^{NI-1} (\bar{\kappa}_k^i - \kappa_k^i)^2 . \quad (\text{A.64})$$

Under the assumption that $\bar{\kappa}$ is of full rank, we can uniquely determine the cost parameters, which indicates the richness of the data.

A.9 Detailed technical reformulation of the instantaneous cost function in (4.1)

We demonstrate the equivalence between the matrix representation of instantaneous cost stated in (4.1) and its normal form as presented in (3.7). We know that the normal cost form has three main parts, waiting cost, rejection cost, and idle time cost. We start with reformulating the total waiting cost given in (3.2) and in the matrix form it is given by the term $L^\top (\mathbb{I}_N \otimes W^\top) u_t$ in (4.1). We can see the visualized simplified version of the term

$L^\top(\mathbb{I}_N \otimes W^\top)$ in (B.7). So the calculation is as follows:

$$L^\top(\mathbb{I}_N \otimes W^\top)u_t = (u_t^\top \quad L^\top(\mathbb{I}_N \otimes W^\top))^\top \quad (\text{A.65a})$$

$$= \begin{bmatrix} u_t^{1,0}, u_t^{2,0}, u_t^{3,0}, \dots, u_t^{I,0}, u_t^{1,1}, u_t^{2,1}, \dots, u_t^{I,1} \dots, u_t^{1,N-1}, \dots, u_t^{I,N-1} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_I \\ 2w_1 \\ 2w_2 \\ 2w_3 \\ \vdots \\ 2w_I \\ \vdots \\ (N-1)w_1 \\ \vdots \\ (N-1)w_I \end{bmatrix} \quad (\text{A.65b})$$

$$= \sum_{t=1}^T \sum_{i=1}^I W_i \sum_{n=0}^{N-1} n u_t^{i,n}. \quad (\text{A.65c})$$

As a result, in (A.65c) we proved that in the matrix form the total waiting cost is covered by the term $L^\top(\mathbb{I}_N \otimes W^\top)$. Now, we turn to the total rejection cost given in (3.3). For this purpose, we rewrite the total rejection cost using matrices as follows:

$$\begin{aligned} & \sum_{t=1}^T \sum_{i=1}^I \rho_i \left(D_{i,t} - \sum_{n=0}^{N-1} u_t^{i,n} \right)^2 \\ &= \sum_{t=1}^T \left(D_t - ((\mathbb{1}_N^\top \otimes I_I) u_t) \right)^\top P \left(D_t - ((\mathbb{1}_N^\top \otimes I_I) u_t) \right). \end{aligned} \quad (\text{A.66})$$

The total idle time cost defined in (3.4) can be rewritten as follows:

$$\begin{aligned} & S(C_{\max} - \sum_{j=j_{\min}}^t \sum_{i=1}^I M_i u_{i,t-j}^j)^2 = \\ & (C_{\max} \mathbb{1}_N - x_t - (\mathbb{I}_N \otimes M^\top) u_t)^\top S e_1 e_1^\top (C_{\max} \mathbb{1}_N - x_t - (\mathbb{I}_N \otimes M^\top) u_t). \end{aligned} \quad (\text{A.67})$$

As a result, we can rewrite the total cost as follows:

$$g(x_t, D_t, u_t) = C_t + 2H_t u_t + 2F x_t + 2x_t^\top G u_t + x_t^\top Q x_t + u_t^\top R u_t, \quad (\text{A.68})$$

where,

$$C_t = D_t^\top P D_t + S C_{\max}^2, \quad (\text{A.69a})$$

$$H_t = \frac{1}{2} L^\top (\mathbb{I}_N \otimes W^\top) - D_t^\top P (\mathbb{1}_N^\top \otimes \mathbb{I}_I) - C_{\max} S e_1 e_1^\top (\mathbb{I}_N \otimes M^\top), \quad (\text{A.69b})$$

$$F = -C_{\max} S e_1 e_1^\top, \quad (\text{A.69c})$$

$$G = (S e_1 e_1^\top) (\mathbb{I}_N \otimes M^\top), \quad (\text{A.69d})$$

$$Q = S e_1 e_1^\top, \quad (\text{A.69e})$$

$$R = (\mathbb{1}_N \otimes \mathbb{I}_I) P (\mathbb{1}_N^\top \otimes \mathbb{I}_I) + (\mathbb{I}_N \otimes M^\top)^\top S e_1 e_1^\top (\mathbb{I}_N \otimes M^\top) + \delta \mathbb{I}_{NI}. \quad (\text{A.69f})$$

A.10 Details on optimality of a path trajectory

In the study detailed in [11], they prove that in the Hamiltonian approach, the optimality of a path trajectory is guaranteed if the following four conditions are satisfied:

1. System Dynamics (also known as state equations): these should be aligned with the dynamics of the system:

$$x_{k+1} = Ax_k + Bu_k, \quad \forall k \in [0, T-1]. \quad (\text{A.70})$$

In the absence of disturbances in the equations, the first condition always holds because the states of the system, which capture allocated capacities, follow the system dynamics. This condition aligns with the system dynamics outlined in (3.15c).

2. Adjoint Equations: The relationship between co-states and the Hamiltonian function is as follows:

$$\lambda_{k-1} = \frac{\partial \mathbb{H}_k}{\partial x_k}, \quad \forall k \in [0, (T-1)]. \quad (\text{A.71})$$

This equation comes from the fact that in the presence of optimality of states and actions, the derivative of the Lagrangian function with respect to both states and actions should be zero.

In equations (A.77),(A.78), it is proved that this equation is precisely the same as $\frac{\partial \mathcal{L}}{\partial x_k} = 0$.

3. Boundary Condition: Since the total cost is calculated up to the day T , the derivative of the total cost with respect to the state of the day $t + 1$ must be zero, i.e.,

$$\lambda_{T-1} = 0. \quad (\text{A.72})$$

4. Hamiltonian Condition: As discussed, in the presence of optimality of states and actions, the derivative of the Lagrangian function with respect to both states and actions is equal to zero. The Hamiltonian condition is as follows:

$$\frac{\partial \mathbb{H}_k}{\partial u_k} = 0, \quad \forall k \in [0, (T-1)]. \quad (\text{A.73})$$

Based on (5.4), the Lagrangian and Hamiltonian functions have similar terms involving the action of day k , and thus their derivatives with respect to the action on day k are equal. Consequently, we can state the following:

$$\frac{\partial \mathcal{L}}{\partial u_k} = \frac{\partial \mathbb{H}_k}{\partial u_k} = 0, \quad \forall k \in [0, (T-1)]. \quad (\text{A.74})$$

To determine the co-states, we refer to the adjoint equations, which are defined as the second condition in (2). The adjoint equations are as follows:

$$\lambda_{k-1} = \frac{\partial \mathbb{H}_k}{\partial x_k}, \quad \forall k \in [0, T-1]. \quad (\text{A.75})$$

Path trajectory optimality is guaranteed by these four conditions. Now, we demonstrate the validity of condition 2. It is known that the derivative of the Lagrangian function with respect to the state variable must equal zero. This is mathematically expressed as:

$$\frac{\partial \mathcal{L}}{\partial x_k} = 0. \quad (\text{A.76})$$

In (A.76), we decompose the term $\frac{\partial \mathcal{L}}{\partial x_k}$ by utilizing the relationship between the Lagrangian and Hamiltonian functions as detailed in (5.4):

$$\frac{\partial \mathcal{L}}{\partial x_k} = \frac{\partial \mathbb{H}_k}{\partial x_k} - \frac{\partial \lambda_{k-1}^\top x_k}{\partial x_k} = \frac{\partial \mathbb{H}_k}{\partial x_k} - \lambda_{k-1} = 0. \quad (\text{A.77})$$

This leads us to:

$$\lambda_{k-1} = \frac{\partial \mathbb{H}_k}{\partial x_k}. \quad (\text{A.78})$$

In (A.78), $\frac{\partial \mathcal{H}_k}{\partial x_k}$ is the change in the Hamiltonian function to the change in the state of the system at time k , and λ_{k-1} signifies the Lagrangian multiplier at the $(k-1)$ -th time instant. Equation (A.78) is identical to the equation stated in the adjoint condition (condition 2).

Appendix B

Matrix Representations

B.0.1 Section State-Action Weight Matrix

1. P , the rejection cost matrix is defined as follows:

$$P \triangleq \begin{bmatrix} \rho_1 & 0 & 0 & \cdots & 0 \\ 0 & \rho_2 & 0 & \cdots & 0 \\ 0 & 0 & \rho_3 & \cdots & 0 \\ & & \vdots & & \\ 0 & 0 & 0 & \cdots & \rho_I \end{bmatrix}. \quad (\text{B.1})$$

2. The visualized version of equation (A.26)

$$(PC_1^\top)V + \delta(C_1^\top C_1)^{-1}C_1^\top V = \phi + \delta C = \begin{bmatrix} \rho_1 \sum_{k=0}^N V_{KI+1,1} + \delta c_{11} & \rho_1 \sum_{k=0}^N V_{KI+1,2} + \delta c_{12} & \cdots & \rho_1 \sum_{k=0}^N V_{KI+1,N} + \delta c_{1N} \\ \rho_2 \sum_{k=0}^N V_{KI+2,1} + \delta c_{21} & \rho_2 \sum_{k=0}^N V_{KI+2,2} + \delta c_{22} & \cdots & \rho_2 \sum_{k=0}^N V_{KI+2,N} + \delta c_{2N} \\ \rho_3 \sum_{k=0}^N V_{KI+3,1} + \delta c_{31} & \rho_3 \sum_{k=0}^N V_{KI+3,2} + \delta c_{32} & \cdots & \rho_3 \sum_{k=0}^N V_{KI+3,N} + \delta c_{3N} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_I \sum_{k=0}^N V_{KI,1} + \delta c_{I1} & \rho_I \sum_{k=0}^N V_{KI,2} + \delta c_{I2} & \cdots & \rho_I \sum_{k=0}^N V_{KI,N} + \delta c_{IN} \end{bmatrix} = C_3. \quad (\text{B.2a})$$

3. The visualized version of (A.23):

$$\phi \triangleq (PC_1^\top) V = P \left[\underbrace{\mathbb{I}_I \quad \mathbb{I}_I \quad \cdots \quad \mathbb{I}_I}_{N \text{ times}} \right] V = \left[\underbrace{P \quad P \quad \cdots \quad P}_{N \text{ times}} \right] V \quad (\text{B.3a})$$

$$(\text{B.3b})$$

$$= \begin{bmatrix} \rho_1 \sum_{k=0}^N V_{KI+1,1} & \rho_1 \sum_{k=0}^N V_{KI+1,2} & \cdots & \rho_1 \sum_{k=0}^N V_{KI+1,N} \\ \rho_2 \sum_{k=0}^N V_{KI+2,1} & \rho_2 \sum_{k=0}^N V_{KI+2,2} & \cdots & \rho_2 \sum_{k=0}^N V_{KI+2,N} \\ \rho_3 \sum_{k=0}^N V_{KI+3,1} & \rho_3 \sum_{k=0}^N V_{KI+3,2} & \cdots & \rho_3 \sum_{k=0}^N V_{KI+3,N} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_I \sum_{k=0}^N V_{KI,1} & \rho_I \sum_{k=0}^N V_{KI,2} & \cdots & \rho_I \sum_{k=0}^N V_{KI,N} \end{bmatrix}. \quad (\text{B.3c})$$

4. The matrix version of the optimization problem discussed in (4.13) is as follows:

$$\min_{\rho_1, \rho_2, \rho_3, \dots, \rho_I, \delta} \|C_3 - \begin{bmatrix} \rho_1 \sum_{k=0}^N V_{KI+1,1} + \delta c_{11} & \rho_1 \sum_{k=0}^N V_{KI+1,2} + \delta c_{12} & \cdots & \rho_1 \sum_{k=0}^N V_{KI+1,N} + \delta c_{1N} \\ \rho_2 \sum_{k=0}^N V_{KI+2,1} + \delta c_{21} & \rho_2 \sum_{k=0}^N V_{KI+2,2} + \delta c_{22} & \cdots & \rho_2 \sum_{k=0}^N V_{KI+2,N} + \delta c_{2N} \\ \rho_3 \sum_{k=0}^N V_{KI+3,1} + \delta c_{31} & \rho_3 \sum_{k=0}^N V_{KI+3,2} + \delta c_{32} & \cdots & \rho_3 \sum_{k=0}^N V_{KI+3,N} + \delta c_{3N} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_I \sum_{k=0}^N V_{KI,1} + \delta c_{I1} & \rho_I \sum_{k=0}^N V_{KI,2} + \delta c_{I2} & \cdots & \rho_I \sum_{k=0}^N V_{KI,N} + \delta c_{IN} \end{bmatrix}\|_F. \quad (\text{B.4a})$$

B.0.2 Section Demand-Action Weight Matrix

1. Here is the visualized version of variable $\bar{\Theta}$ defined in (4.19)

$$\bar{\Theta} \triangleq P(\mathbb{I}_I - C_1^\top W_D) + \delta(C_1^\top C_1)^{-1}C_1^\top W_D = (C_1^\top C_1)^{-1}C_1^\top \gamma \quad (\text{B.5a})$$

$$= \begin{bmatrix} \rho_1 & 0 & 0 & 0 & 0 & \dots \\ 0 & \rho_2 & 0 & 0 & 0 & \dots \\ 0 & 0 & \rho_3 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & \rho_I \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \dots a_{1N} \\ a_{21} & a_{22} & a_{23} \dots a_{2N} \\ a_{31} & a_{32} & a_{33} \dots a_{3N} \\ \vdots & & \\ a_{I1} & a_{I2} & a_{I3} \dots a_{IN} \end{bmatrix} + \delta \begin{bmatrix} P_{11} & P_{12} & P_{13} \dots P_{1N} \\ P_{21} & P_{22} & P_{23} \dots P_{2N} \\ P_{31} & P_{32} & P_{33} \dots P_{3N} \\ \vdots & & \\ P_{I1} & P_{I2} & P_{I3} \dots P_{IN} \end{bmatrix} \quad (\text{B.5b})$$

$$= \begin{bmatrix} \rho_1 a_{11} + \delta P_{11} & \rho_1 a_{12} + \delta P_{12} & \dots & \rho_1 a_{1I} + \delta P_{1I} \\ \rho_2 a_{21} + \delta P_{21} & \rho_2 a_{22} + \delta P_{22} & \dots & \rho_2 a_{2I} + \delta P_{2I} \\ \rho_3 a_{31} + \delta P_{31} & \rho_3 a_{32} + \delta P_{32} & \dots & \rho_3 a_{3I} + \delta P_{3I} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_I a_{I1} + \delta P_{I1} & \rho_I a_{I2} + \delta P_{I2} & \dots & \rho_I a_{II} + \delta P_{II} \end{bmatrix} = (C_1^\top C_1)^{-1}C_1^\top \gamma \triangleq C_4. \quad (\text{B.5c})$$

2. In this section, the matrix representation of the optimization approach, as elaborated in (4.18), is presented.

$$\min_{\rho_1, \rho_2, \rho_3, \dots, \rho_I, \delta} \|C_4 - \begin{bmatrix} \rho_1 a_{11} + \delta P_{11} & \rho_1 a_{12} + \delta P_{12} & \dots & \rho_1 a_{1I} + \delta P_{1I} \\ \rho_2 a_{21} + \delta P_{21} & \rho_2 a_{22} + \delta P_{22} & \dots & \rho_2 a_{2I} + \delta P_{2I} \\ \rho_3 a_{31} + \delta P_{31} & \rho_3 a_{32} + \delta P_{32} & \dots & \rho_3 a_{3I} + \delta P_{3I} \\ \dots & \dots & \dots & \dots \\ \rho_I a_{I1} + \delta P_{I1} & \rho_I a_{I2} + \delta P_{I2} & \dots & \rho_I a_{II} + \delta P_{II} \end{bmatrix}\| \quad (\text{B.6a})$$

B.0.3 Hamiltonian Approach

1.

$$(I_N \otimes W)L = \begin{bmatrix} w_1 & 0 & 0 & \cdots & 0 \\ w_2 & 0 & 0 & \cdots & 0 \\ w_3 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_I & 0 & 0 & \cdots & 0 \\ 0 & w_1 & 0 & \cdots & 0 \\ 0 & w_2 & 0 & \cdots & 0 \\ 0 & w_3 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & w_I & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & w_1 \\ 0 & 0 & \cdots & 0 & w_2 \\ 0 & 0 & \cdots & 0 & w_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & w_I \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ \vdots \\ (N-1) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_I \\ 2w_1 \\ 2w_2 \\ 2w_3 \\ \vdots \\ 2w_I \\ \vdots \\ (N-1)w_1 \\ \vdots \\ (N-1)w_I \end{bmatrix}. \quad (\text{B.7})$$

2.

$$\left(\frac{1}{2}L^\top(\mathbb{I}_N \otimes W^\top)\right)^\top = \left(\frac{1}{2}(\mathbb{I}_N \otimes W^\top)^\top L\right) = \frac{1}{2}(I_N \otimes W)L \quad (\text{B.8})$$

3.

$$\begin{aligned}
(\mathbf{1}_N \otimes I_I)PD_k = & \begin{bmatrix} \rho_1 & 0 & 0 & 0 & 0 & \dots \\ 0 & \rho_2 & 0 & 0 & 0 & \dots \\ 0 & 0 & \rho_3 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & \rho_I \\ \rho_1 & 0 & 0 & 0 & 0 & \dots \\ 0 & \rho_2 & 0 & 0 & 0 & \dots \\ 0 & 0 & \rho_3 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & \rho_I \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \rho_1 & 0 & 0 & 0 & 0 & \dots \\ 0 & \rho_2 & 0 & 0 & 0 & \dots \\ 0 & 0 & \rho_3 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & \rho_I \end{bmatrix}_{NI \times I} & \begin{bmatrix} D_k^1 \\ D_k^2 \\ D_k^3 \\ \vdots \\ D_k^I \end{bmatrix}_{I \times 1} = \begin{bmatrix} \rho_1 D_k^1 \\ \rho_2 D_k^2 \\ \rho_3 D_k^3 \\ \vdots \\ \rho_I D_k^I \\ \rho_1 D_k^1 \\ \rho_2 D_k^2 \\ \rho_3 D_k^3 \\ \vdots \\ \rho_I D_k^I \end{bmatrix}_{NI \times 1}. \quad (\text{B.9})
\end{aligned}$$

4.

$$\begin{aligned}
& (\mathbf{1}_N \otimes I_I) P (\mathbf{1}_N^\top \otimes I_I) u_k \\
&= \begin{bmatrix} \rho_1 & 0 & \cdots & 0 & \rho_1 & 0 & \cdots & \rho_1 & 0 & \cdots & 0 \\ 0 & \rho_2 & \cdots & 0 & 0 & \rho_2 & \cdots & 0 & \rho_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \rho_I & 0 & 0 & \cdots & 0 & 0 & 0 & \rho_I \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \rho_1 & 0 & \cdots & 0 & \rho_1 & 0 & \cdots & \rho_1 & 0 & \cdots & 0 \\ 0 & \rho_2 & \cdots & 0 & 0 & \rho_2 & \cdots & 0 & \rho_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \rho_I & 0 & 0 & \cdots & 0 & 0 & 0 & \rho_I \end{bmatrix} \begin{bmatrix} u_k^{1,0} \\ u_k^{2,0} \\ u_k^{3,0} \\ \vdots \\ u_k^{I,0} \\ \vdots \\ u_k^{1,1} \\ u_k^{2,1} \\ u_k^{3,1} \\ \vdots \\ u_k^{I,1} \\ \vdots \\ u_k^{1,(N-1)} \\ u_k^{2,(N-1)} \\ u_k^{3,(N-1)} \\ \vdots \\ u_k^{I,(N-1)} \end{bmatrix}_{NI \times 1} \\
&= \begin{bmatrix} \rho_1 u_k^1 \\ \rho_2 u_k^2 \\ \rho_3 u_k^3 \\ \vdots \\ \rho_I u_k^I \\ \rho_1 u_k^1 \\ \rho_2 u_k^2 \\ \vdots \\ \rho_I u_k^I \end{bmatrix}_{NI \times 1}. \tag{B.10}
\end{aligned}$$

where, u_k^i is the total number of scheduled patients from class i on day k .

5. This item shows the visualized version of κ_k as defined in (A.61). We use the result

of previous equations in (B.7), (B.9), and (B.10).

$$\kappa_k \triangleq (I_N \otimes W)L - 2(\mathbf{1}_N \otimes I_I)PD_k + 2(\mathbf{1}_N \otimes I_I)P(\mathbf{1}_N^\top \otimes I_I)u_k + 2\delta u_k = \quad (\text{B.11a})$$

$$\begin{aligned}
(-2) \begin{bmatrix} \rho_1(D_k^1 - u_k^1) \\ \rho_2(D_k^2 - u_k^2) \\ \rho_3(D_k^3 - u_k^3) \\ \vdots \\ \rho_I(D_k^I - u_k^I) \\ \rho_1(D_k^1 - u_k^1) \\ \rho_2(D_k^2 - u_k^2) \\ \vdots \\ \vdots \\ \rho_I(D_k^I - u_k^I) \end{bmatrix}_{NI \times 1} &+ \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ w_1 \\ w_2 \\ \vdots \\ w_I \\ 2w_1 \\ 2w_2 \\ \vdots \\ 2w_I \\ \vdots \\ (N-1)w_1 \\ \vdots \\ (N-1)w_I \end{bmatrix} + 2\delta \begin{bmatrix} u_k^{1,0} \\ u_k^{2,0} \\ \vdots \\ u_k^{I,0} \\ \vdots \\ u_k^{1,1} \\ \vdots \\ u_k^{2,1} \\ \vdots \\ \vdots \\ u_k^{1,(N-1)} \\ \vdots \\ u_k^{2,(N-1)} \\ \vdots \\ \vdots \\ u_k^{I,(N-1)} \end{bmatrix}_{NI \times 1} \quad (\text{B.11b})
\end{aligned}$$

$$= - \begin{bmatrix} 2\rho_1(D_k^1 - u_k^1) - 2\delta u_k^{1,0} \\ 2\rho_2(D_k^2 - u_k^2) - 2\delta u_k^{2,0} \\ 2\rho_3(D_k^3 - u_k^3) - 2\delta u_k^{3,0} \\ \vdots \\ 2\rho_I(D_k^I - u_k^I) - 2\delta u_k^{I,0} \\ 2\rho_1(D_k^1 - u_k^1) - 2\delta u_k^{1,1} - W_1 \\ 2\rho_2(D_k^2 - u_k^2) - 2\delta u_k^{1,2} - W_2 \\ \vdots \\ \vdots \\ 2\rho_I(D_k^I - u_k^I) - 2\delta u_k^{I,(N-1)} - (N-1)W_I \end{bmatrix}_{NI \times 1} \quad (\text{B.12a})$$

$D_k^i - u_k^i$ represents the number of patients rejected from class i . Within this vector, each element is the result of multiplying the number of rejected patients by its corresponding rejection cost and then multiplying it by -2.

Appendix C

More Details on the Hamiltonian Approach

The Hamiltonian approach, as utilized in [28] and [27], incorporates the definition of co-states and Lagrangian multipliers. Based on concepts from [11] section 3.3, these studies relaxed system dynamics, which can be considered constraints. Our constraints, defined in (3.3), align with those in [28] and [27], and can be represented as follows:

$$x_{t+1} = Ax_t + Bu_t \quad \forall t \in [1, T]. \quad (\text{C.1})$$

[11], [28], and [27] assume system components are deterministic. The optimal action in these studies can be identified using the equation:

$$\bar{\mathcal{J}}(x_t) = \min_{u_t} g(x_t, u_t) + \bar{\mathcal{J}}(Ax_t + Bu_t), \quad (\text{C.2a})$$

$$\text{s.t. } x_{t+1} = Ax_t + Bu_t, \quad \forall t \in [0, T-1]. \quad (\text{C.2b})$$

In (C.2a), the lack of stochastic terms allows for deterministic forecasting of future events. However, the cost function we assume for the expert, as outlined in (3.14), incorporates stochastic elements:

$$\begin{aligned} \bar{\mathcal{J}}(x_t, D_t) &= \min_{u_t} g(x_t, D_t, u_t) + \mathbb{E} [\bar{\mathcal{J}}(Ax_t + Bu_t, D_t)|x_t] + \delta u_t^\top u_t, \\ \text{s.t. } x_{t+1} &= Ax_t + Bu_t, \quad \forall t \in [0, T-1]. \end{aligned} \quad (\text{C.3})$$

In (C.2a), there is an absence of stochastic terms, which allows for deterministic future predictions. However, the cost function we assume for the expert, as detailed in (3.14), includes stochastic elements:

$$\begin{aligned} \bar{\mathcal{J}}(x_t, D_t) &= \min_{u_t} g(x_t, D_t, u_t) + \mathbb{E} [\bar{\mathcal{J}}(Ax_t + Bu_t, D_t)|x_t] + \delta u_t^\top u_t, \\ \text{s.t. } x_{t+1} &= Ax_t + Bu_t, \quad \forall t \in [0, T-1]. \end{aligned} \quad (\text{C.4})$$

This cost function contrasts with the quadratic, deterministic cost functions used in [28] and [27] (C.2a), primarily because it accounts for stochastic demand, a factor not considered in their studies. Moreover, our cost function contains a smoothness coefficient term (δ), which is added to maintain non-negativity of actions, a concept not included in [28] and [27].

Our methodology, inspired by section 3.3 of [7], defines the optimality conditions for actions and states in a similar fashion. Further details on the optimality of path trajectories are discussed in A.10. Although [7] does not consider stochasticity or a smoothness

coefficient term, we adapt their methods to our unique cost function in (4.1) to obtain our desired results.

Appendix D

More Details on the Riccati Approach

The Riccati approach, as extensively discussed in [7], is also employed in [28] and [27] for determining cost parameters in systems with linear dynamics and quadratic cost functions. Our study shares these assumptions, including a quadratic state cost matrix (Q) and quadratic action cost matrix (R). However, unlike [28] and [27], our study includes a stochastic demand in the cost function and cross-term multipliers.

In [7], the dynamics include noise, and they demonstrate that the optimal action is linearly dependent on the state:

$$u_t^* = Lx_t, \quad \forall t \in [1 : T]. \quad (\text{D.1})$$

In our study, Lemma 1 establishes the optimal action as:

$$u_t = Vx_t + W_D D_t + W_E, \quad \forall t \in [1 : T]. \quad (\text{D.2})$$

If noise is considered in the system dynamics, we redefine the state as matrix Y in Proposition 4.1.1, with each row representing the system state at a particular time and matrix W as the action matrix. This allows our approach to align with the Riccati approach in [7]. The optimal expert actions are as follows:

$$u_t = WY_t, \quad \forall t \in [1 : T]. \quad (\text{D.3})$$

In (D.3), the expert's actions are reformulated as a linear function of the system state. The system dynamics in (3.3) are modified to:

$$Y_t = AY_t + Bu_t + w_t, \quad \forall t \in [1 : T]. \quad (\text{D.4})$$

In (D.4), daily demand is added as noise to the system. [7] assumes this noise has a mean of zero, but in our case, we consider the expected demand as the mean for the noise components related to daily demand.

By adapting this notation, our work connects to the Riccati approach in [7]. They discuss the linear relationship between state and action and the conditions ensuring a solution to the Riccati equations. Our focus, however, is on solving the inverse problem rather than verifying the existence of action matrices. We are primarily concerned with identifying the expert's cost parameters. The conditions regarding the positive semi-definiteness of the quadratic state cost matrix (Q) and the positive definiteness of the quadratic action cost matrix (R) in [7] are insufficient for our purposes. Instead, we utilize the unique characteristics of our proposed cost function. For instance, we express matrix Q as $Q = Se_1e_1^\top$, a specification not assumed in [7] as they address the forward problem in its general form.

Our focus is specifically on advance scheduling, leveraging the distinct aspects of our cost function.