IMPROVING HEALTHCARE DELIVERY: LIVER HEALTH UPDATING AND SURGICAL PATIENT ROUTING

by

Sepehr Nemati Proon

B.S., Sharif University of Technology, 2005

M.S., Sharif University of Technology, 2007

Submitted to the Graduate Faculty of

the Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2014

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This dissertation was presented

by

Sepehr Nemati Proon

It was defended on

October 18 2013

and approved by

Andrew J. Schaefer, William Kepler Whiteford Professor, Department of Industrial Engineering

Lisa M. Maillart, Ph.D., Associate Professor, Department of Industrial Engineering

Oleg A. Prokopyev, Ph.D., Associate Professor, Department of Industrial Engineering

Jayant Rajgopal, Professor, Department of Industrial Engineering

Prakash Mirchandani, Professor, Katz Graduate School of Business

Oleg V. Shylo, Assistant Professor, Department of Industrial and Systems Engineering, University of Tennessee

Dissertation Director: Andrew J. Schaefer, William Kepler Whiteford Professor, Department of Industrial Engineering

IMPROVING HEALTHCARE DELIVERY: LIVER HEALTH UPDATING AND SURGICAL PATIENT ROUTING

Sepehr Nemati Proon, PhD

University of Pittsburgh, 2014

Growing healthcare expenditures in the United States require improved healthcare delivery practices. Organ allocation has been one of the most controversial subjects in healthcare due to the scarcity of donated human organs and various ethical concerns. The design of efficient surgical suites management systems and rural healthcare delivery are long-standing efforts to improve the quality of care. In this dissertation, we consider practical models in both domains with the goal of improving the quality of their services.

In the United States, the liver allocation system prioritizes among patients on the waiting list based on the patients' geographical locations and their medical urgency. The prioritization policy within a given geographic area is based on the most recently reported health status of the patients, although blood type compatibility and waiting time on the list are used to break ties. Accordingly, the system imposes a health-status updating scheme, which requires patients to update their health status within a timeframe that depends on their last reported health. However, the patients' ability to update their health status at any time point within this timeframe induces information asymmetry in the system. We study the problem of mitigating this information asymmetry in the liver allocation system. Specifically, we consider a joint patient and societal perspective to determine a set of Pareto-optimal updating schemes that minimize information asymmetry and data-processing burden. This approach combines three methodologies: multi-objective optimization, stochastic program-

ming and Markov decision processes (MDPs). We exploit the structural properties of our proposed modeling approach and develop a decomposition algorithm to identify the exact efficient frontier of the Pareto-optimal solutions within any given degree of accuracy.

Many medical centers offer transportation to eligible patients. However, patients' transportation considerations are often ignored in the scheduling of medical appointments. In this dissertation, we propose an integrated approach that simultaneously considers routing and scheduling decisions of a set of elective outpatient surgery requests in the available operating rooms (ORs) of a hospital. The objective is to minimize the total service cost that incorporates transportation and hospital waiting times for all patients. Focusing on the need of specialty or low-volume hospitals, we propose a computationally tractable model formulated as a set-partitioning based problem. We present a branch-and-price algorithm to solve this problem, and discuss several algorithmic strategies to enhance the efficiency of the solution method. An extensive computational test using clinical data demonstrates the efficiency of our proposed solution method. This also shows the value of integrating routing and scheduling decisions, indicating that the healthcare providers can substantially improve the quality of their services under this unified framework.

Keywords: Organ allocation,operating rooms scheduling problem, Markov decision processes, branch- and-price, multi-objective decision making, two-stage stochastic programming.

TABLE OF CONTENTS

1.0 INTRODUCTION 1

1.1 Liver Allocation In the United States 2

1.2 Surgical Suite Scheduling Problem 5

1.3 Transportation Barriers to Access Healthcare 6

1.4 Problem Statements and Dissertation Outline 7

2.0 LITERATURE REVIEW 11

2.1 Multi-Objective Programming (MOP) 11

2.1.1 Solution Approaches for MOPs 12

2.2 Two-Stage Stochastic Programs 13

2.3 Markov Decision Processes (MDPs) 16

2.4 SPAR: Stochastic Programming with Adversarial Recourse 18

2.5 Branch and Price 20

3.0 BALANCING EQUITY AND EFFICIENCY IN LIVER ALLOCATION VIA REVISED HEALTH REPORTING FREQUENCIES . . . 21

3.1 Introduction 21

3.2 Literature Review and Our Contributions 23

3.3 Designing Updating Schemes 24

3.4 Patient Decision-Making Problem 27

3.4.1 Mathematical Formulation 27

3.4.2 Modeling the Inequity and Efficiency of Updating Schemes 31

3.4.3 Structural Properties of the Optimal Patient Policy 33

3.5 Mathematical Formulation of USDP[μ] 38

3.5.1 A Decomposition-based Algorithm for USDP[μ] 38
3.6 Approximating the Efficient Frontier of Updating Schemes 46
3.7 Numerical Study 49
3.7.1 Parameter Estimation and Implementation Details 49
3.7.2 Evaluating Patients' Behavior 51
3.7.3 Mitigating Information Asymmetry in Liver Allocation 54
3.8 Conclusion 58
4.0 THE SURGICAL PATIENT ROUTING PROBLEM 60
4.1 Introduction 60
4.2 Prior Work and Our Contributions 61
4.3 Mathematical Formulation 62
4.3.1 General Surgical Patient Routing Problem 63
4.3.2 Batch Surgical Patient Routing Problem (BSPRP) 67
4.4 A Branch-and-Price Approach 72
4.4.1 Route Generation 73
4.4.2 Algorithmic Enhancements 75
4.4.2.1 Patient-Pair Branching 76
4.4.2.2 Improving Column Generation. 77
4.5 Computational Study 80
4.5.1 Test Instances 80
4.5.2 Solution Method Performance 81
4.5.3 Value of Integrating Surgery Scheduling and Vehicle Routing Decisions 85
4.6 Conclusion 86
5.0 CONCLUSIONS AND FUTURE RESEARCH 87
5.1 Balancing Equity and Efficiency in Liver Allocation via Revised Health Reporting frequencies 87
5.2 The Surgical Patient Routing Problem 89
BIBLIOGRAPHY 90

LIST OF TABLES

1 The current MELD score updating scheme of UNOS [52] 4

2 List of updating schemes on the approximated efficient frontier proposed in [62] 55

3 Dominant updating schemes compared to [62] 56

4 Test instances for computational study 81

5 Performance of the developed branch-and-price approach compared to CPLEX. 82

6 Description of column generation strategy. 82

7 Performance of the branch-and-price method under different algorithmic strategies. The reported solution times are in seconds. 83

8 Comparing objective function values of BSPRP when using exact, TSP- and TRP-based methods. 84

LIST OF FIGURES

1 UNOS waiting list and organ donation trends in recent years [52] 2

2 Registration and transplant trends on liver waiting list in recent years [52] . . 3

3 Updating scheme design problem 8

4 Hypothetical Pareto-optimal updating schemes 25

5 USDP[μ] model representation 26

6 Optimal transplant behavior of patient under UNOS updating for $\tau = 0$ compared to a weekly updating scheme 51

7 Patient benefits under UNOS updating with same updating and do nothing rewards 53

8 Patient benefits under UNOS updating when updating is penalized 55

9 Efficient frontier of updating schemes compared to [62] 56

10 Efficient frontier of updating schemes compliant with the UNOS requirements 57

11 Efficient frontier of monotonic updating schemes 57

12 Planning horizon consisting of 2 days and 4 stages 63

1.0 INTRODUCTION

Healthcare is one of the world's largest and fastest growing industries [112]. Healthcare in the U.S. accounted for 17.6% of Gross Domestic Product (GDP) in 2010, and is expected to grow to 19.8% by 2020 [51]. Both on a per-capita basis and as a fraction of the GDP, the U.S. spends more on healthcare than any other member state of the World Health Organization [51, 116]. This burden highlights the need for cost-saving measures to be taken by the private and public sectors.

These aforementioned concerns about the performance of the U.S. healthcare system have motivated significant interest in medical decision making over the past two decades. Operations Research (OR) techniques have been utilized to address questions on how to design, schedule and manage healthcare systems, leading to a variety of healthcare applications. OR applications in healthcare constitute a vast body of literature ranging from policy and system design studies, like ambulance location [21, 22], emergency room or operating room scheduling [64, 83, 115], organ allocation policy design [33, 69], immunization and vaccine selection [88], to the treatment of individual patients, such as cancer treatment [74, 75, 79, 92, 118], and the optimal timing of organ transplants [4, 5, 6, 59, 104, 105]

In this dissertation, we focus on the design of healthcare systems with the ultimate goal of improving the quality of their services. More specifically, we focus our attention on the design of organ allocation systems and surgical suite management policies. First, we study the problem of allocating organs for liver transplantation, the second most transplanted organ. Next, we consider patients' mobility issues to access medical care in the U.S., specifically when the medical centers provide both surgical and transportation services to patients.

1.1 LIVER ALLOCATION IN THE UNITED STATES

Organ transplantation dates back to the first kidney transplant in 1954 [52]. Donated human organs are highly perishable and scarce resources that should be allocated efficiently and equitably in order to maximize the possible transplant outcomes and minimize waste. As of October 2013, 118,754 patients were waiting for organs (primarily kidney, liver, heart, lung, pancreas and intestine) in the U.S. and on average a new name is added to the national patient waiting list every 13 minutes [52]. Although the number of donated organs has steadily increased during the past two decades, as shown in Figure 1, the supply of organs have not kept pace.

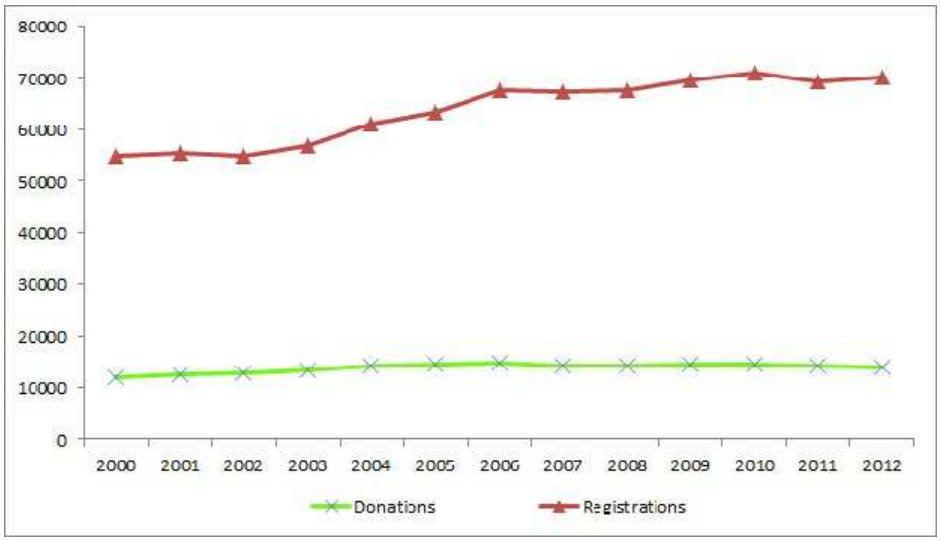

Figure 1: UNOS waiting list and organ donation trends in recent years [52]

End-stage liver disease (ESLD) is any acute or chronic condition that leads to irreversible liver dysfunction. ESLD, the 12th leading cause of death in the U.S. [50], includes diseases such as chronic liver diseases, primary biliary cirrhosis and hepatitis among many others [29]. Unlike other organs such as kidney, for which dialysis is an alternate therapy, transplantation is the only viable therapy for ESLD patients. There are two sources of livers for transplantation: living donors and deceased donors. Living donor transplantation has emerged in recent decades and involves removing a segment of a liver from a healthy living

donor and implanting it into a recipient. Under normal circumstances, the livers of both donor and recipient grow to normal size in a few weeks [4]. Although living donor transplantation has alleviated the shortage of livers, living donations still represent a small percentage of donated livers.

Figure 2 shows the gap between the number of liver transplants and the registrations to the national liver waiting list between the years 2000 and 2012 in the U.S. The scarcity of donated livers together with the absence of alternate therapies for ESLD patients and the need for equity in liver allocation suggest a need for an efficient and fair liver allocation system.

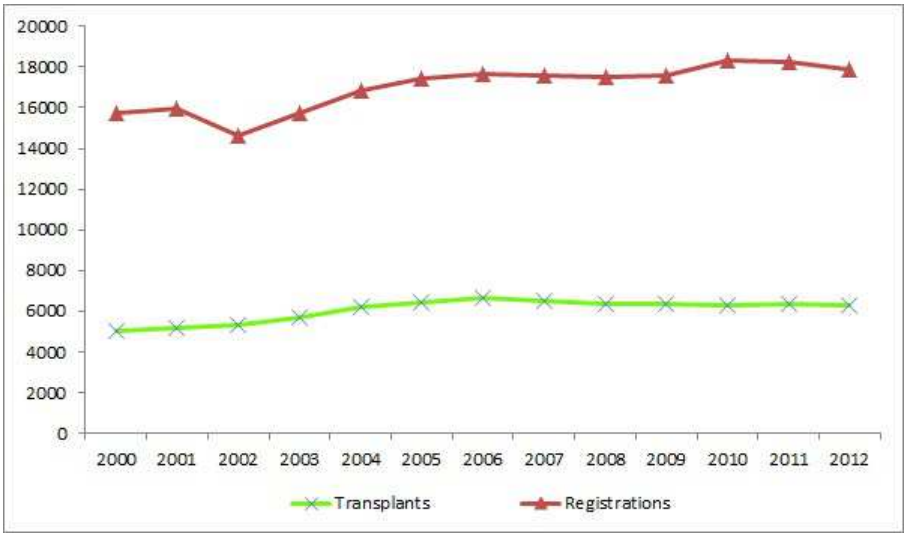

Figure 2: Registration and transplant trends on liver waiting list in recent years [52]

The liver allocation system in the U.S. is administered by the United Network for Organ Sharing (UNOS). UNOS divides the U.S. into 11 geographic regions, where each region is further divided into sub-regions called donation service areas (DSAs) of Organ Procurement Organizations (OPOs). Currently, there are 58 OPOs serving unique areas of varying sizes, population densities, donation rates and transplantation activities [86]. These OPOs, each composed of multiple transplant centers, are responsible for the identification of organ donors, organ retrieval, preservation, transportation, and transplantation [52]. The transplant cen-

ters within the OPOs maintain a list of eligible candidates for donation and transplantation operations. As of October 2014, there were 127 liver transplant centers, and the number of these centers is increasing [86].

Upon the arrival of a donated liver, UNOS ranks the patients on the national waiting list. This ranking determines the priority among potential recipients for allocating the donated liver and is primarily based on two main factors; a patient's location, and patient's medical urgency, where compatibility with the donor and waiting time are used to break ties. Next, UNOS contacts the transplant centers responsible for the patients with the highest priority. The transplant team responsible for the patient is given one hour to decide whether to accept the liver offer or not, where there is no penalty if the offer is declined.

At the level of medical urgency, the current liver allocation system considers two main categories for adult patients, Status 1A and Meld for End-stage Liver Disease (MELD) patients. Status 1A patients who have fulminant liver failure with a life expectancy without liver transplant of less than 7 days receive the highest in allocating livers and constitute only 0.1% of the total number of patients on the waiting list [93], hence, we ignore such patients in our models. The severity of ESLD for patients who are not eligible to be listed as Status 1A patients is assessed using their MELD scores. The MELD is a scoring system to predict a patient's probability of pre-transplant death, and incorporates specific clinical lab values of the patient. These scores assume integer values between 6 (healthiest patient) to 40 (gravely ill patient), [52, 80, 117].

UNOS requires patients to update their clinical values within a timeframe that depends on their last reported health. Table 1 shows the current updating scheme for the MELD patients [52]. Patients can update more frequently than what the updating scheme dictates. However, if a patient fails to update by the required time, UNOS temporarily downgrades her MELD score to 6, the healthiest score, until new results are received [52]. Transplant

Table 1: The current MELD score updating scheme of UNOS [52]

Last reported MELD score	≤ 10	11-18	19-24	≥ 25
Minimum update frequency (days)	365	90	30	7

centers caring for candidates on the waiting list are responsible for updating patients' MELD scores. Roberts et al. [100] estimates the data processing cost at a large transplant center to be over $175,000 per year that excludes the costs incurred by UNOS. Moreover, reporting health status requires clinical tests that are not only costly but often result in patients' inconvenience.

1.2 SURGICAL SUITE SCHEDULING PROBLEM

Surgery generates more than 40% of the total expenses and revenues of a hospital [51], making it an important opportunity for cost reduction. The surgery delivery process consists of a variety of activities and is known as *peri-operative services*. There are two types of surgery scheduling processes [56]. First, under a so-called *advance surgery* paradigm, patients are scheduled on a future date, depending on the availability of surgical resources and surgeon's schedule. Second, under a so-called *allocation scheduling* paradigm, surgeries are scheduled in available operating rooms (ORs) on the day of surgery, usually on a first-come first-serve (FCFS) basis. This scheduling process is mainly based on the assumption that all the patients are in the hospital and ready for surgery [39, 56, 57].

Typically surgeries are divided into those performed on an inpatient and an outpatient basis. In the inpatient case, patients are admitted to the hospital on or prior to the day of surgery and stay in the hospital until the completion of their recovery period. Conversely, in an outpatient setting, patients arrive on the day of surgery and leave after the completion of the surgical and post-operative procedures.

Hospitals typically choose to schedule surgeries in the ORs under an open-booking, a block-booking framework, or a combination of these two approaches. Under an open-booking framework, the surgical resources are shared among specialty teams, individual surgeons or surgical departments. In a block-booking system a block of OR time is assigned to individual surgeons or surgical departments where the duration of each block is determined by the medical staff in the hospital based on the past surgical data. A block-booking framework have several advantages as well as disadvantages, compared to an open-booking scheme [40]. The main advantage of a block-booking framework is that it usually simplifies the surgery

scheduling process and is more convenient for both surgeons and the hospital staff, making it more common in practice relative to open-booking approach. However, it is likely that the schedules under this framework tend to be unbalanced, potentially leading to inefficient utilization of ORs [40, 41, 47].

Scheduling surgeries in ORs is a complex process requiring the cooperation of the medical staff in the hospitals, patients and the surgical teams. There are a variety of cost measures in surgical suite scheduling literature to evaluate the different surgical suites management policies. Typically, the key objective in these practices is to balance the cost-effectiveness and efficiency in OR utilization. However, in health service environments such as government hospitals, the main goal is to improve patients' satisfaction [24].

1.3 TRANSPORTATION BARRIERS TO ACCESS HEALTHCARE

Access to healthcare is a major public health policy issue in the U.S., where the Institute of Medicine defines access as the timely use of the personal health services to achieve the best possible outcome [81]. Greater access to healthcare leads to better health, higher utilization of health resources, and improves the quality of care [81, 87]. There are a variety of barriers that limit a timely and efficient healthcare access such as health insurance coverage, health provider policies and transportation issues.

Transportation has been identified as a common barrier to healthcare in the U.S., posing many challenges to health policy makers. This issue has become more crucial in the past decade due to the rapidly increasing elderly population and those with special medical needs [7, 87]. In 2002, 9% of the elderly population (those 65 and older) did not obtain needed medical care due to transportation problems [87], and this number is expected to increase. Transportation was the third common barrier to a regular access to heathcare [97].

Recognizing the growing need to improve access to medical care, a number of organizations in the United States offer transportation services for patients. One example of such efforts is the Beneficiary Travel plans by the Veterans Health Administration (VHA) [34], which provides medical care to approximately 23 million veterans, 40 percent of whom are elderly. Another example is the transportation support offered by the Disabled American

Veterans organization (DAV). The DAV serves veterans using a pool of volunteer drivers, working with Veterans Administration Medical Centers (VAMCs) and other community organizations.

Although the volume of these services is increasing, there are numerous issues that remain unaddressed. Burkhardt et al. [24] report a variety of strategies to improve the quality of supporting transportation services for veterans. However, these strategies are not limited to veterans, and they can also help patients in general. Coordinating transportation with medical schedulers and hospital services is among the most major concerns in current practices. The reason is due to the fact that medical appointment schedulers do not necessarily perceive patients' transportation problems when they set up the appointments. This issue is even more crucial when: (i) the transportation resources are limited, or (ii) patients are in a need of special medical care; or (iii) the requested medical services require high-cost resources at the hospital such as surgical suites. Consequently, most medical centers schedule patients' medical appointments and their transportation plans separately and sequentially [23, 24].

1.4 PROBLEM STATEMENTS AND DISSERTATION OUTLINE

The contributions of this dissertation are two-fold. First, we propose an approach to mitigate the information asymmetry in the current liver allocation system due to patients' possible gaming ability in the sense that patients may exploit flexibility in the health status reporting requirements. Second, we study an operational problem to improve the efficiency of surgical suites management policies in hospitals by simultaneously considering patients' transportation considerations and surgical scheduling decisions.

As noted in Section 1.1, UNOS imposes a health status updating scheme that requires patients to report their health status within a given timeframe that depends on their last reported health. However, patients' ability to update their health status at any time within this timeframe may allow them to game the system by concealing changes in their health status. This gaming ability may induce information asymmetry between the UNOS and the patients in the liver allocation mechanism, leading to misallocation of livers. This information

asymmetry can be alleviated through more frequent updating requirements, but with a price of an increase in the already existing significant data processing burden. Consequently, an ideal updating requirement balances the two possibly conflicting objectives of: (i) minimizing the information asymmetry, and (ii) minimizing the data-processing burden associated with the significant data collection costs and patient inconvenience due to frequent clinical tests.

We propose a method to determine a set of Pareto-optimal updating requirements that minimize the information asymmetry with respect to the tolerance levels of data-processing burden. As shown in Figure 3, our modeling approach is consists of three related models: (i) a multi-objective model that simultaneously minimizes the system inequity due to the resulting information asymmetry, and maximizes the system efficiency, i.e., reducing the data processing burden, (ii) an *updating scheme design problem* that minimizes the information asymmetry while preserving a given desired tolerance level for the data-processing burden; and (iii) a *patient decision model* that captures the adversarial effect of a set of autonomous self-interested patients' decisions.

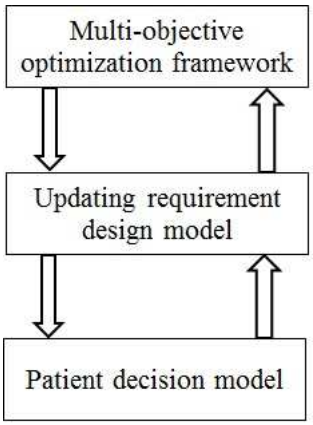

Figure 3: Updating scheme design problem

We measure the system inequity as the expected increase in the total discounted lifetime of the patients due to gaming. We define the system efficiency as the expected decrease in the data processing burden as a result of the given flexibility in reporting health. We model the problem of determining an updating scheme with minimum system inequity for a given

desired level of reduction in data processing burden using two-stage stochastic programming problem with adversarial recourse (SPAR) [8]. The first-stage decisions in this model involve selecting an updating scheme. The second-stage recourse problem refers to the decision-making process of a set of autonomous and self-interested patient, modeled as discrete-time, infinite horizon Markov Decision Processes (MDPs). Each MDP model involves determining the optimal patient's decisions regarding updating his/her health status, do nothing without transplantation, or accept a liver offer (if any) at each decision point in order to maximize his/her expected life time. We exploit several structural properties of the optimal patients' decisions within a decomposition algorithm to efficiently solve the stochastic programming problem. We further extend our proposed algorithm to determine the exact frontier of the Pareto-optimal updating schemes. Our extensive numerical results show that the current updating requirements of UNOS can be revised to mitigate the resulting system inequity without additional increase in the data processing burden.

As discussed in Section 1.3, many patients face difficulties when accessing medical facilities in the U.S., particularly in rural areas. To alleviate these difficulties, medical centers offer transportation to patients, including shuttle services, that provide shared rides for patients. Unfortunately, the patient transportation plan of the patients is not usually coordinated with the scheduling of medical appointments. In Chapter 4, we focus on a problem to improve the quality of such services with an emphasis on the patients' perspective. To do so, we propose an integrated approach that simultaneously considers routing and scheduling decisions of a set of patients with outpatient surgery requests where the medical centers provide transportation. This transportation is typically shared-type round trip shuttle services. The overall objective of the model is to minimize the total service cost that incorporates transportation and hospital waiting times for all the patients. We formulate this as a mixed-integer program that is impossible to solve using commercial solvers. We use the structural properties of the proposed model and develop a branch-and-price algorithm. We further investigate several algorithmic strategies to improve solution efficiency of the proposed model. The performance of the proposed approach is evaluated through extensive computational study using clinical data. Our results demonstrate that healthcare providers can substantially improve the quality of their services by integrating scheduling and routing decisions.

The remainder of this dissertation is organized as follows. In Chapter 2, we present a review of the modeling and solution methodologies utilized in this dissertation. In Chapter 3, we describe the patient decision model, structural properties of patients optimal decisions, and a model to determine an updating scheme with minimum system inequity for a given desired data processing burden. We then propose our solution method to solve this problem and describe a general algorithm to approximate the exact efficient frontier of the updating schemes. In Chapter 4, we present our proposed model to integrated surgery and transportation decisions and the branch-and-price algorithm to solve it. We discuss conclusions and highlight future research directions in Chapter 5.

2.0 LITERATURE REVIEW

In this chapter, we briefly discuss the methodologies used in this dissertation. In Section 2.1, we review multi-objective programming techniques. Section 2.2 provides an introduction to the stochastic programming with an emphasis on two-stage stochastic programming problems and their solution techniques. In Section 2.3, we discuss Markov decision processes and briefly review a modeling framework that combines stochastic programming and Markov decision processes in Section 2.4. We conclude this chapter with a review of branch and price and column generation in Section 2.5.

2.1 MULTI-OBJECTIVE PROGRAMMING (MOP)

Multi-objective programming (MOP) is the process of systematically and simultaneously optimizing a collection of possibly conflicting objective functions over a set of decisions [44]. In this section, we first introduce the general multi-optimization programming framework. We also briefly discuss the most common solution methods for this problem. We follow the notation in [44] and [45] throughout the rest of this section. The general MOP can be written as follows:

$$\begin{aligned} \min \quad & f(y) \\ \text{s.t.} \quad & y \in Y, \end{aligned} \tag{2.1}$$

where $f(f_1, ..., f_k)^T : Y \to R^k$ is a vector of possibly conflicting objective functions, and $Y \subseteq R^n$ is a nonempty set referred to as the feasible region. The feasible region Y can be represented by a number of inequality constraints. Because the objective functions in MOP

can be conflicting, solving the problem and comparing these objective functions require an ordering of these functions. This measure of efficiency is referred to as the Pareto optimality [89] in the MOP literature and is defined as follows:

Definition (Pareto Optimality) A solution $y_e \in Y$ with the objective function $f(y_e)$ is called (globally) *Pareto optimal* or *efficient*, if and only if there exists no other solution $y \in Y$ such that $f_i(y) \leq f_i(y_e)$ for all $i = 1, ..., k$ and $f_j(y) < f_j(y_e)$ for at least $j \in \{1, ..., k\}$.

Hence, the objective of MOP is to find the set of all efficient solutions y_e, denoted by $Y_e \subseteq Y$. Next, we briefly review the most common solution approaches for the MOPs.

2.1.1 Solution Approaches for MOPs

Solution methods for MOPs often embed *scalarization*, which involves combining the multiple objective functions of an MOP into a single-objective scalar function [45]. This approach in general is known as the *scalarization* or *weighted-sum* method. More specifically, the weighted-sum method minimizes a positively weighted sum of the objectives in MOP, as:

$$\min \quad \sum_{i=1}^{k} \lambda_i f_i(y) \tag{2.2}$$

$$\text{s.t.} \quad y \in Y, \tag{2.3}$$

with $\lambda_i \geq 0$ as the positive weight of the ith objective function, for $i = 1, ..., k$.

The main advantage of this method is that for each λ, the computational effort needed to solve the weighted sum problem is the same as the single objective counterpart of an MOP [45]. Another advantage of this approach is that if all of the weights are positive, the optimal solution of 2.3 is Pareto optimal for convex and continuous efficient frontier [44]. The main disadvantage of the weighted sum approach is that it fails to obtain efficient solutions on the non-convex portions of the Pareto optimal set. Hence, this method is not appropriate for problems with discrete or non-convex Pareto optimal frontier [44]. Several other variants of this approach have been proposed to alleviate the deficiencies of the weighted sum method, including the weighted t^{th} power approach, and the weighted quadratic method [45].

Another popular scalarization approach that overcomes some of the convexity problems of the weighted sum technique is the ϵ-constraint method. This involves minimizing a primary objective, $f_p(y)$ for some $p \in \{1, ..., k\}$, and introducing all the other objectives in the form of new constraints as:

$$\min \quad f_p(y) \tag{2.4}$$

$$\text{s.t.} \quad f_i(y) \leq \epsilon_i, \qquad \forall i \in \{1, ..., k\} \setminus \{p\}, \tag{2.5}$$

$$y \in Y. \tag{2.6}$$

This approach identifies a number of Pareto optimal solutions on a non-convex domain when the weighted sum technique fails to identify them. However, the problem with this method is the suitable selection ϵ_i to ensure a feasible solution. We refer the reader to [26] for more details of this method.

Among all the other methods to solve MOPs, *goal programming* (GP) is perhaps the best known method. GP was introduced by Charnes et al. [34] and involves expressing a set of desirable levels $\{f_1^*(y), ..., f_k^*(y)\}$ associated with the set of objective functions $\{f_1(y), ..., f_k(y)\}$. This problem formulation allows the objectives to be under- or overachieved. The relative degree of under- or overachievement of the desired levels is controlled by a vector of weighting coefficients, and the problem is expressed as a standard optimization problem [27].

Finally, the solution methods to approximate the exact efficient frontier of the Pareto optimal solution, often involve iterative methods. These iterative methods embed a scalarization technique of choice to generate Pareto optimal points during the solution procedure. For a recent survey of these techniques, the reader is referred to [44, 45].

2.2 TWO-STAGE STOCHASTIC PROGRAMS

Stochastic programming as a branch of mathematical programming deals with the optimization problems where the model parameter are subject to uncertainty. In this section, we

focus our attention on the two-stage stochastic programming model, the most widely studied stochastic programming models. We refer the reader to Birge and Louveaux [18] and Kall and Wallace [63] for fundamentals of stochastic programming.

A two-stage stochastic program is a mathematical program involving uncertain model parameters and is composed of first and second decision stages. Under this framework, the first-stage problem is solved in the face of uncertainty in the model parameters. Once the first-stage problem is solved, the uncertainty in model parameters is realized and the decision maker solves a set of second-stage problems, known as *recourse* problems. In this context, the recourse model depends on the first-stage solution and the outcome of a random event. The objective of the two-stage stochastic programming model is to minimize the first-stage cost plus the expected recourse cost.

Let A be a real-valued matrix of size $m_1 \times n_1$, with n_i and m_i as the number of decision variables and constraints in stage i for $i = 1, 2$, respectively. Let b be a vector in $\mathbb{R}^{m_1}$ and ξ be a discrete random variable describing the uncertain model parameter with a finite support denoted by Ξ. Each $\xi^k \in \Xi$ for $k = 1, ..., K = |\Xi|$ denotes the kth element in Ξ, and is referred to as a *scenario*. Additionally, let p^k be the probability that ξ^k is realized. Using this notation, Beale [12] and Dantzig [31] formulated the two-stage stochastic linear programming as:

$$\min \quad c^T x + \mathbb{E}_\xi[d(\xi)^T y(\xi)] \tag{2.7}$$

$$\text{s.t.} \quad Ax \geq b, \tag{2.8}$$

$$T(\xi^k)x + W(\xi^k)y(\xi^k) \geq h(\xi^k), \qquad k = 1, ..., K, \tag{2.9}$$

$$x \geq 0, \tag{2.10}$$

$$y(\xi^k) \geq 0, \qquad k = 1, ..., K. \tag{2.11}$$

In the above formulation, c is a known vector in $\mathbb{R}^{n_1}$ and denotes first-stage decisions cost, and for each scenario $d(\xi^k) \in \mathcal{R}^{n_2}$, $h(\xi^k) \in \mathcal{R}^{n_2}$. For each scenario, $T(\xi^k)$ and $W(\xi^k)$ referred to as the technology matrix and the recourse matrix, are of size $m_2 \times n_1$ and $m_2 \times n_2$, respectively. These matrices define the set of feasible second-stage solutions in (2.9) and (2.11) for each scenario. Alternatively, a scenario vector can be represented as

$\xi^k = (d(\xi^k)^T, h(\xi^k)^T, T(\xi^k), W(\xi^k))$ for $k = 1, ..., K$. Thus, for a first-stage decision vector x, the recourse problem decomposes into K independent subproblems, defined for each scenario. The above-mentioned model can be formulated as the so-called *deterministic equivalent program* as follows:

$$\begin{aligned} \min \quad & c^T x + \mathcal{Q}(x) \\ \text{s.t.} \quad & Ax \geq b, \\ & x \geq 0, \end{aligned} \tag{2.12}$$

with $\mathcal{Q}(x) = \mathbb{E}_\xi[Q(x, \xi^k)]$ known as the *expected recourse function*, where for scenario ξ^k,

$$\begin{aligned} Q(x, \xi^k) = \min \quad & d(\xi^k)^T y \\ \text{s.t.} \quad & W(\xi^k) \geq h(\xi^k) - T(\xi^k)x, \\ & y \geq 0. \end{aligned} \tag{2.13}$$

The most common approach to solve two-stage stochastic programming problems is to use a decomposition algorithm. A decomposition algorithm involves approximating the second-stage objective function through a series of supporting hyperplanes that guarantee convergence to an optimal first-stage solution. For example, the L-shaped algorithm [114] decomposes the problem into first-stage (or master) problem and second-stage problem, also known as the subproblem. This algorithm is an iterative method that approximates the second-stage value function with a convex combination of Benders algorithm cuts [14] for each subproblem scenario and a given first-stage solution.

Two-stage stochastic mixed-integer programs are a class of problems when a subset of the decision variables are discrete. The problem with solving the two-stage stochastic mixed-integer programs is that they lack many of the necessary properties that enable designing efficient algorithms apposed to other branches of optimization. We refer the reader to [1, 18, 66, 67, 76] for a surveys of algorithms for two-stage stochastic programming problems.

2.3 MARKOV DECISION PROCESSES (MDPS)

In this section, we briefly review the basic concepts and definitions of Markov decision processes (MDPs). More details are provided elsewhere [15, 94]. MDPs provide a mathematical framework to model sequential decision making problems, where outcomes are partly random and partly under the control of the decision maker. More specifically, we focus our attention on those sequential decision making problems under uncertainty where the decisions are made at discrete points of time, referred to as the *decision epochs*. Let $T = \{1, ..., N\}$ be the set of decision epochs, where N need not be finite. At each decision epoch t, the system occupies a *state* s in the *state space*, S of the process, which is assumed to be known to the decision maker. This modeling framework provides the decision maker a control over the system by choosing an *action* a from the *action space* A_s at each decision epoch t. We content ourself with so-called *stationary* MDPs where the model parameters do not vary over time. Furthermore, we assume that both S and A_s are discrete and finite. The evolution of the process is governed probabilistically, as the decision maker chooses action a in state s at decision epoch t, where the state of the system at the next decision epoch $t+1$ is determined by the *transition probability distribution* $P\{\cdot|s, a\}$. As a result of choosing action a in state s at each decision epoch, the decision maker receives an immediate reward $r(s, a)$. More general treatment of this subject allows the dependency of the rewards to not only the current state s and chosen action a, but also to the next visited state s' at the next decision epoch denoted by $r(s, a, s')$. However, this dependency does not incur further complexity to the model as we can consider the its expected value by computing

$$r(s, a) = \sum_{s' \in S} r(s, a, s') P\{s'|s, a\}.$$

We refer to the collection of objects $\{T, S, A_s, P_t\{.|s, a\}, r(s, a)\}$ as a *Markov decision process*.

The decision maker's strategy for choosing actions throughout the lifetime of the system is known as a *decision rule*, denoted by d_t. In this dissertation, we restrict our attention to the *Markovian* decision rules under which the action choice under d_t only depends on the current state of the proces and is independent of the entire history of the system.

Decision rules can be either *randomized* or *deterministic*. Given a randomized decision rule, at each state an action is chosen based on a non-degenerate probability distribution, whereas a deterministic decision rule prescribes to choose an action with certainty (i.e., with probability 1). Deterministic decision rules are often preferred to the randomized ones due to their simplicity to implement. However, in an MDP with finite state and action sets, there exists a deterministic decision rule that is always optimal provided that the states of the process are completely observable [94]. A policy δ specifies which decision rule to use at each decision epoch t, denoted as $\delta = (d_1, d_2, ..., d_{N-1})$. A stationary policy prescribes the same decision rule at all decision epochs, i.e., $d_t = d$ for all $t \in T$ and are easy to implement. An *infinite-horizon* MDP ($N = \infty$) with finite state space and action set always has an optimal stationary policy given that the states are completely observable [94].

We restrict our attention to the *expected total discounted reward* criterion:

$$\mathbb{E}\left[\sum_{t=0}^{N-1} \lambda^t r_r(s,a)\right], \tag{2.14}$$

where $0 \leq \lambda < 1$ is the discount rate. For an infinite-horizon problem, an optimal stationary Markovian policy that maximizes the expected total discounted reward can be found by solving the following recursive equations also known as the Bellman's optimality equations:

$$v^*(s) = \max_{a \in A_s} \left\{ r(s,a) + \lambda \sum_{s' \in S} P\{s'|s,a\} v^*(s') \right\}, \tag{2.15}$$

where $v^*(s)$ is the optimal value function in state s. Thus, the optimal action a^* is one that maximizes the right-hand side of equation (2.15).

The optimality equations (2.15) can be solved using a variety of methods, including value iteration [15, 19, 94], policy iteration [13, 61] and linear programming [35]. One deficiency of these methods is that they often fail to work as the MDP problem becomes large, referred to as the "curse of dimensionality" [13, 28]. To overcome these this issue, a broad range of efficient solution techniques have been proposed, including variants of the basic methods such as modified policy iteration[84], relative value iteration [94] among many others.

Discounted MDPs can also be formulated as linear programs. The linear programming formulation of a discounted Markov decision process problem with $\alpha(j)$, for $j \in S$ as any positive scalar is as follows:

$$\min \quad \sum_{j \in S} \alpha(j) v(j) \tag{2.16}$$

$$\text{s.t.} \quad v(s) - \sum_{j \in S} \lambda p(j|s,a) v(j) \geq r(s,a), \qquad \forall a \in A_s, \forall s \in S, \tag{2.17}$$

When the positive scalars $\alpha(j)$ for all $j \in S$ are chosen such that $\sum_{j \in S} \alpha(j) = 1$, they can be interpreted as a probability distribution over the state space of the process S. The dual of LP formulation of a discounted MDP problem is usually more informative. The dual decision variables for each state $s \in S$ and action choice $a \in A_s$ are denoted by $x(s,a)$ and represent the total discounted joint probability under initial state distribution $\alpha(j)$ for all $j \in S$ that the system occupies state s and chooses action a. The dual formulation is as follows:

$$\min \quad \sum_{s \in S} \sum_{a \in A_s} r(s,a) x(s,a) \tag{2.18}$$

$$\text{s.t.} \quad \sum_{a \in A_j} x(j,a) - \sum_{s \in S} \sum_{a \in A_s} \lambda p(j|s,a) x(s,a) = \alpha(j), \qquad \forall a \in A_j, \forall j \in S, \tag{2.19}$$

$$x(s,a) \geq 0, \qquad \forall a \in A_s, \forall s \in S. \tag{2.20}$$

Note that the dual LP formulation has fewer constraints compared to the primal formulation, hence, solving the dual formulation may be preferable to solving the primal.

2.4 SPAR: STOCHASTIC PROGRAMMING WITH ADVERSARIAL RECOURSE

As noted earlier, stochastic programs and Markov decision processes are powerful mathematical programming tools to model a variety of problems. In this section, we briefly discuss a general modeling technique referred to as stochastic programming with adversarial recourse (SPAR). This method combines the best aspects of stochastic programming and MDPs in a

unified framework. We refer the reader to [8, 11] for details of SPAR. SPAR involves optimally designing a system of interest with the knowledge that an adversary may subsequently attempts to degrade the system. This problem can be formulated as a two-stage stochastic programming model. The first-stage decisions correspond to selecting a system design in the face of uncertainty in the adversarial effect. Once a design decision is made, an adversary exercises his choice of best possible decisions that can incur the most disruption to the system, i.e., negatively impact the system performance, and represent the second-stage recourse problem. This second-stage recourse problem is modeled as an MDP whose parameters are defined by the design uncertainty.

The objective of system designer (first-stage decision maker) is to select design that minimizes the design selection cost and the expected discounted cost of adversary's optimal decisions. The general mathematical programming formulation of SPAR is as follows.

$$\begin{aligned} \min \quad & c^T x + \mathbb{E}_\xi[Q(x,\xi)] \\ \text{s.t.} \quad & x \in X. \end{aligned} \tag{2.21}$$

In this model, X denoted the set of all feasible system designs and ξ is a discretely distributed random variable with finite support Ξ that represents the design uncertainty. Given a design x, $Q(x,\xi)$ represents the expected reward of the optimal policy of the associated adversarial MDP under scenario ξ. As noted earlier, the parameters of the adversarial MDP are defined based on the possible scenarios. In SPAR, the design decisions x are linked to the subsequent MDP using a bounded adversary impediment values. These values denoted by $t_i(s,a,\xi) \geq 0$ for $i = 1, ..., n$ with $x = \{x_1, ..., x_n\}$ are considered as the benefits to the system designer and link the elements of design decision vector x to the reward of an action choice a from the states a in the realized MDP under scenario ξ. Hence, the modified rewards of subsequent MDP under scenario ξ are defined as:

$$r(s,a,\xi) - \sum_{i=1}^{n} t_i(s,a,\xi)x_i. \tag{2.22}$$

The special structure of SPAR can be exploited to solve efficient methods, see [8] for a modified L-shaped method using the recourse function properties of SPAR. However, the efficiency of any specific method to solve SPAR is highly problem dependent.

2.5 BRANCH AND PRICE

Column generation is a method for solving mathematical programming problems that have too many variables to consider explicitly. Because most of the variables in such large-scale linear programming problems will be non-basic in the optimal solution, only a small subset of variables need to be considered explicitly for solving the problem. Hence, column generation is based on the idea to generate only those variables that may potentially improve the objective function, i.e, generate the columns as needed. To do so, column generation splits the problem to be solved into a master problem and a set of pricing problems. The master problem is the original problem restricted to a subset of variables. The objective function of the pricing problem is to find a column with the most favorable reduced cost with respect to the current dual variables obtained by solving the master problem. At each iteration of the column generation, the pricing problem is solved and if favorable columns exist, some subset of them are inserted into the master problem; if not, the algorithm stops. Column generation was first introduced by Ford and Fulkerson [53] to enable the implicit handling of variables in a multi-commodity flow problem. Dantzig and Wolfe [32] established column generation as a powerful technique for large-scale mathematical programming by utilizing this method in the algorithm known as Dantzig-Wolfe decomposition. The idea of combining the technique of column generation with an LP-based branch-and-bound algorithm was first suggested by [12].

Column generation and branch-and-price have been successfully applied to an enormous number of deterministic applications, including cutting stock problems [55], airline crew scheduling [9], multi-commodity flow [2], vehicle routing [37, 38], and organ allocation problems [33, 68]. We refer the reader to [10] and [77] for a complete description of the branch-and-price framework.

3.0 BALANCING EQUITY AND EFFICIENCY IN LIVER ALLOCATION VIA REVISED HEALTH REPORTING FREQUENCIES

3.1 INTRODUCTION

In Chapter 1, we described how UNOS prioritizes End-Stage Liver Disease (ESLD) patients who are on the liver waiting list by considering: (i) their geographical location and (ii) patients' medical urgency. The prioritization policy within a given geographic area is based on the patients' most recently reported health status, although blood type compatibility and waiting time on the list are used to break ties. Thus, UNOS obligates patients to update their health status within a timeframe that depends on their last reported health. We refer to these health updating (reporting) requirements as an "updating scheme."

Currently, the priority of over 99% of adult ESLD patients is assessed by the model for end-stage liver disease (MELD) [52]. Hence, we restrict our attention to the MELD patients. Patients can update more frequently than what the updating scheme dictates. However, if a patient fails to update by the required time, UNOS downgrades her MELD score to 6, the healthiest score, until new results are received. Because the chance of receiving an organ offer increases with a rise in patient's last reported MELD score, UNOS requires patients to update more frequently as they get sicker.

The ability of patients to update anytime within the required timeframe allows them to "game" the liver allocation system. Because reporting a healthier health status may decrease a patient's chance of receiving an organ offer, she may decide to delay reporting improved health status. On the other hand, reporting a sicker health status may leave the patient with a shorter amount of time until the next required update. Hence, a patient whose health deteriorates may conceal her health status change by not reporting it. Both

these situations illustrate information asymmetry between UNOS and patients that may lead to a misallocation of livers. This information asymmetry can be alleviated through more frequent updating requirements, but at a price of an increased data-processing burden. This data-processing burden is associated with the existing significant data collection costs incurred at both transplantation centers and UNOS [100], as well as patients' inconvenience due to more frequent clinical tests.

In this chapter, we study the problem of mitigating information asymmetry in the liver allocation system by revising updating requirements. Specifically, we propose an approach to determine a set of Pareto-optimal updating schemes that simultaneously minimize the information asymmetry and the data-processing burden. This approach focuses on a joint patient and societal perspective and involves solving a set of parametric stochastic optimization problems. The model parameters at different iterations denote various desired tolerance levels for the data-processing burden defined by policymakers. Hence, at each iteration we seek to identify an updating scheme that minimizes the information asymmetry while preserving a given desired tolerance level for the data-processing burden.

Our modeling approach captures the characteristics of patients on the liver waiting-list by using a set of scenarios, each of which represents a class of autonomous self-interested patients. This set of patient classes can be defined based on the clinical and demographical characteristics of patients. We also incorporate the adversarial effect of patients' decisions, referred to as the "patient decision process", who may game the system into the updating scheme selection problem. We exploit the structural properties of the proposed modeling approach and present a decomposition algorithm to solve instances of parametric model. Finally, we extend our solution method to determine the exact efficient frontier of the Pareto optimal updating schemes.

The rest of this chapter is organized as follows. In Section 3.2, we review the prior related literature and discuss our contributions. The outline of our modeling framework is discussed in Section 3.3. In Section 3.4, we present a Markov Decision Process (MDP) formulation of patients' decision-making problem, and discuss the equity and efficiency measures to quantify the information asymmetry and data processing burden for any updating scheme, respectively. We then investigate several structural properties of patients' optimal policy.

In Section 3.5, we first present a mathematical formulation of the model to determine an updating scheme with minimum system inequity for a desired level of data processing burden. We then discuss the structural properties of this model and propose a decomposition-based algorithm to solve it. Our approach to approximate the actual efficient frontier of the Pareto-optimal solutions is presented in Section 3.6, and Section 3.7 includes the details of our parameter estimation and numerical results. Concluding remarks and the future research directions are provided in Section 3.8.

3.2 LITERATURE REVIEW AND OUR CONTRIBUTIONS

As noted in Chapter 1, donated organs are scarce and vital resources. Hence, improving the organ allocation and transplantation procedures is crucial. Recognizing this need, operations research applications in this area have grown significantly over the past two decades. Among these studies, a number of researchers focus on a patient's perspective and investigate the problem of accepting/rejecting an organ to maximize a special welfare measure of the patient [4, 5, 6, 60, 105]. Another group focuses on designing optimal organ allocation policies with respect to a curtain measure of societal welfare [16, 69, 110]. Finally, a third group of researchers considers a joint patient and the societal perspective [109, 111]. We refer the reader to [33, 70, 103] for surveys of the literature.

To the best of our knowledge, Içten [62] is the only study that addresses information asymmetry in the liver allocation mechanism. Focusing on a patient's perspective, she first addresses the degree to which a patient can benefit from the flexibility in an updating scheme. To do so, she investigates a decision process in which a patient exercises her choice of action between updating her health status, doing nothing without transplantation, or accepting an organ offer (if any) to maximize her life expectancy. She formulates this problem as an infinite horizon discrete-time MDP problem that maximizes patient's total expected discounted life-time. The decisions are made daily and the state of the process constitutes of: (1) patient's true health status, (2) her last reported health status, (3) remaining time until the next updating requirement; and (4) the quality of offered liver (if any). She calibrates the

model parameters using clinical data and explores several structural properties of a patient's optimal decisions under mild conditions. She uses an MDP to evaluate the inequity and efficiency measures of an updating scheme, as defined in Section 3.3. Finally, for a cohort of patients and a heuristically chosen of alternative updating schemes, several MDP problems are solved to heuristically approximate the efficient frontier of Pareto-optimal solutions. The patient decision model in [62] provides deeper insights to investigate a single patient's gaming ability who may exploit the health reporting system. However, the problem of mitigating information asymmetry remains unaddressed since the modeling approach in [62] can only be employed to evaluate the performance of a given updating scheme. Our specific contributions in this chapter are as follows:

- We propose an exact approach to identify a set of Pareto-optimal updating schemes to balance the trade-off between information asymmetry and data processing burdens.
- Our modeling framework enables us to incorporate the gaming effect of a set of autonomous self-interested patients' decisions into the updating scheme selection problem.
- Unlike a single-patient-focused model of [62], our proposed model can incorporate various restrictions in selecting a particular form of an updating scheme. For example, we can identify the exact efficient frontier of those updating schemes that are compliant with the current MELD aggregation scheme of UNOS, i.e. specific MELD scores require the same updating timeframe.
- Our solution method is based on the structural properties of the USDP problem that avoids enumerating all potential updating schemes to obtain an optimal solution.

3.3 DESIGNING UPDATING SCHEMES

As noted in Section 3.1, the information asymmetry in allocating livers can be alleviated by more frequent updating requirements at the price of an increase in data processing burden. Thus, we propose a multi-objective optimization problem that simultaneously considers the two possibly conflicting objectives of: (i) minimizing the system inequity as measured by the increase in patients' expected lifetime due to their gaming ability, and (ii) maximizing the

system efficiency as measured by the reduction in the data possessing burden. To do so, we adopt an ϵ-constraint approach to multi-objective optimization problems [26, 42, 44], where we iteratively solve a sequence of parametric two-stage stochastic programming problems, referred to as "updating scheme design problem" (USDP). The objective of this iterative method is to identify the exact efficient frontier of Pareto-optimal updating schemes that minimize the system inequity with respect to the tolerance levels of system efficiency. Figure 4 illustrates a hypothetical efficient frontier of the Pareto optimal updating schemes, denoted by black points.

Figure 4: Hypothetical Pareto-optimal updating schemes

The objective of USDP is to identify an updating scheme that minimizes the expected system inequity for a given minimum required reduction in the data processing burden, denoted by μ. Figure 5 illustrates the decision flow in USDP for a given μ, denoted by USDP[μ], that consists of two phases, referred to as the design phase (first-stage decision process) and the patients' decision-making model also known as the second-stage recourse problems. The design phase decisions in this model are made in the face of uncertainty in the cohort of patients on the liver waiting list in the long run, referred to as design uncertainty. This design uncertainty is not revealed until and updating scheme is chosen, and is represented through a set of autonomous self-interested patient types. Each patient type is defined based on patient demographic and clinical characteristics. After an updating

scheme is chosen, a set of patient decision-making problems are defined for each patient type. Each patient decision-making problem is modeled as an infinite horizon discounted MDP problem with the objective of determining patient's optimal sequence of decisions in: (1) updating her health status, (2) doing nothing without transplantation; or (3) accepting a liver offer (if any) and transplant so as to maximize her expected lifetime. These optimal policies are then used to approximate the anticipated system inequity and efficiency under for the chosen updating scheme.

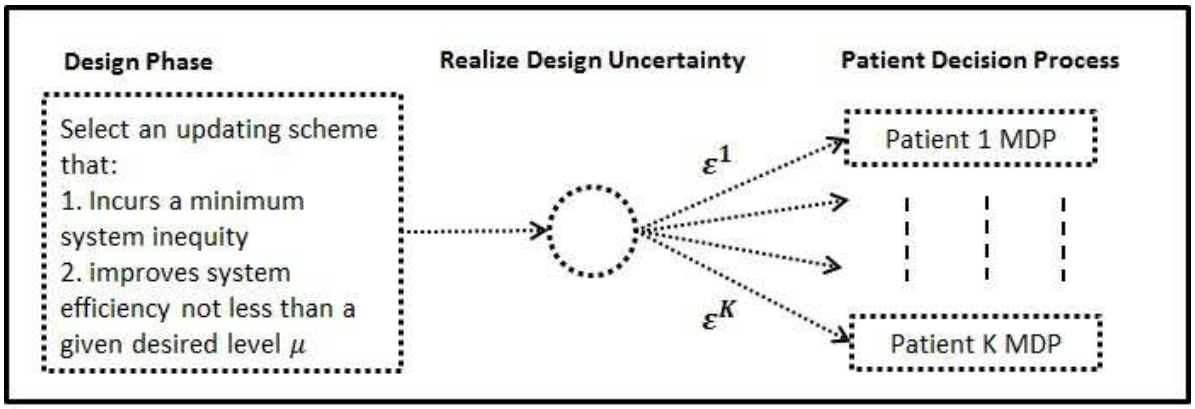

Figure 5: USDP[μ] model representation

We use the following notation and assumptions throughout the rest of this chapter. The scenario space of USDP[μ] is represented by a finite set Ξ of discrete patient classes, denoted by $\xi^1, ..., \xi^K$, with q^k as the probability of realizing patient class ξ^k. We denote each patient class ξ^k as k for notational continence. Let $S^{\mathcal{M}} = \{M_1, ..., M_{\max}\}$ be an ordered set that represents all possible heath statuses of a patient. For the case of liver allocation in the U.S., $M_1 = 6$ and $M_{\max} = 40$. However, we represent $M_1 = 1$ and $M_{\max} = |S^{\mathcal{M}}|$ for the ease of notation. Furthermore, the minimum and the maximum allowable updating frequencies at any MELD score $i \in S^{\mathcal{M}}$ is denoted by $F_{\min}$ and $F_{\max}$, respectively, and $\mathcal{F} = \{f \in Z_+ | F_{\min} \leq f \leq F_{\max}\}$ denotes the set of all possible integer valued updating frequencies. The set of binary decision variables $x_{ij} = 1$ if patients who currently report a MELD score of $i \in S^{\mathcal{M}}$, should next update their health status not later than j periods of time, and $x_{ij} = 0$ otherwise. Hence, x denotes an updating scheme in our notation. Finally,

we restrict our attention to monotonic updating schemes that requires patients to update no less frequently as they get sicker. Monotonic updating schemes are consistent with the prioritization rules of UNOS.

3.4 PATIENT DECISION-MAKING PROBLEM

In this section, we extend the model introduced by Içten [62] to formulate patients' decisions regarding updating/doing nothing/accepting an organ offer (if any) for each patient type and for a given updating scheme x. To do so, we formulate a patient's decision-making process as a discrete-time, infinite horizon MDP model that maximizes the total expected lifetime of a patient of type k for a given updating scheme x. This model involves a sequence of patient's decisions that determines if the patient should: (i) update her health status, (ii) do nothing without transplantation, or (iii) accept a liver offer (if any) at each decision point, so as to maximize her expected lifetime. We formulate this problem as a discrete-time, infinite horizon MDP model defined for each patient type and a given updating scheme. Next in Section 3.4.2, we introduce the details of the inequity and efficiency measures to evaluate the information asymmetry and data processing burdens under a given updating scheme. To quantify these measures, we use the patients' optimal policies obtained by solving the proposed MDP model. In Section 3.4.3, we investigate several structural properties of patients' optimal updating policies that are subsequently used to devise an efficient solution method to solve instances of USDP.

3.4.1 Mathematical Formulation

For each patient type k, $1 \leq k \leq K$ and a given updating scheme x, we model the patient's decision-making problem as a discrete-time, infinite horizon MDP problem. The decision epochs in the MDP model are discrete points of time, and decisions are made periodically at the beginning of each epoch. The state space of the process, denoted by (h, m, τ, ℓ) consists of: $h \in S^{\mathcal{H}} = \mathcal{S}^M \cup \{\Delta\}$, the patient's current health status defined by MELD scores $\mathcal{S}^M$ with

Δ denoting death; $m \in S^{\mathcal{M}}$, the patient's last reported MELD score; $\tau \in \{0, 1, ..., F_{\max}\}$, remaining time until the next required update; and $\ell \in S^{\mathcal{L}} = \{1, 2, ..., L+1\}$, the quality of a deceased-donor liver offer, where lower numbers represent better quality livers, and $L+1$ denotes no liver offer. Unlike the definition of τ in the state space description of the MDP model in [62], the state of our proposed model is independent of the choice of any updating scheme. Hence, the state space of the model in [62] is a special case of our proposed MDP. The action set of the extended model is scenario independent and is given by [62]:

$$A(h, m, \tau, \ell) = \begin{cases} T = \text{Transplant},\ DN = \text{Do nothing},\ U = \text{Update} & \ell < L+1, \\ DN = \text{Do nothing},\ U = \text{Update} & \ell = L+1. \end{cases}$$

The transition probability matrix $\mathcal{H}^k$ models the health state changes of the patient type k, for $1 \leq k \leq K$, where $\mathcal{H}^k(h'|h)$ denotes the probability that a patient with current MELD score of h transitions to MELD score h' in the next decision point given that he/she does not transplant, for all h and h' and $\mathcal{H}^k(\Delta|\Delta) = 1$. Additionally, for each scenario k, $\mathcal{L}^k$ models the probability of receiving a liver offer of quality ℓ given that patient's last reported MELD score is m, represented by $\mathcal{L}^k(\ell|m)$.

For each patient type k, we define $r^k(h, m, \tau, \ell, a)$ as the reward of taking action $a \in A(h, m, \tau, \ell)$ in state (h, m, τ, ℓ) regardless of the chosen updating scheme. We consider $r^k_{DN}(h) \in [0, \infty)$ and $r^k_U(h) \in [0, \infty)$ for $h \in \mathcal{S}^M$, and $r^k_{DN}(h) = r^k_U(h) = 0$ for $h = \Delta$ as the rewards of doing nothing and updating actions for patient type k, respectively. Furthermore, let $R^k(h, \ell) \in [0, \infty)$ for $h \in \mathcal{S}^{\mathcal{M}}$, with $R^k(\Delta, \ell) = 0$ for $\ell < L+1$ and $R^k(h, L+1) = 0$ be the post-transplant lump sum reward when a liver offer of quality ℓ in health state h by patient type k is accepted. Hence, the updating scheme independent rewards of the process, $r^k(h, m, \tau, \ell, a)$, are defined as:

$$r^k(h, m, \tau, \ell, a) = \begin{cases} r^k_{DN}(h) & \text{for } a = DN, \\ r^k_U(h) & \text{for } a = U, \\ R^k(h, \ell) & \text{for } a = T. \end{cases}$$

Our model incorporates patients' decision-making model into the updating scheme selection process. To do so, we link the scheme selection decisions x to the rewards of the MDPs for each scenario k through specific penalty values. Hence, the updating scheme dependent rewards of the model for a patient type k under an updating scheme x, denoted by $\overline{r}_x^k(h, m, \tau, \ell, a)$, are defined as:

$$\overline{r}_x^k(h, m, \tau, \ell, a) = r^k(h, m, \tau, \ell, a) - \sum_{i \in \mathcal{S}^M} \sum_{j \in \mathcal{F}} t_{ij}^k(h, m, \tau, \ell, a) \cdot x_{ij}, \tag{3.1}$$

with $t_{ij}^k(h, m, \tau, \ell, a)$ as the impediment values [8]. These impediment values can assume any arbitrary large value, e.g., any upper bound on the total expected discounted lifetime of a patient type k at any state of the process, denoted by Y^k. The following lemma provides an lower bound for the maximum possible life expectancy of a patient which is subsequently used to define the impediment values.

Lemma 3.4.1. *An upper bound on the total expected discounted lifetime of patient type $k \in \{1, ..., K\}$, is given by:*

$$\frac{\max_{h \in S^M} r^k(h)}{1 - \max_{h \in S^M} \sum_{h' \in S^M} \mathcal{H}^k(h'|h)} + \max_{h \in S^M, \ell \in S^{\mathcal{L}}} R^k(h, \ell), \tag{3.2}$$

where $r^k(h) = \max\{r_{DN}^k, r_U^k\}$.

Hence, we define the each impediment value $t_{ij}^{\xi^k}(h, m, \tau, \ell, a)$ for each state (h, m, τ, ℓ) when action $a \in A(h, m, \tau, \ell)$ is chosen as:

$$t_{ij}^k(h, m, \tau, \ell, a) = \begin{cases} r^k(h) + Y^k & \text{if } (\tau \geq j) \text{ or } (j = 1, h \neq m) \text{ for } a = DN \text{ or } U, \\ R^k(m, \ell) & \text{if } (\tau \geq j) \text{ or } (j = 1, h \neq m) \text{ for } a = T, \\ r^k(h) + Y^k & \text{if } j = 1 \text{ and } h = m \text{ for } a = DN, \\ 0 & \text{otherwise,} \end{cases} \tag{3.3}$$

where Y^k is set to their upper bound as defined in (3.2). Our proposed patient decision model further extends [62] in the following two ways: (i) the rewards of the process are both scenario and first-stage decision dependent; (ii) we differentiate between the immediate rewards associated with doing nothing and updating actions.

For a given discount factor $\lambda \in [0,1]$ and an updating scheme x, we define $v_x^k(h,m,\tau,\ell)$ as the maximum expected total discounted life days of a patient type k who starts from state (h,m,τ,ℓ):

$$v_x^k(h,m,\tau,\ell) = \begin{cases} \max\{v_x^{k,T}(h,\ell), v_x^{k,DN}(h,m,\tau), v_x^{k,U}(h)\} & \forall h,m,\tau,\ell < L+1, \\ \max\{v_x^{k,DN}(h,m,\tau), v_x^{k,U}(h)\} & \forall h,m,\tau,\ell = L+1, \end{cases} \tag{3.4}$$

where

$$v_x^{k,T}(h,\ell) = \overline{r}_x^k(h,m,\tau,\ell,T), \tag{3.5}$$

$$v_x^{k,DN}(h,m,\tau) = \overline{r}_x^k(h,m,\tau,\ell,DN) + \lambda \sum_{h' \in S^H} \mathcal{H}^k(h'|h) \left\{ \sum_{\ell' \in S^{\mathcal{L}}} \mathcal{L}^k(\ell'|m) v_x^k(h',m,\tau-1,\ell') \right\}, \tag{3.6}$$

$$v_x^{k,DN}(h,m,0) = \overline{r}_x^k(h,m,\tau,\ell,DN) + \lambda \sum_{h' \in S^H} \mathcal{H}^k(h'|h) \left\{ \sum_{\ell' \in S^{\mathcal{L}}} \mathcal{L}^k(\ell'|m) v_x^k(h',M_1,\overline{\tau}_x(M_1)-1,\ell') \right\}, \tag{3.7}$$

$$v_x^{k,U}(h) = \overline{r}_x^k(h,m,\tau,\ell,U) + \lambda \sum_{h' \in S^H} \mathcal{H}^k(h'|h) \left\{ \sum_{\ell' \in S^{\mathcal{L}}} \mathcal{L}^k(\ell'|h) v_x^k(h',h,\overline{\tau}_x(h)-1,\ell') \right\}. \tag{3.8}$$

In the definition of the optimality equations, $\overline{\tau}_x : S^{\mathcal{M}} \to R_+$ denotes a mapping that determines updating frequencies assigned to the MELD scores under updating scheme x.

Proposition 3.4.2. *Given an updating scheme x and patient type k, transplantation is optimal if $\tau \geq \overline{\tau}_x$ in all state (h,m,τ,ℓ) for all h,m,ℓ.*

Proof. For a given patient type k and updating scheme x, in any state (h,m,τ,ℓ) where $\tau \geq \overline{\tau}_x$,

$$v_x^{k,DN}(h,m,\tau) < -Y^k + \lambda \sum_{h' \in S^H} \mathcal{H}^k(h'|h) \left\{ \sum_{\ell' \in S^{\mathcal{L}}} \mathcal{L}^k(\ell'|m) v_x^k(h',m,\tau-1,\ell') \right\} \tag{3.9}$$

$$\leq Y^k(\lambda \sum_{h' \in S^{\mathcal{M}}} H^k(h'|h) - 1) \leq 0. \tag{3.10}$$

where inequalities (3.9) and (3.10) follow by (3.3) and (3.2), respectively. Similarly,

$$v_x^{k,U}(h,m,\tau) < -Y^k + \lambda \sum_{h' \in S^H} \mathcal{H}^k(h'|h) \left\{ \sum_{\ell' \in S^{\mathcal{L}}} \mathcal{L}^k(\ell'|h) v_x^k(h',h,\overline{\tau}_x(h)-1,\ell') \right\} \tag{3.11}$$

$$\leq Y^k(\lambda \sum_{h' \in S^{\mathcal{M}}} H^k(h'|h) - 1) \leq 0. \tag{3.12}$$

□

Proposition 3.4.3. *Given an updating scheme x, such that $\overline{\tau}_x(m) = 1$ for all $m \in S^{\mathcal{M}}$:*

(a). It is always optimal to transplant in state (h, m, τ, ℓ) if $h \neq m$,

(b). The optimal action is either to transplant or update in state (h, m, τ, ℓ) if $h = m$.

Proof.

$$v_x^{k,DN}(h, m, 0) < -Y^k + \lambda \sum_{h' \in S^H} \mathcal{H}^k(h'|h) \left\{ \sum_{\ell' \in S^{\mathcal{L}}} \mathcal{L}^k(\ell'|m) v_x^k(h', M_1, \overline{\tau}_x(M_1) - 1, \ell') \right\} \tag{3.13}$$

$$\leq Y^k(\lambda \sum_{h' \in S^{\mathcal{M}}} H^k(h'|h) - 1) \tag{3.14}$$

$$\leq 0. \tag{3.15}$$

where inequalities (3.13) and (3.14) by (3.3) and (3.2), respectively. With a similar argument, $v_x^{k,U}(h, m, 0) < 0$ which completes the proof of part (a). The proof of part (b) is similar and is omitted. □

Our proposed patient's decision model provides a more realistic decision model compared to [6, 62] by: (i) considering a more general state space description that is independent of the choice of any particular updating scheme, and (ii) relaxing the restricted assumption of identical immediate rewards for doing nothing and updating actions. Moreover, the definition of process rewards that links the updating scheme decisions to the patients' decition model enables us embed patients' MDPs into a general modeling framework to design updating schemes.

3.4.2 Modeling the Inequity and Efficiency of Updating Schemes

We let $Q(x, \xi^k)$ denote the maximum benefit of patient type k as a measure of an increase in patient's life expectancy due to her the gaming ability under updating scheme x. Similarly, $G(x, \xi^k)$ denotes the benefit of patient type k through reduction in the data-processing burden. These values are patient-specific measures of inequity and efficiency, respectively. Before formally defining these performance measures, we introduce some additional notation. We consider x_w as the updating scheme that obligates patients to update their MELD scores to the most frequent extreme. Hence, x_w in our model is analogous to the continuous

updating scheme in [5, 6, 104, 105] that results in no information asymmetry between UNOS and patients. As discussed in Section 3.1, updating scheme x_w leaves patients with no gaming ability by not reporting their MELD score changes. However, it will increase the data processing burden to its maximum possible value.

For a given updating scheme x, we define a set of all possible policies of a patient type k associated with patient's decision process by $\Pi^k(x)$. Hence, for each patient type k, each policy $\pi^k \in \Pi^k(x)$ defines patient's individual state-specific actions when updating scheme x dictates the updating requirements. Furthermore, let $\hat{\pi}^k$ patient k's optimal policy associated with the Bellman's optimality equations (3.4)-(3.8).

Içten [62] characterizes the inequity and efficiency induced by selecting a particular updating scheme for a single patient. She quantifies the source of induced inequity as the weighted average percentage increase in the expected lifetime and the efficiency as weighted average percentage decrease in the number of expected updates compared to the same measures under the daily updating scheme. In this dissertation, we use her approach to quantify the expected system inequity and efficiency with a slight change in the notation.

For a given updating scheme x and a patient type k with MELD score h and no liver offer at the time of listing, the percentage increase in expected lifetime compared to the continuous updating scheme, x_w, is:

$$\omega_x^k(h) = \frac{v_x^k(h, h, \overline{\tau}_x(h) - 1, L + 1) - v_{x_w}^k(h, h, \overline{\tau}_{x_w}(h) - 1, L + 1)}{v_{x_w}^k(h, h, \overline{\tau}_{x_w}(h) - 1, L + 1)} \times 100. \tag{3.16}$$

Similarly, the percentage decrease in the expected number of updates is calculated using

$$\phi_x^k(h) = \frac{U_{x_w}^{\hat{\pi}^k}(h, h, \overline{\tau}_{x_w}(h) - 1, L + 1) - U_x^{\hat{\pi}^k}(h, h, \overline{\tau}_x(h) - 1, L + 1)}{U_{x_w}^{\hat{\pi}^k}(h, h, \overline{\tau}_{x_w}(h) - 1, L + 1)} \times 100, \tag{3.17}$$

where $U_x^{\hat{\pi}^k}(h, h, \overline{\tau}_x(h) - 1, L + 1)$ is the expected number of times that a patient denoted by scenario k who starts with no liver offer and MELD score h at the time of listing updates, following the optimal policy π^k. To determine the expected number of times that a patient updates, we propose a reward process where transplant and do nothing actions induce an immediate reward of zero and updating incurs a reward of one.

Following the patient's optimal updating policy, we can subsequently calculate the expected number of health updates. Using the aforementioned notation, we define the expected inequity under updating scheme x for patient type ξ^k, $Q(x, \xi^k)$ as:

$$Q(x, \xi^k) = \sum_{h \in S^{\mathcal{M}}} p^k(h) \times \omega_x^k(h), \tag{3.18}$$

where $p^k(h)$ is the scenario-dependent probability that a patient of type k registers to the waiting list with MELD score h. Furthermore, the expected efficiency value of system under updating scheme x for patient type k is defined as:

$$G(x, \xi^k) = \sum_{h \in S^{\mathcal{M}}} p^k(h) \times \phi_x^k. \tag{3.19}$$

Finally, we express the total system inequity, $\mathbb{E}_\xi[Q(x, \xi^k)]$, and total system efficiency, $\mathbb{E}_\xi[G(x, \xi^k)]$, under updating scheme x are defined as the expected values of patient-specific inequity and efficiency measures on the support space of scenarios, as:

$$\mathbb{E}_\xi[Q(x, \xi^k)] = \sum_{k=1}^{K} \sum_{h \in S^{\mathcal{M}}} q^k Q(x, \xi^k), \tag{3.20}$$

and

$$\mathbb{E}_\xi[G(x, \xi^k)] = \sum_{k=1}^{K} \sum_{h \in S^{\mathcal{M}}} q^k G(x, \xi^k). \tag{3.21}$$

3.4.3 Structural Properties of the Optimal Patient Policy

We use the two following clinically realistic assumptions to investigate the structural properties the patients' optimal updating policy.

Assumption 3.4.4. *The post-transplant rewards, $R^k(h, \ell)$, are decreasing in h and ℓ, for $1 \leq k \leq K$.*

Assumption 3.4.5. *The rows of the liver transition probability matrix $\mathcal{L}^k$ are in decreasing stochastic order, i.e., $\sum_{\ell=i}^{L+1} \mathcal{L}^k(\ell|m) \geq \sum_{\ell=i}^{L+1} \mathcal{L}^k(\ell|m+1)$, $\forall m$, $1 \geq i \geq L+1$ and $1 \leq k \leq K$.*

For all patient types, Assumption 3.4.4 implies that the post-transplant rewards are nonincreasing as a patient get sicker, or the quality of the offered liver is degraded for all patient types. Furthermore, Assumption 3.4.5 indicates that the chance of receiving a higher quality liver offer is nondecreasing as a patient gets sicker for all patient types. Note that these assumptions are clinically reasonable and are supported by clinical data, as shown in Section 3.7. In Proposition 3.4.6 we show the intuitive facts that for a given updating scheme x and a patient type k, (a) if the patient is in the healthiest status, then the remaining time until the next required update is irrelevant, (b) it is better to have a liver offer with a higher quality, (c) it is better to have a sicker health status recorded by UNOS, and (d) it is better to have more time left until the next required updating time.

Proposition 3.4.6. *Given an updating scheme x and a patient type k, for $\tau < \overline{\tau}_x(m)$:*

(a). $v_x^k(h, M_1, \tau, \ell)$ is constant in τ for all h and ℓ.

(b). Under Assumption 3.4.4, $v_x^k(h, m, \tau, \ell)$ is decreasing in ℓ for each h, m, τ .

(c). Under Assumptions 3.4.4 and 3.4.5, $v_x^k(h, m, \tau, \ell)$ is increasing in m for each h, τ, ℓ.

(d). Under Assumptions 3.4.4 and 3.4.5, $v_x^k(h, m, \tau, \ell)$ is increasing in τ for each h, m, ℓ.

Proof. We proceed with the steps of the value iteration algorithm, with $v_x^{n,k}(h, m, \tau, \ell)$ as the value function at the nth step of the algorithm, for patient type k and a given updating scheme x. Additionally, let $v_{x,a}^{n,k}$ be the value function in step n of the algorithm when action $a \in \{DN, U, T\}$ is chosen.

(a). Without loss of generality, let $v_x^{0,k}(h, M_1, \tau, \ell) = 0$ for $h \in \mathcal{S}^H, \tau \in \mathcal{F}$ and $\ell \in \mathcal{S}^{\mathcal{L}}$. Additionally, assume that $v_x^{n,k}(h, M_1, \tau, \ell)$ is constant in $\tau < \overline{\tau}_x(M_1)$ for $h \in \mathcal{S}^M$, $\ell \in \mathcal{S}^{\mathcal{L}}$, and $n = 1, ..., u$. Thus, when the patient chooses to do nothing for $\tau < \overline{\tau}_x(m)$:

$$\begin{aligned} v_{x,DN}^{u+1,k}(h, M_1, 0, \ell) &= r_{DN}^k(h) + \lambda \sum_{h' \in S^H} \mathcal{H}^k(h'|h) \left\{ \sum_{\ell' \in S^{\mathcal{L}}} \mathcal{L}^k(\ell'|M_1) v_x^{u,k}(h', M_1, \overline{\tau}_x(M_1) - 1, \ell') \right\} \\ &= r_{DN}^k(h) + \lambda \sum_{h' \in S^H} \mathcal{H}^k(h'|h) \left\{ \sum_{\ell' \in S^{\mathcal{L}}} \mathcal{L}^k(\ell'|M_1) v_x^{u,k}(h', M_1, \tau - 1, \ell') \right\} \\ &= v_{x,DN}^{u,k}(h, M_1, \tau, \ell), \end{aligned} \tag{3.22}$$

where (3.22) follows from the induction hypothesis. Note that $v_{x,U}^{u+1,k}(h,m,\tau,\ell)$ and $v_{x,T}^{u+1,k}(h,m,\tau,\ell)$ are constant in $\tau < \overline{\tau}_x(M_1)$ by definition, hence, the result directly follows.

(b). Note that $v_{x,U}^{k}(h,m,\tau,\ell)$ and $v_{x,DN}^{k}(h,m,\tau,\ell)$ do not depend on ℓ, and $v_{x,T}^{k}(h,m,\tau,\ell)$ is decreasing in ℓ due to Assumption 3.4.4.

(c). By induction on the steps of value iteration algorithm, assume that $v_x^{n,k}(h,m,\tau,\ell)$ is increasing in m for $h \in \mathcal{S}^H$, $\tau < \overline{\tau}_x(m)$, $\ell \in \mathcal{S}^{\mathcal{L}}$ and $n = 1, ..., u$. Consider $v_{x,DN}^{u+1,T}(h,m,\tau,\ell)$ for $1 \leq \tau < \overline{\tau}_x(m)$:

$$
\begin{aligned}
v_{x,DN}^{u+1,k}(h,m,\tau,\ell) &= r_{DN}^{k}(h) + \lambda \sum_{h' \in S^H} \mathcal{H}^k(h'|h) \left\{ \sum_{\ell' \in S^{\mathcal{L}}} \mathcal{L}^k(\ell'|m) v_x^{u,k}(h',m,\tau-1,\ell') \right\} \\
&\leq r_{DN}^{k}(h) + \lambda \sum_{h' \in S^H} \mathcal{H}^k(h'|h) \left\{ \sum_{\ell' \in S^{\mathcal{L}}} \mathcal{L}^k(\ell'|m+1) v_x^{u,k}(h',m,\tau-1,\ell') \right\} \qquad (3.23) \\
&\leq r_{DN}^{k}(h) + \lambda \sum_{h' \in S^H} \mathcal{H}^k(h'|h) \left\{ \sum_{\ell' \in S^{\mathcal{L}}} \mathcal{L}^k(\ell'|m+1) v_x^{u,k}(h',m+1,\tau-1,\ell') \right\} \qquad (3.24) \\
&= v_{x,DN}^{u+1,k}(h,m+1,\tau,\ell).
\end{aligned}
$$

Inequality (3.23) follows by Assumption 3.4.5, Proposition 3.4.6 part (b) and Lemma 4.7.2 in [94]. Finally, inequality (3.24) follows by the induction. Note that by definition, $v_{x,U}^{u+1,k}(h,m,\tau,\ell)$, $v_{x,U}^{u+1,k}(h,m,\tau,\ell)$ and $v_{x,DN}^{u+1,k}(h,m,0,\ell)$ do not depend on m, hence, the result follows.

(d). By induction on the steps of value iteration algorithm, assume that $v_x^{n,k}(h,m,\tau,\ell)$ is increasing in $\tau < \overline{\tau}_x(m)$ for $h \in \mathcal{S}^H$, $m \in \mathcal{S}^M$, $\ell \in \mathcal{S}^{\mathcal{L}}$ and $n = 1, ..., u$.

$$
\begin{aligned}
v_{x,DN}^{u+1,k}(h,m,0,\ell) &= r_{DN}^k(h) + \lambda \sum_{h' \in S^H} \mathcal{H}^k(h'|h) \left\{ \sum_{\ell' \in S^{\mathcal{L}}} \mathcal{L}^k(\ell'|M_1) v_x^{u,k}(h', M_1, \overline{\tau}_x(M_1) - 1, \ell') \right\} \\
&\leq r_{DN}^k(h) + \lambda \sum_{h' \in S^H} \mathcal{H}^k(h'|h) \left\{ \sum_{\ell' \in S^{\mathcal{L}}} \mathcal{L}^k(\ell'|m) v_x^{u,k}(h', M_1, \overline{\tau}_x(M_1) - 1, \ell') \right\} \qquad (3.25) \\
&= r_{DN}^k(h) + \lambda \sum_{h' \in S^H} \mathcal{H}^k(h'|h) \left\{ \sum_{\ell' \in S^{\mathcal{L}}} \mathcal{L}^k(\ell'|m) v_x^{u,k}(h', M_1, 0, \ell') \right\} \qquad (3.26) \\
&\leq r_{DN}^k(h) + \lambda \sum_{h' \in S^H} \mathcal{H}^k(h'|h) \left\{ \sum_{\ell' \in S^{\mathcal{L}}} \mathcal{L}^k(\ell'|m) v_x^{u,k}(h', m, 0, \ell') \right\} \qquad (3.27) \\
&= v_{x,DN}^{u+1,k}(h,m,1,\ell),
\end{aligned}
$$

where the inequality (3.25) follows by Assumption 3.4.5, Proposition 3.4.6 part (b) and Lemma 4.7.2 in [94]. Inequality (3.26) follows by Proposition 3.4.6 part (a) and inequality (3.27) follows by Proposition 3.4.6 part(b). Now, when $0 < \tau < \overline{\tau}_x(m)$:

$$
\begin{aligned}
v_{x,DN}^{u+1,k}(h,m,\tau,\ell) &= r_{DN}^k(h) + \lambda \sum_{h' \in S^H} \mathcal{H}^k(h'|h) \left\{ \sum_{\ell' \in S^{\mathcal{L}}} \mathcal{L}^k(\ell'|m) v_x^{u,k}(h', m, \tau - 1, \ell') \right\} \\
&\leq r_{DN}^k(h) + \lambda \sum_{h' \in S^H} \mathcal{H}^k(h'|h) \left\{ \sum_{\ell' \in S^{\mathcal{L}}} \mathcal{L}^k(\ell'|m) v_x^{u,k}(h', m, \tau, \ell') \right\} \qquad (3.28) \\
&= v_{x,DN}^{u+1,k}(h,m,\tau+1,\ell),
\end{aligned}
$$

where inequality (3.28) follows by induction. Note that when $\tau \geq \overline{\tau}_x(m)$ the result directly follows by the definition of impediment values in model formulation. Finally, $v_{x,a}^{u+1}(h,m,\tau,\ell)$ is independent of τ for $a \in \{U, T\}$ which concludes the proof.

□

In Proposition 3.4.7, we show the existence of a *liver-based* control limit optimal policy [6], based on the liver quality, the time remaining until the next updating requirement, and patient's last reported health status for all patient types. Control-limit policies are easier to implement and provide insights to devise computationally enhanced solution methods to USDP[μ]. In Proposition 3.4.7, we establish the existence of a critical liver quality ℓ^* where it is optimal to accept the liver offer of quality ℓ for all $\ell \leq \ell^*$, and do nothing or transplant otherwise. In (b) and (c), we show the existence of such optimal policies based on the remaining time until the next required update and patient's last reported health status, respectively.

Proposition 3.4.7. *Given an updating scheme x, for all patient types $1 \leq k \leq K$ and $\tau < \overline{\tau}_x(m)$ where $m \in \mathcal{S}^M$:*

(a). *Under Assumption 3.4.4, for a given h, m, τ, there exists a liver quality ℓ^* such that transplanting is optimal for $\ell \leq \ell^*$ and doing nothing or updating is optimal otherwise.*

(b). *Under Assumptions 3.4.4 to 3.4.5, for a given h, m, ℓ there exists a time remaining until next required update τ^* such that doing nothing is optimal for $\tau \geq \tau^*$ and transplanting or updating is optimal otherwise.*

(c). *Under Assumptions 3.4.4 to 3.4.5, for a given h, τ, ℓ there exists a MELD score m^* such that doing nothing is optimal for $m \leq m^*$ and transplanting or updating is optimal otherwise.*

Proof. Given an updating scheme x, for all patient types $1 \leq k \leq K$ and $\tau < \overline{\tau}_x(m)$ where $m \in \mathcal{S}^M$:

(a). The result follows by Proposition (3.4.6) part (b), since $v_x^{k,DN}(h, \ell)$ and $v_x^{k,U}(h)$ are constant in ℓ and $v_x^{k,T}(h, \ell)$ is decreasing in ℓ.

(b). The result follows by Proposition (3.4.6) part (c), since $v_x^{k,T}(h, \ell)$ and $v_x^{k,U}(h)$ are constant in ℓ and $v_x^{k,DN}(h, \ell)$ is increasing in τ.

(c). The result follows by Proposition (3.4.6) part (d), since $v_x^{k,T}(h, \ell)$ and $v_x^{k,U}(h)$ are constant in ℓ and $v_x^{k,DN}(h, \ell)$ is increasing in m.

□

3.5 MATHEMATICAL FORMULATION OF USDP[μ]

In Section 3.3, we defined μ as the minimum desired level of reduction in the data processing burden, i.e., the minimum desired increase in system efficiency. We further noted that the objective of USDP[μ] is to determine an updating scheme that minimizes the expected system inequity while preserving a minimum desired level of reduction in the data processing burden μ. Considering the design decision variables x as defined in Section 3.3, the mathematical formulation of USDP[μ] is as follows.

$$\text{USDP}[\mu] \qquad \eta(\mu) = \min \quad \mathbb{E}_{\xi}[Q(x, \xi^k)] \tag{3.29}$$

$$\text{s.t.} \quad x_{ij} + \sum_{j' \in \mathcal{F}, j' > j} x_{(i+1)j'} \leq 1, \qquad i \in S^{\mathcal{M}} \setminus \{M_{\max}\}, j \in \mathcal{F} \setminus \{F_{\max}\}, \tag{3.30}$$

$$\sum_{j \in \mathcal{F}} x_{ij} = 1, \qquad i \in \mathcal{S}^M, \tag{3.31}$$

$$\mathbb{E}_{\xi}[G(x, \xi^k)] \geq \mu, \tag{3.32}$$

$$x_{ij} \in \{0, 1\}. \tag{3.33}$$

The objective of the model (3.29) is to minimize the expected system inequity. Constraints (3.30) define monotone updating schemes that require patients to update no less frequently as they get sicker. Constraints (3.31) assign an updating frequency for each MELD score i, and constraint (3.32) ensures a reduction in the data processing burden no less than μ.

3.5.1 A Decomposition-based Algorithm for USDP[μ]

In Section 3.4.3, we explored the properties of patients' optimal policies and value functions under different updating schemes. These properties provide deeper insights into the expected recourse function value of USDP[μ] that enable devise of a computationally decomposition-based solution method.

Definition A monotonic updating scheme x is *frequency-wise dominated* by updating scheme $\hat{x}$ if

$$\sum_{i \in \mathcal{S}^{\mathcal{M}}} \left(\sum_{j \geq \overline{\tau}_{\hat{x}}(i)} x_{ij} - \sum_{j < \overline{\tau}_{\hat{x}}(i)} x_{ij} \right) = |\mathcal{S}^M|,$$

and is denoted by $\hat{x} \preccurlyeq x$.

Intuitively, for any two monotonic updating schemes x and $\hat{x}$ where $x \preccurlyeq \hat{x}$, the frequency-wise dominated updating scheme $\hat{x}$ increases patients' gaming ability, but reduces the data processing cost. Proposition 3.5.1 provides insights on these intuitive facts where we compare two monotonic updating schemes x and $\hat{x}$ when $x \preccurlyeq \hat{x}$. This proposition indicates that (i) a frequency-wise dominated updating scheme is preferable from a patient perspective by increasing patients' expected life, and (ii) focusing on societal perspective, the dominant updating scheme yields a higher reduction in data processing burden, compared to a dominated updating scheme.

Proposition 3.5.1. *Under Assumptions 3.4.4 and 3.4.5, for any monotonic updating scheme $\hat{x}$ that is frequency-wise dominated by x,*

(1). $v_x^k(h, m, \tau, \ell) \leq v_{\hat{x}}^k(h, m, \tau, \ell)$ *for all* h, m, τ, ℓ *and* $1 \leq k \leq K$,

(2). $U^{\hat{\pi}_x^k}(h, h, \overline{\tau}_x(h) - 1, L + 1) \geq U^{\hat{\pi}_{\hat{x}}^k}(h, h, \overline{\tau}_{\hat{x}}(h) - 1, L + 1)$ *for all* h *and* $1 \leq k \leq K$.

Proof. (1). The proof for the case when $\tau \geq \overline{\tau}_{\hat{x}}(m)$ for $m \in \mathcal{S}^M$, $h \in \mathcal{S}^H$ and $\ell \in \mathcal{S}^{\mathcal{L}}$ is trivial by definition. Thus, we proceed by induction on the steps of the value iteration algorithm for the case when $\tau < \overline{\tau}_{\hat{x}}(m)$. Let $v_x^{n,k}(h, m, \tau, \ell)$ and $v_{\hat{x}}^{n,k}(h, m, \tau, \ell)$ be the value functions under updating schemes x and $\hat{x}$ in the nth step of the value iteration algorithm. Without loss of generality, let $v_x^{0,k}(h, m, \tau, \ell) = v_{\hat{x}}^{0,k}(h, m, \tau, \ell) = 0$ for all h, m, τ and ℓ and assume that $v_x^{n,k}(h, m, \tau, \ell) \leq v_{\hat{x}}^{n,k}(h, m, \tau, \ell)$ for $n = 1, ..., u$. Note that $v_{x,T}^{u+1,k}(h, \ell) \leq v_{\hat{x},T}^{u+1,k}(h, \ell)$ since they are independent of τ. Now, for any $h \in \mathcal{S}^H$, $m \in \mathcal{S}^M$, $\tau < \overline{\tau}_{\hat{x}}(m)$ and $\ell \in \mathcal{S}^{\mathcal{L}}$, we have:

$$
\begin{aligned}
v^{u+1,k}_{\hat{x},U}(h,m,\tau,\ell) &= r^k_U(h) + \lambda \sum_{h' \in S^H} \mathcal{H}^k(h'|h) \left\{ \sum_{\ell' \in S^{\mathcal{L}}} \mathcal{L}^k(\ell'|m) v^{u,k}_{\hat{x}}(h', h, \overline{\tau}_{\hat{x}}(h) - 1, \ell') \right\} \\
&\geq r^k_U(h) + \lambda \sum_{h' \in S^H} \mathcal{H}^k(h'|h) \left\{ \sum_{\ell' \in S^{\mathcal{L}}} \mathcal{L}^k(\ell'|m) v^{u,k}_{\hat{x}}(h', h, \overline{\tau}_{x}(h) - 1, \ell') \right\} \quad (3.34) \\
&\geq \overline{r}^k_{x,U}(h,m,\tau,\ell) + \lambda \sum_{h' \in S^H} \mathcal{H}^k(h'|h) \left\{ \sum_{\ell' \in S^{\mathcal{L}}} \mathcal{L}^k(\ell'|m) v^{u,k}_{x}(h', h, \overline{\tau}_{x}(h) - 1, \ell') \right\} \quad (3.35) \\
&= v^{u+1,k}_{x,U}(h,m,\tau,\ell), \quad (3.36)
\end{aligned}
$$

where inequality (3.34) follows from by Proposition 3.4.6 part (d) since $v^k_{\hat{x}}(h,m,\tau,\ell)$ is increasing in $\tau < \overline{\tau}_{\hat{x}}(m)$, and inequality (3.35) follows by the induction assumption. Next, for any $h \in \mathcal{S}^H$, $m \in \mathcal{S}^M$, $\tau < \overline{\tau}_{\hat{x}}(m)$ and $\ell \in \mathcal{S}^{\mathcal{L}}$, we have:

$$
\begin{aligned}
v^{u+1,k}_{\hat{x},DN}(h,m,\tau,\ell) &= r^k_{DN}(h) + \lambda \sum_{h' \in S^H} \mathcal{H}^k(h'|h) \left\{ \sum_{\ell' \in S^{\mathcal{L}}} \mathcal{L}^k(\ell'|m) v^{u,k}_{\hat{x}}(h', m, \tau - 1, \ell') \right\} \\
&\geq \overline{r}^k_{x,DN}(h,m,\tau,\ell) + \lambda \sum_{h' \in S^H} \mathcal{H}^k(h'|h) \left\{ \sum_{\ell' \in S^{\mathcal{L}}} \mathcal{L}^k(\ell'|m) v^{u,k}_{x}(h', m, \tau - 1, \ell') \right\} \quad (3.37) \\
&= v^{u+1,k}_{x,DN}(h,m,\tau,\ell), \quad (3.38)
\end{aligned}
$$

where inequality (3.37) follows by the induction assumption. Finally, for any $h \in \mathcal{S}^H$, $m \in \mathcal{S}^M$, $\tau = 0$ and $\ell \in \mathcal{S}^{\mathcal{L}}$, we have:

$$
\begin{aligned}
v^{u+1,k}_{\hat{x},DN}(h,m,0,\ell) &= r^k_{DN}(h) + \lambda \sum_{h' \in S^H} \mathcal{H}^k(h'|h) \left\{ \sum_{\ell' \in S^{\mathcal{L}}} \mathcal{L}^k(\ell'|M_1) v^{u,k}_{\hat{x}}(h', M_1, \overline{\tau}_{\hat{x}}(M_1) - 1, \ell') \right\} \\
&\geq r^k_{DN}(h) + \lambda \sum_{h' \in S^H} \mathcal{H}^k(h'|h) \left\{ \sum_{\ell' \in S^{\mathcal{L}}} \mathcal{L}^k(\ell'|M_1) v^{u,k}_{\hat{x}}(h', M_1, \overline{\tau}_{x}(M_1) - 1, \ell') \right\} \quad (3.39) \\
&\geq \overline{r}^k_{x,DN}(h,m,0,\ell) + \lambda \sum_{h' \in S^H} \mathcal{H}^k(h'|h) \left\{ \sum_{\ell' \in S^{\mathcal{L}}} \mathcal{L}^k(\ell'|M_1) v^{u,k}_{x}(h', M_1, \overline{\tau}_{x}(M_1) - 1, \ell') \right\} \quad (3.40) \\
&= v^{u+1,k}_{x,DN}(h,m,0,\ell), \quad (3.41)
\end{aligned}
$$

where inequality Proposition 3.4.6 part (d) since $v^k_{\hat{x}}(h, m, \tau, \ell)$ is increasing in $\tau < \overline{\tau}_{\hat{x}}(m)$, and inequality (3.40) follows by the induction assumption. Considering,

$$v^{u+1,k}_{\hat{x}}(h, m, \tau, \ell) = \begin{cases} \max\{v^{u+1,k}_{\hat{x},T}(h, \ell), v^{u+1,k}_{\hat{x},DN}(h, m, \tau), v^{u+1,k}_{\hat{x},U}(h)\} & \forall h, m, \tau, \ell < L+1, \\ \max\{v^{u+1,k}_{\hat{x},DN}(h, m, \tau), v^{u+1,k}_{\hat{x},U}(h)\} & \forall h, m, \tau, \ell = L+1, \end{cases}$$

and

$$v^{u+1,k}_{x}(h, m, \tau, \ell) = \begin{cases} \max\{v^{u+1,k}_{x,T}(h, \ell), v^{u+1,k}_{x,DN}(h, m, \tau), v^{u+1,k}_{x,U}(h)\} & \forall h, m, \tau, \ell < L+1, \\ \max\{v^{u+1,k}_{x,DN}(h, m, \tau), v^{u+1,k}_{x,U}(h)\} & \forall h, m, \tau, \ell = L+1, \end{cases}$$

and inequalities (3.36), (3.38) and (3.41), we conclude that $v^{u+1}_{\hat{x}}(h, m, \tau, \ell) \geq v^{u+1}_{x}(h, m, \tau, \ell)$ for all h, m, τ, ℓ.

(2). Let $\hat{\pi}^k_x$ and $\hat{\pi}^k_{\hat{x}}$ be the optimal policies of patient type k for given updating schemes x and $\hat{x}$, respectively. Consider a Markov reward process induced by these optimal policies for each updating scheme. In this reward process, we consider a unit cost when the patient updates and no cost for doing nothing or transplant decisions. Hence, for a given discount factor $\lambda \in [0, 1)$, $U^{\hat{\pi}^k_x}(h, h, \tau_x, \ell)$ and $U^{\hat{\pi}^k_{\hat{x}}}(h, h, \tau_{\hat{x}}, \ell)$ are the total expected discounted cost of the associated reward process for a patient with health state h and no liver offer at the time of listing under updating schemes x and $\hat{x}$, respectively. The state of the process under each updating scheme is denoted by (h, m, τ_x, ℓ) and $(h, m, \tau_{\hat{x}}, \ell)$, where $\tau_x \leq \tau_{\hat{x}}$ for all h, m and ℓ by definition. Note that: (i) $v^k_{\hat{x},U}(h, m, \tau, \ell)$ and $v^k_{\hat{x},T}(h, m, \tau, \ell)$ are independent of τ for $\tau < \overline{\tau}_{\hat{x}}$, and (ii) $v^k_{\hat{x},T}(h, m, \tau, \ell)$ is increasing in τ for $\tau < \overline{\tau}_{\hat{x}}$ by Proposition 3.4.7 part (d). This implies that if updating is an optimal decision in (h, m, τ, ℓ) for $\tau < \overline{\tau}_{\hat{x}}$, then it is also optimal for all (h, m, τ', ℓ) where $\tau' \leq \tau$.

□

This USDP[μ] model is a variant of SPAR [8, 30]. Motivated by this general modeling framework, our proposed model is also a two-stage stochastic programming model where: (i) design decisions are made in the face of uncertainty in the patient population, and (ii) patients' long-run decisions are modeled as MDPs. Moreover, our proposed model can also be formulated as a special case of a discounted zero-sum stochastic game [48, 61].

However, a stochastic game model of this problem requires a state-space description for every feasible updating scheme which makes the problem computationally intractable due to the exponential number of such designs. We refer the reader to [8] for a more comprehensive discussion of the modeling frameworks in similar design problems.

The computational efficiency of traditional L-shaped method [114] can be improved, depending on the special structure of some two-stage stochastic programs. Solving USDP[μ] involves determining the second-stage recourse function value as a solution to a number of patients' MDPs. However, solving an extended linear programming formulation of the recourse model is not usually computationally appealing since the MDPs can be efficiently solved using other more enhanced techniques. Hence, we resort an alterative decomposition algorithm that lend itself to the underling special structure of USDP[μ].

Our decomposition algorithm to solve USDP[μ], is motivated by [72] and [73] on two-stage stochastic programming problems with pure first-stage binary decision variables. Laporte and Louveaux [72] present an algorithm to solve this class of problems by deriving a set of supporting hyperplanes, each associated with a feasible first-stage binary solution. The set of supporting hyperplanes, as a set of optimality cuts, are used to approximate the second-stage value function. Their assumptions in deriving these hyperplanes are that there exists a valid lower bound for the second-stage objective function, and that the second-stage problem is well-defined for each feasible first-stage solution. The latter assumption is referred as the relatively complete recourse assumption in the stochastic programming literature [18].

We refer to the algorithm in [72] as the L2 algorithm. However, before we present the details of our proposed algorithm, we need to mention an important issue with the original optimality cuts used in the L2 algorithm. The issue is associated with the weakness property of these cuts in the sense that there is little chance that they will result in a non-trivial lower bound to any feasible first-stage solution other than the one for which the cut was made. Consequently, the algorithm often involves a total enumeration of the first-stage solution space in most computational experiments. However, for some special problem instances, it is possible to improve these cuts using the extra information gleaned from the structural properties of the second-stage objective function. Let $\mathcal{Q}(x)$ be the expected second-stage objective function of USDP at a feasible first-stage solution, denoted by $\mathcal{Q}(x) = \mathbb{E}_{\tilde{\xi}}[Q(x, \tilde{\xi})]$.

Proposition 3.5.2. *For any feasible first-stage solution $\hat{x}$ to USDP[μ],*

$$\mathcal{Q}(\hat{x})\left(\sum_{i\in\mathcal{S}^{\mathcal{M}}}\left\{\sum_{j\geq\overline{\tau}_{\hat{x}}(i)}x_{ij}-\sum_{j<\overline{\tau}_{\hat{x}}(i)}x_{ij}\right\}-(|\mathcal{S}^{M}|-1)\right) \tag{3.42}$$

is a supporting hyperplane of $\mathcal{Q}(x)$ at $\hat{x}$.

Proof. For a given updating scheme $\hat{x}$, any binary decision vector x satisfying (3.30), (3.31), and

$$\sum_{i\in\mathcal{S}^{\mathcal{M}}}\left(\sum_{j\geq\overline{\tau}_{\hat{x}}(i)}x_{ij}-\sum_{j<\overline{\tau}_{\hat{x}}(i)}x_{ij}\right)=|\mathcal{S}^{M}| \tag{3.43}$$

denotes an updating scheme $\hat{x}\preccurlyeq x$ with $x=\hat{x}$ if and only if $x_{ij}=\hat{x}_{ij}$ for all $i\in\mathcal{S}^{M}$, $j\in\mathcal{F}$. Thus, Proposition 3.5.1 part (a) indicates that:

$$\begin{aligned}\omega^{k}_{\hat{x}}(h)&=\frac{v^{k}_{\hat{x}}(h,h,\overline{\tau}_{\hat{x}}(h)-1,L+1)-v^{k}_{x_w}(h,h,\overline{\tau}_{x_w}(h)-1,L+1)}{v^{k}_{x_w}(h,h,\overline{\tau}_{x_w}(h)-1,L+1)}\times 100\\&\leq\frac{v^{k}_{x}(h,h,\overline{\tau}_{x}(h)-1,L+1)-v^{k}_{x_w}(h,h,\overline{\tau}_{x_w}(h)-1,L+1)}{v^{k}_{x_w}(h,h,\overline{\tau}_{x_w}(h)-1,L+1)}\times 100\\&=\omega^{k}_{x}(h).\end{aligned}$$

Hence, $\mathcal{Q}(\hat{x})\leq\mathcal{Q}(x)$ by (3.20). Finally, a similar argument shows that $\mathcal{Q}(x)\geq 0$ for any updating scheme x such that $x_w\preccurlyeq x$ which completes the proof. □

By Proposition 3.5.2, we derive a supporting hyperplane for $\mathcal{Q}(x)$ at each feasible first-stage solution vector x. These supporting hyperplanes are induced by patients' optimal policies through the MDP problems under scenarios $k=1,...,K$. In addition, there are finitely many first-stage solutions that each determines an updating scheme. Hence, approximating the second-stage objective function using these hyperplanes guarantees convergence to an optimal first-stage solution for USDP.

We refer to a binary first-stage decision vector x as a "relaxed monotonic solution" (RMS), if it satisfies constraints (3.30) and (3.31). Additionally, an RMS solution is called "feasible monotonic solution" (FMS) at a desired level of efficiency μ, if it also satisfies (3.32). Hence, the set of all feasible first-stage solutions of USDP consists of all possible FMS solutions, denoted by $X(\mu)$. Next, we using the structural properties of the second-stage MDPs, we introduce a set of valid inequalities for $X(\mu)$.

Proposition 3.5.3. *For a given RMS, $\hat{x} \notin X(\mu)$, the constraint*

$$\sum_{i \in \mathcal{S}^M} \sum_{j \leq \overline{\tau}_{\hat{x}}(i)} x_{ij} \leq |\mathcal{S}^M| - 1 \tag{3.44}$$

is a valid inequality for $X(\mu)$.

Proof. Given an RMS denoted by x that violates (3.44), Proposition 3.5.1 part (b), and equations (3.17), (3.21) indicate that $\mathbb{E}_\xi[G(x,\xi)] \leq \mathbb{E}_\xi[G(\hat{x},\xi)]$, that shows $x \notin X(\mu)$. □

Therefore, if (x^*, θ^*) is an optimal solution to the following MIP, referred as the "revised formulation" (RF):

$$[RF] \qquad \eta(\mu) = \min \quad \theta \tag{3.45}$$

$$\text{s.t.} \quad \theta \geq \mathcal{Q}(\hat{x}) \left(\sum_{i \in \mathcal{S}^{\mathcal{M}}} \left\{ \sum_{j \geq \overline{\tau}_{\hat{x}}(i)} x_{ij} - \sum_{j < \overline{\tau}_{\hat{x}}(i)} x_{ij} \right\} - (|\mathcal{S}^M| - 1) \right), \quad \forall \hat{x} \in X(\mu), \tag{3.46}$$

$$x \in X(\mu), \quad \theta \in R, \tag{3.47}$$

then x^* is an optimal first-stage solution of USDP. The objective function in RF involves θ, that approximates the second-stage objective function value.

Based on Proposition 3.5.2, each constraint (3.46) is a supporting hyperplane for the second-stage objective function at $\hat{x} \in X(\mu)$, providing a lower bound for θ. These constraints are regarded as strengthened optimality cuts in our application. The set of all FMS solutions $X(\mu)$, is exponential in size and impractical to enumerate explicitly. Additionally, constraints (3.32) can further complicate identifying those solutions in $X(\mu)$. In order to overcome the computational difficulty of determining $X(\mu)$, we propose a decomposition method that works on a so called, "master problem" (MP). The MP initially relaxes RF in two ways by: (i) relaxing the efficiency constraints (3.32) to identify $X(\mu)$; and (ii) the

optimality constraints (3.46). We modify MP by dynamically introducing a set of feasibility and optimality cuts at each RMS solution. Our proposed initial master problem is as follows:

$$
\begin{aligned}
[MP] \qquad & \eta(\mu) = \min \quad \theta \\
& \text{s.t.} \qquad (3.30) - (3.31), \\
& \qquad\qquad x_{ij} \in \{0,1\}, && i \in \mathcal{S}^M, j \in \mathcal{F}, \\
& \qquad\qquad \theta \in R.
\end{aligned}
$$

Next, we present an algorithm to gradually modify the MP. At each iteration of the algorithm, a master module solves the MP and attains an optimal RMS, x^ν, as well as an approximate value of the second-stage objective function θ^ν, with ν as the iteration number. Next, these optimal solutions are passed to a worker module. The worker is comprised of three submodules: (1) a submodule to solve the proposed patient' MDPs problems under each scenario for the given RMS solution; (2) a submodule to derive a feasibility cut in case the efficiency requirement is violated; (3) a submodule to determine an improved optimality cut if it is violated. The algorithm proceeds by iteratively introducing the violated feasibility and optimality cuts into the MP until no feasibility and optimality requirements are violated, terminating with an optimal solution for USDP. The general formulation of the MP problem at iteration ν to determine $\eta(\mu)$ with $\mathcal{D}$ and $\mathcal{Y}(\mu)$ as the sets of optimality and feasibility cuts is:

$$
\begin{aligned}
[MP^\nu] \qquad & \min \quad \theta \\
\text{s.t.} \quad & (3.30), (3.31) \\
& \theta \geq \mathcal{Q}(x^d) \left(\sum_{i \in \mathcal{S}^{\mathcal{M}}} \left\{ \sum_{j \geq \overline{\tau}_{x^d}(i)} x_{ij} - \sum_{j < \overline{\tau}_{x^d}(i)} x_{ij} \right\} - (|\mathcal{S}^M| - 1) \right), && d \in \mathcal{D}, \\
& \sum_{i \in \mathcal{S}^M} \sum_{j \leq \overline{\tau}_{x^y}(i)} x_{ij} \leq |\mathcal{S}^M| - 1, && y \in \mathcal{Y}(\nu), \\
& x_{ij} \in \{0,1\}, && i \in \mathcal{S}^M, j \in \mathcal{F}, \\
& \theta \in R,
\end{aligned}
$$

Algorithm 1. A Decomposition Algorithm to Solve USDP[μ]

Step 0. (Initialization) Set the iteration count $\nu := 0$.

Step 1. Set $\nu := \nu + 1$ and solve MP. Let (x^ν, θ^ν) be an optimal solution.

Step 2. Solve the induced MDP model (3.4) for each scenario $k = 1, ..., K$ for x^ν, and determine patient's optimal policy $\pi^k_{x^\nu}$ and life expectancy $v^k_{x^\nu}$.

Step 3. (Feasibility Check) Calculate $\mathbb{E}_{\tilde{\xi}}[G(x^\nu, \tilde{\xi})]$ using $\pi^k_{x^\nu}$, $k = 1, ..., K$ by (3.21). If $\mathbb{E}_{\tilde{\xi}}[(x^\nu, \tilde{\xi})] \leq \mu - \epsilon$ for some $\epsilon > 0$, add the constraint

$$\sum_{i \in \mathcal{S}^M} \sum_{j \leq \overline{\tau}_{x^\nu}(i)} x_{ij} \leq |\mathcal{S}^M| - 1$$

to $\mathcal{Y}(\mu)$ and return to Step 1.

Step 4. (Optimality Check) Calculate $\mathcal{Q}(x^\nu) = \mathbb{E}_{\tilde{\xi}}[Q(x^\nu, \tilde{\xi})]$ using $v^k_{x^\nu}$, for $k = 1, ..., K$ by (3.20). If $\theta^\nu \geq \mathcal{Q}(x^\nu)$, then (x^ν, θ^ν) is the optimal solution for USDP and terminate, otherwise, add constraint

$$\theta \geq \mathcal{Q}(x^\nu) \left(\sum_{i \in \mathcal{S}^{\mathcal{M}}} \left\{ \sum_{j \geq \overline{\tau}_{x^\nu}(i)} x_{ij} - \sum_{j < \overline{\tau}_{x^\nu}(i)} x_{ij} \right\} - (|\mathcal{S}^M| - 1) \right)$$

to $\mathcal{D}$ and return to Step 1.

3.6 APPROXIMATING THE EFFICIENT FRONTIER OF UPDATING SCHEMES

As noted in Section 3.5, the USDP[μ] model is to determine an updating scheme that minimizes the expected system inequity subject to a given desired level of system efficiency μ. We proposed a decomposition-based algorithm that utilizes the structural properties of this parametric model in order to determine the optimal updating scheme.

In this section, we seek to approximate the efficient frontier of Pareto optimal solutions. To do so, we extend the previously discussed algorithm in Section 3.5.1, where several instances of the parametric model, each associated with a specific desired level of improvement in system efficiency, are solved repeatedly in an iterative process.

Let $\mu_{\min}$ and $\mu_{\max}$ denote the minimum and maximum attainable reduction in data processing burden, i.e., improvement in system efficiency, through revising the updating requirements, respectively. Recall that $\mu_{\min}$ is associated with an updating scheme that requires patients to report their health status to the most frequent extreme, i.e, at each decision epoch in patient's decision process. Conversely, an updating scheme that allows patients to update their health status with the least frequency $F_{\max}$, improves the system efficiency to its maximum possible value $\mu_{\max}$. Recall that we represented the minimum attainable inequity value for a given efficiency measure μ with $\eta(\mu)$. We approximate the actual efficient frontier by finding $\eta(\mu)$, where $\mu_{\min} \leq \mu \leq \mu_{\max}$ and μ are gradually increased starting at $\mu = \mu_{\min}$, and enumerating the nondominated solutions on our path. Hence, we employ the ϵ-constraint method [26, 42, 44] within an iterative procedure to find the Pareto-optimal set of the system. The following lemma indicates that the minimum attainable system inequity is not decreased as the efficiency measure μ increases.

Lemma 3.6.1. *$\eta(\mu) \leq \eta(\hat{\mu})$ for $\mu \leq \hat{\mu}$.*

Proof. For $\mu \leq \hat{\mu}$, constraint (3.32) implies $X(\hat{\mu}) \subseteq X(\mu)$. □

As a result of Lemma 3.6.1, our proposed algorithm can be employed to approximate the actual efficient frontier of Pareto-optimal solutions to any level of accuracy, depending on the level of increase in the value of μ at each iteration. We consider a stationary level of increase in the value of μ in the iterative method, denote it by δ. As supported by our computational results in Section 3.7, the optimality and feasibility cuts obtained while finding each $\eta(\mu)$ provide additional information to ease the computational effort in the subsequent iterations, and substantially decreases the overall solution time. Hence, we extend Algorithm 1 to exploit the information obtained from optimality and feasibility cuts in earlier iterations in order to enhance the overall performance of the algorithm to determine the efficient frontier.

We denote the set of all updating schemes forming the efficient frontier of the Pareto-optimal solutions by $\mathcal{B}^{\delta}$. Similar to the notation in Section 3.5.1, $\mathcal{D}$ and $\mathcal{Y}$ are the sets of optimality and feasibility cuts at each iteration, respectively. The schematic representation of the iterative algorithm to determine the efficient frontier of the Pareto-optimal solution is as follows:

Algorithm 2. Algorithm to Approximate the Efficient Frontier of Pareto Optimal Solutions

Step 0: Set $n, u = 0$, $t = 1$, $\mathcal{D} = \mathcal{Y} = \emptyset$ and $\mu^{t-1} = \mu_{\min}$.

Step 1: Solve MP with (x^u, θ^u) be the optimal solution at iteration u.

Step 2: Solve the induced patient decision-making model (3.4) for each patient type $k = 1, ..., K$ given x^u. Determine patient's optimal policy $\pi^k_{x^u}$ and life expectancy $v^k_{x^u}$.

Step 3: Calculate $\mathbb{E}_{\tilde{\xi}}[G(x^u, \tilde{\xi})]$ using $\pi^k_{x^u}$, $k = 1, ..., K$ by (3.21). If $\mathbb{E}_{\tilde{\xi}}[(x^u, \tilde{\xi})] \leq \mu^t - \epsilon$ for some $\epsilon > 0$, then add constraint

$$\sum_{i \in \mathcal{S}^M} \sum_{j \leq \overline{\tau}_{x^u}(i)} x_{ij} \leq |\mathcal{S}^M| - 1$$

to $\mathcal{Y}$, return to Step 1.

Step 4. Calculate $\mathcal{Q}(x^u) = \mathbb{E}_{\tilde{\xi}}[Q(x^u, \tilde{\xi})]$ using $v^k_{x^u}$, for $k = 1, ..., K$ by (3.20). If $\theta^u \geq \mathcal{Q}(x^u)$, set $f^t_1 = \mathcal{Q}(x^u)$ and $f^t_2 = \mathbb{E}_{\tilde{\xi}}[G(x^u, \tilde{\xi})]$, go to Step 5. Otherwise, add constraint

$$\theta \geq \mathcal{Q}(x^u) \left(\sum_{i \in \mathcal{S}^{\mathcal{M}}} \left\{ \sum_{j \geq \overline{\tau}_{x^u}(i)} x_{ij} - \sum_{j < \overline{\tau}_{x^u}(i)} x_{ij} \right\} - (|\mathcal{S}^M| - 1) \right)$$

to $\mathcal{D}$ and return to Step 1.

Step 5. If $f^t_1 - \epsilon \geq f^{t-1}_1$ for $t \geq 1$ and some $\epsilon > 0$ where $f^0_1 = 0$, then set $n \leftarrow n + 1$, $(f^n_1, f^n_2) \leftarrow (f^{t-1}_1, f^{t-1}_2)$, and x^u to $\mathcal{B}^{\delta}$.

Step 6. If $f^n_2 + \delta \geq \mu_{\max}$, then terminate, otherwise, let $t \leftarrow t + 1$, $u \leftarrow u + 1$, and $\mu^t = f^n_2 + \delta$. Go to step 1.

3.7 NUMERICAL STUDY

In this section, we first describe parameter estimation and the design of our numerical experiments. We then focus on a patient's perspective and illustrate how a patient can benefit through information asymmetry in the current updating scheme of UNOS. Finally, we provide a detailed discussion on how to alleviate these concerns and present alternative updating schemes by changing the health reporting requirements. These results consist of the efficient frontier of Pareto-optimal updating schemes and those that improve upon the solutions proposed in [62].

3.7.1 Parameter Estimation and Implementation Details

As noted in Section 3.3, we capture the uncertainty in the cohort of patients on the liver waiting list by considering a finite set of patient types. We consider a patient's age, liver disease type, and gender as the main clinical and demographical factors to identify different patient types. Specifically, we focus on three age categories: 22 or younger, 23 to 59, and 60 or older, and five classes of ESLD [5, 6]. We use publicly available liver data provided by UNOS to calibrate the patient MDPs. Similar to [62], we fixed other patient characteristics to the most commonly occurring ones in the population. We define each period to be one week where the objective is to maximize the patient's total expected remaining lifetime. To differentiate between the times that a patient is not updating and when patient is going to a physician to perform clinical tests for updating, we set the rewards of doing nothing to 1.00 and updating to 0.95, respectively. For each patient type k, the patient-specific post-transplant rewards $R^k(h, \ell)$, $\forall h$, $\ell < L + 1$ are estimated using the post-transplant survival model of [101]. The data satisfy Assumption 3.4.4 and are described in Alagoz et al. [6].

The health transition probability matrices for each patient type $\mathcal{H}^k$, $k = 1, ..., K$, are estimated as in [6] and [62]. Due to the limitation in existing comprehensive data on the natural history of liver disease, Alagoz et al. [6] estimate health transition probability matrices for different disease groups using the natural history model (NHM) [3]. The NHM as an empirical stochastic model uses cubic spline function to estimate incomplete lab values

to calculate MELD scores. In this model patients are classified based on their disease group and location, where for each disease group, cubic splines are sampled at daily intervals for those at the hospital and in the intensive care unit (ICU), and at monthly intervals for patients at home to determine the complete history. These simulated lab values are employed to obtain MELD scores.

As noted earlier, we exclude Status 1A patients in our calculations and focus on MELD adult ESLD patients. Due to the sparsity of available data, Alagoz et al. [6] represent patient health by MELD scores aggregated in groups of two. However, we use a different aggregation scheme, similar to [62] to facilitate computational tractability. Içten [62] considers a MELD score aggregation scheme where the MELD scores (6-10) are aggregated into one group and the next eight MELD scores are aggregated into two groups of four (11-14 and 15-18). She further aggregates the remaining MELD scores as in [6] into groups of two.

We estimate $\mathcal{L}^k$ for $k = 1, ..., K$ similar to the liver classification in [6]. Alagoz et al. [6] consider 14 liver qualities as determined by the age, race, and gender of the donor [101]. We refer the reader to [6] for details of the liver quality assignment scheme. We consider a national liver waiting list in our experiments, hence, the liver arrival probability matrices in our model are not patient type dependent. Içten [62] used the following metric:

$$\epsilon = \max_{n,m} \left\{ \max\{0, \sum_{\ell=n}^{L+1} \mathcal{L}(\ell|m+1) - \mathcal{L}(\ell|m)\} \right\}, \tag{3.48}$$

for $n = 1, ..., L+1$ and $m = 1, ..., H-1$ to quantify the violation of Assumption 3.4.5. She reported a maximum violation of 0.0128 in only two rows of $\mathcal{L}$.

We consider an annual discount rate of 0.97 in the MDPs for all patient types. Furthermore, for each patient type, we solve the MDP model presented in Section 3.4 using Gauss-Seidel modified policy iteration [94]. As a common assumption in previous studies [4, 5, 6, 104], we assume that all other patients behave as they do now when determining an optimal accept/reject/update policy for each patient type. We used an Intel Xeon machine with 3.20 GHz CPU and 12 GB of RAM for our computational tests, and implemented the optimization algorithms using C and Cplex 12.4 Callable Library.

3.7.2 Evaluating Patients' Behavior

Our objective in this section is to evaluate the effect of information asymmetry due to patients' gaming ability in the liver allocation system from a patient perspective.

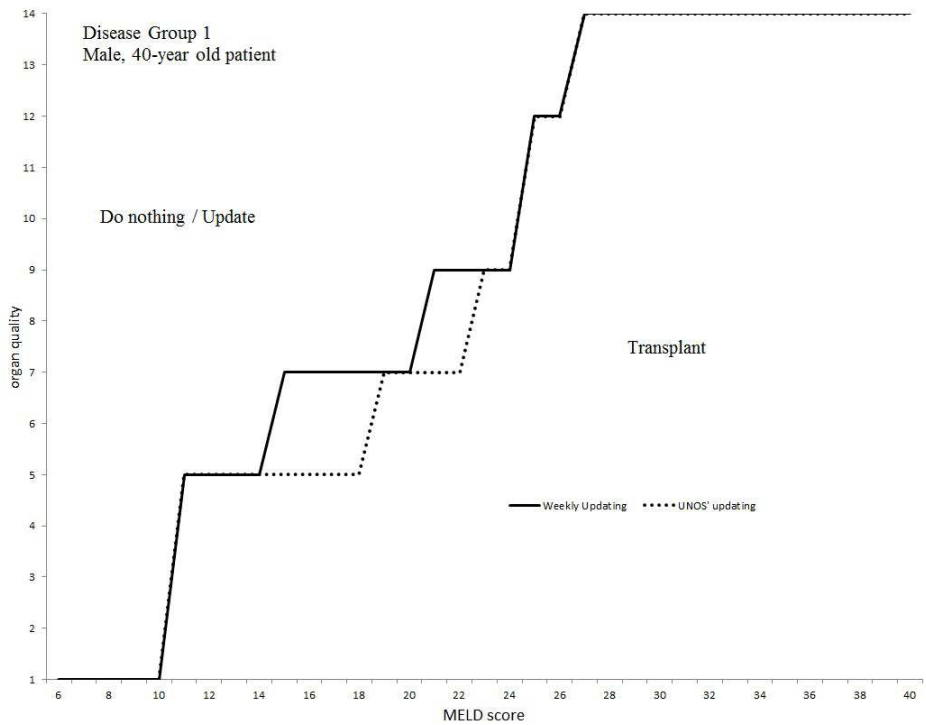

Figure 6: Optimal transplant behavior of patient under UNOS updating for $\tau = 0$ compared to a weekly updating scheme

Similar to [62], we first compare the optimal policy for a 40-year-old male patient from disease group 1 under the current updating scheme of UNOS to his optimal updating policy under the updating scheme with no information symmetry [6]. This disease group includes primary biliary cirrhosis, primary sclerosing cholangitis, alcoholic liver disease, and autoimmune disorders. Furthermore, consistent with [62] we focus on those states in which there is no remaining time for the patient until the next required update, i.e., $\tau = 0$. Figure 6 shows our benchmark results indicating that the patient's optimal liver threshold does not increase under the current UNOS updating requirements. We observe that the current updating re-

quirements result in more cautious patients with respect to organ quality, i.e., the patient is more likely to accept an organ offer of higher quality compared to the case where he has no gaming opportunity due to the flexibility in reporting his health. Our results show a similar behavior for all other patient types.

Next, we examine the degree to which a patient can benefit under the current UNOS updating scheme. To do so, for all male patients of ages 22, 40 and 60 in each disease group, we calculate: (i) the average percentage increase in their lifetime, and (ii) the average percentage decrease in expected number of updates, compared to the updating scheme with no information asymmetry, as described in Section 3.4. Because our numerical experiments show similar trends for female patients, we only report the results for male patients. More specifically, we consider the two following cases. First, we focus on a case where updating incurs no inconvenience to the patients, i.e., equal immediate updating and doing reward in patients' MDPs as it appears in [62]. Second, we evaluate the effect of updating dissatisfaction on patients' benefits.

Figure 7 illustrates our results for the former case with $r^k_{DN}(h) = r^k_U(h) = 1.00$ for all patient types, $k \in \Xi$, and health statuses $h \in S^M$. These results are consistent with those reported in [62] showing that: (i) patients with relatively healthy and sick initial MELD scores benefit less than those with mid-range initial MELD scores in both disease groups, and (ii) patients in disease group 2 benefit more than those in disease group 1. The intuition behind observation (i) is that the MELD scores changes are typically slow over time in ESLD patients. Hence, those patients with initially healthy MELD scores often spend a fair amount of time having a MELD score in which they are unlikely to receive offers, regardless of whether or not they report minor improvements in their MELD score. Also, initially very ill patients are very close to death and receive frequent organ offers, leaving them little room to benefit. Furthermore, the intuition behind observation (ii) is that the diseases in group 1 are more aggressive than those in group 2. Hence, the MELD score changes in disease group 2 patients are slower than those in disease group 1, leading to observation (ii) based on the argument in (i).

However, when we penalize updating rewards so that $r^k_U(h) = 0.95$ and $r^k_{DN}(h) = 1.00$ for all patient types $k \in \Xi$, and $h \in S^M$, we obtain a new set of intuitive insights on patients'

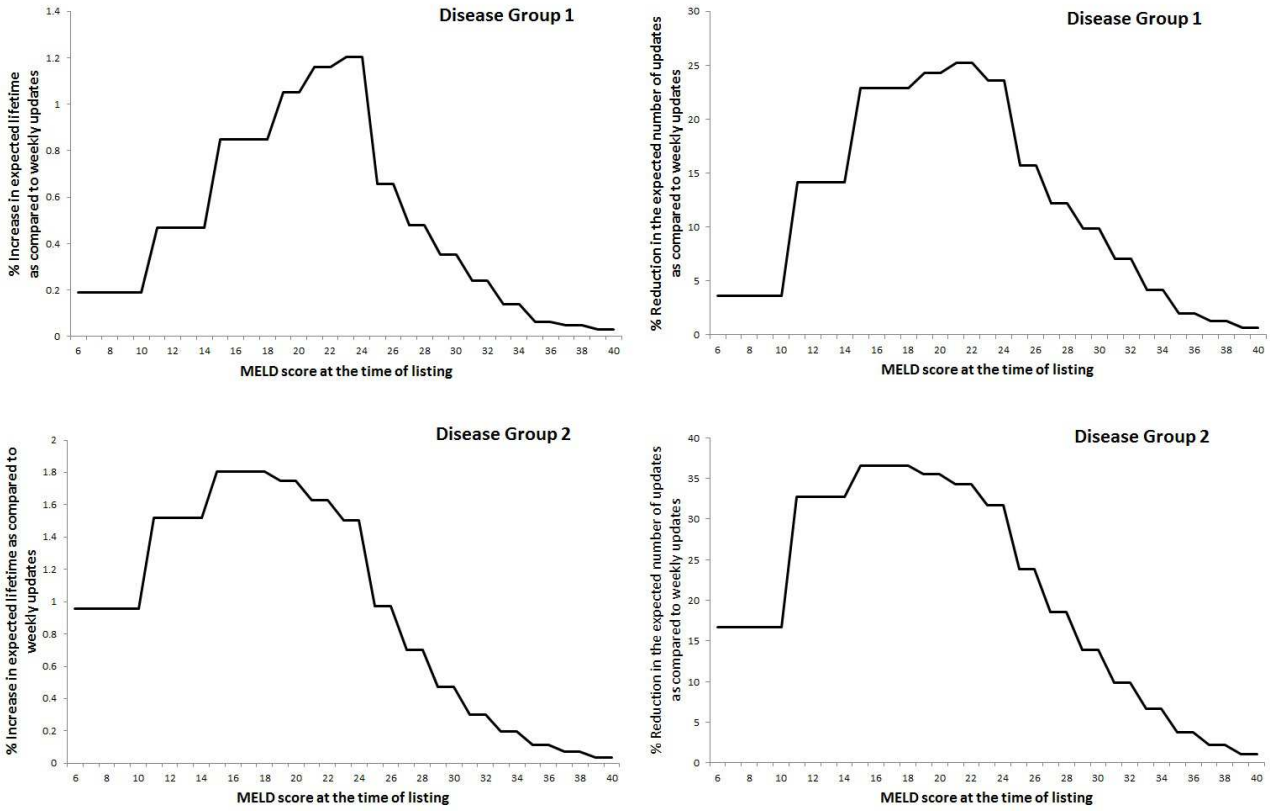

Figure 7: Patient benefits under UNOS updating with same updating and do nothing rewards

benefit, shown in Figure 8. These results show that the increase in the expected lifetime of the patients with relatively healthy or relatively sick initial MELD scores is less than those with mid-range MELD scores for both disease groups, similar to those observed for case one when $r_U^k(h) = r_{DN}^k(h)$. However, we observe a new trend in patients' benefit through the reduction in percentage expected number of updates. These results show that patients' benefit is decreasing as they get sicker as apposed to our findings for the former case. The intuition behind this observation is that the health status change is slow over time for the ESLD patients and the healthier patients spend longer time in those states that require less frequent updates due to the monotonicity of the updating schemes. Hence, healthier patients are less likely willing to update unless there is a significant change in their health status. This observation highlights the importance of patient dissatisfaction as when the objective is to assess the effect of changes in updating reporting requirements.

3.7.3 Mitigating Information Asymmetry in Liver Allocation

In this section, we present sets of alternative updating schemes that: (i) dominate the current updating scheme of UNOS with respect to both inequity and efficiency burdens, and (ii) dominate the heuristically constructed updating schemes by Içten [62]. Finally, we present the exact efficient frontier of all Pareto-optimal updating schemes. The system inequity and efficiency metrics in this section are defined as descried in Section 3.4.

Içten [62] provides an approximate efficient frontier of Pareto-optimal updating schemes. A noted earlier, these updating schemes are heuristically constructed. Hence, we first show that USDP model can be employed to identify any solution that dominates her proposed solutions. As shown in Table 2, she focuses on those monotonic updating schemes that are compatible with the current MELD score aggregation approach in UNOS health reporting requirements, i.e., curtain groups of MELD scores are required to assume the same updating timeframe.

Our approach to determine the set of dominant solutions with respect to the ones in Table 2 is as follows. We employ the USDP model for a given tolerance level μ on the system efficiency as it appears in each row of Table 2, and determine a monotonic updating

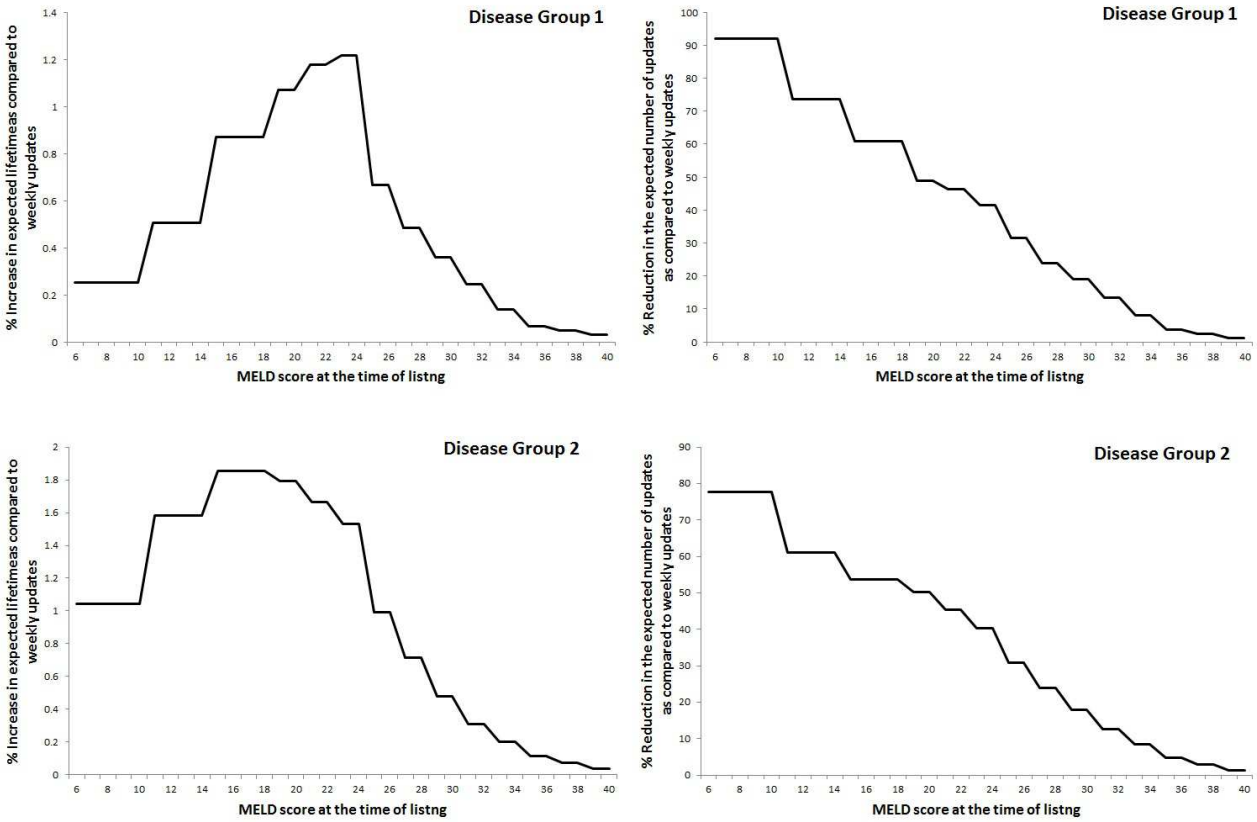

Figure 8: Patient benefits under UNOS updating when updating is penalized

Table 2: List of updating schemes on the approximated efficient frontier proposed in [62]

MELD Score					
6-10	**11-18**	**19-24**	**25-40**	**System Inequity (%)**	**System Efficiency (%)**
12	12	1	1	0.48	37.52
53	24	2	1	0.67	43.62
53	24	4	1	0.83	46.93
53	53	4	4	2.81	66.67
12	12	12	8	3.70	76.22

Table 3: Dominant updating schemes compared to [62]

MELD Score					
6-10	**11-18**	**19-24**	**25-40**	**System Inequity (%)**	**System Efficiency (%)**
18	18	2	1	0.43	38.10
10	10	7	1	0.57	45.01
46	46	7	1	0.79	48.39
53	37	6	5	2.70	67.00
39	39	9	9	3.53	77.66

scheme that minimizes the system inequity. These updating schemes are shown in Table 3 and Figure 9 illustrates their efficient frontier as apposed to the ones reported in Table 2.

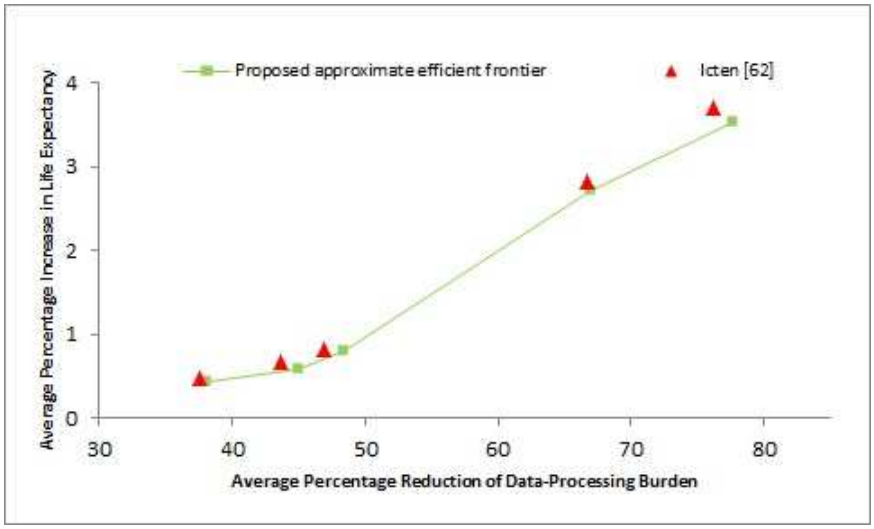

Figure 9: Efficient frontier of updating schemes compared to [62]

Next, we focus on our main goal in this study that is to balance information asymmetry and the data-processing burdens liver allocation system. To do so, we first restrict our attention to monotonic updating schemes compatible with the MELD score aggregation scheme of UNOS. Our results show that the UNOS updating scheme results in a 1.00%

increase in the average life expectancy of the patients compared to the continuous updating scheme which has no information asymmetry. On the other hand, it leads to a 46.06% decrease in the average expected number of updates, i.e., reduction in the data-processing burden. Figure 10 illustrates the set of all Pareto-optimal updating schemes denoted by squares and the current updating scheme of UNOS as a triangle.

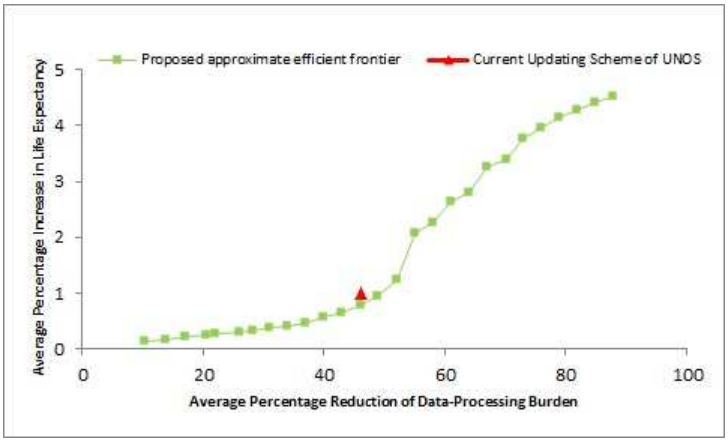

Figure 10: Efficient frontier of updating schemes compliant with the UNOS requirements

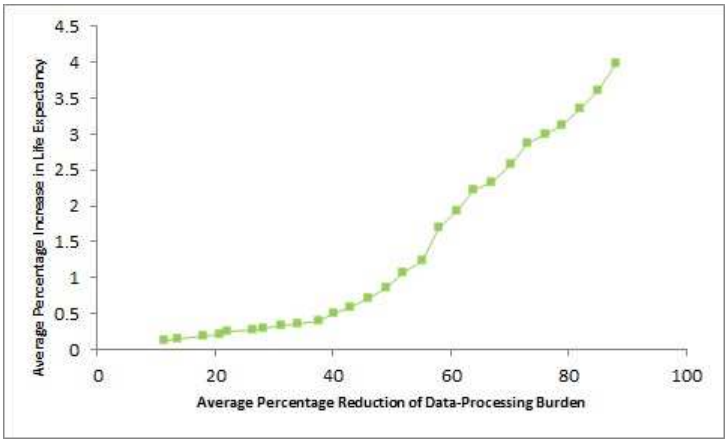

Figure 11: Efficient frontier of monotonic updating schemes

Finally, in Figure 11 we present the exact efficient frontier of all monotonic updating schemes regardless of the MELD aggregation approach in the current health reporting requirements of UNOS. Our results show that the current updating requirements can be improved with respect to both measures by requiring more frequent updates. Moreover, the current classification of MELD scores into 4 groups in the UNOS' updating requirements is not efficient and disregards a set of other efficient updating schemes.

As summary, the results presented in this section show that the current UNOS' updating scheme results in a system inequity due the patients' gaming ability. We showed that a typical patient can benefit from the current updating requirements and increase his life expectancy up to 1.00% and reduce the required data processing burden to 46.06% compared to an updating scheme with no information asymmetry between the UNOS and patients. Furthermore, our results indicate that the patients' dissatisfaction factor due to updating health status is crucial in determining the data processing burden of a typical updating scheme.

3.8 CONCLUSION

In this chapter, we propose a multi-objective model to revise the current health reporting requirements for the ESLD patients that: (i) minimizes the information asymmetry between the UNOS and patients, and (ii) minimizes the current significant data processing burden. This model focuses on both patients' and the societal perspective. First, we extend the prior patient decision-making problem to determine the optimal updating strategy and exploit the patients' optimal policies to quantify the degree to which patients can benefit from the flexibility in typical updating scheme. Second, combining stochastic programming and multi-objective optimization models, we propose a model to determine an updating scheme with minimum expected system inequity while ensuring a minimum desired level of reduction in the data processing burden. We further propose an iterative procedure using the latter model to approximate the exact efficient frontier of Pareto optimal updating schemes that improve the current reporting requirements with respect to one or both measures. We

calibrate the model parameters using clinical data and our computational results using the clinical data suggest that: (i) patients can benefit from the current UNOS' updating requirements by increasing their expected lifetime compared to an updating scheme with perfect information, (ii) healthier patients benefit more than the sicker patients under the current UNOS' updating scheme when reporting health status incurs dissatisfaction to the patients, and (iii) requiring the sicker (healthier) patients to update more (less) frequently than they must under the current policy can improve both metrics. We discuss the related future research directions in Chapter 5.

4.0 THE SURGICAL PATIENT ROUTING PROBLEM

4.1 INTRODUCTION

As we explained in Chapter 1, access to healthcare is a major problem faced by many patients. This issue is particularly important for rural patients or those who require special medical care. Furthermore, we discussed that to alleviate patients transportation issues, several organizations in the U.S. offer various transportation supports for patients, e.g., the Beneficiary Travel plans by the Veterans Health Administration (VHA) [34]. Although the volume of these services is increasing, the optimal design of such practices has attracted little attention in the literature [24]. In Chapter 1 we discussed the complexity of surgery scheduling process due to its dependency to many factors including resource availability [46]. When a medical center provides transportation services, the problem becomes even more complex. Unfortunately, planners typically ignore transportation considerations when scheduling medical appointments [23, 24].

Motivated by our collaborations with Pittsburgh Veterans Health Administration (VHA) hospital, in this chapter we propose an integrated approach that simultaneously considers patient scheduling and vehicle routing decisions. We refer to this problem as the *Surgical Patient Routing Problem* (*SPRP*). Specifically, we focus on scheduling and routing of the outpatient surgery requests, the requests that do not imply an overnight stay at the hospital: patients arrive on a day of surgery and leave after the completion of the surgical and post-operative procedures. The proposed approach assumes an open-booking scheduling framework, where surgical resources are shared among specialty teams, individual surgeons or surgical departments [56]. The planning horizon in *SPRP* is comprised of a finite number of time stages, which might span multiple days. The decisions in the model can be described

as a three-step process: assign surgeries to available time stages, schedule surgeries between operating rooms (ORs) at each time stage, and, finally, determine the transportation plan using a fleet of available vehicles.

The optimal decisions should minimize the total service cost of all patients, a cost that is defined as a weighted sum of patient's total travel time and the time spent at the hospital. Since long travel and waiting times cause dissatisfaction among patients [57], the total service cost adequately addresses the effectiveness of the appointment decisions. Maximizing on patient satisfaction is in line with VA goals [24], however our model can be extended to consider other objectives as well.

The remainder of this chapter is organized as follows. In Section 4.2, we briefly discuss the literature on medical appointment scheduling problem and transportation models. We conclude this section by identifying our contributions. In Section 4.3, we present a mixed integer programming (MIP) formulation for the *SPRP* and discuss its computational complexity. Next, we focus on a computationally tractable model, referred to as single-vehicle surgical patient routing problem *SSPRP*, that captures the needs of low-volume rural hospitals, where a relatively small number of surgeries are scheduled at each time stage. We show that solving the extended MIP formulation of both models is computationally prohibitive. To overcome this issues, we present an efficient set-partitioning formulation for the *SSPRP* and describe a branch-and-price algorithm for this problem in Section 4.4. Furthermore, we discuss several algorithmic strategies that enhance the performance of the proposed algorithm. Extensive computational experiments using medical data are presented in Section 4.5 to evaluate the effectiveness of our approach and to estimate the value of integrating surgery scheduling with routing decisions. We summarize general insights of our analysis and conclude in Section 4.6.

4.2 PRIOR WORK AND OUR CONTRIBUTIONS

The medical appointment scheduling problem is an active research area [57]. More specifically, managing and planning of operating rooms has been extensively studied [20, 25, 56, 57].

An extensive list of publications on operating room scheduling is maintained by [39]. The problem introduced in this dissertation differs from the existing routing and transportation models. In typical routing problems, such as the Vehicle Routing Problems, the objective is to minimize the travel distance subject to vehicle capacity or time window constraints. However, we consider a cost function that is defined as a sum of the total time that patients spend at the hospital and their total travel time, allowing us to establish a balance between the transportation and service costs. See [43, 71, 90, 98, 108, 113], for comprehensive reviews of the models and solution approaches for similar problems.

Our routing model has many similarities to the Delivery-Man Problem or Traveling Repairman Problem (TRP) and the K-Traveling Repairman Problem (KTRP) [49, 58, 78, 85, 99]. In the TRP the objective is to minimize the total arrival times to customers' locations by a single vehicle, rather than minimizing the length of the tours. The KTRP is a generalization of the TRP for multiple vehicles, where routing decisions are not limited to a single vehicle, but involve a homogenous fleet of vehicles. If we assume identical surgery durations, our model can be reduced to the KTRP. To the best of our knowledge, there is no exact algorithm for the KTRP and our solution method is the first exact approach for this type of problems.

Although scheduling medical appointments and vehicle routing problems have been extensively studied in the literature, to the best of our knowledge no previous work integrates these two classes of problems into a single framework. Additionally, our notion of the total service time cost is new in the medical appointment scheduling literature, as it is primarily focusing at improving the quality of medical services from the patients' perspective.

4.3 MATHEMATICAL FORMULATION

In this section, we introduce the *SPRP* model, a mathematical model that combines transportation and surgery scheduling of elective outpatient surgery requests, using a set of available ORs, a given planning horizon, and a fleet of homogeneous vehicles. We briefly discuss its complexity and focus on a computationally tractable case, exploit its structural properties, and reformulate it as a set-partitioning problem.

4.3.1 General Surgical Patient Routing Problem

The planning horizon in the *SPRP* model consists of a number of time stages, which may differ in duration. Figure 4.3.1 illustrates a planning horizon consisting of two days and 4 stages. At each stage, round-trip shared-type rides using a fleet of identical vehicles are provided for the patients.

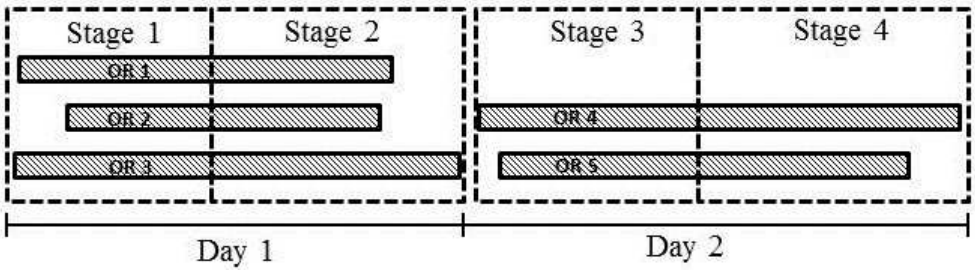

Figure 12: Planning horizon consisting of 2 days and 4 stages

The objective function is to minimize the total service time cost of the patients. As mentioned earlier, each patient's service cost is defined as the weighted sum of his/her total travel cost (home-to-hospital and hospital-to-home) and the cost associated with his/her time spent at the hospital. The home-to-hospital and hospital-to-home routes are referred to as the *pick-up* and the *drop-off* routes, respectively. Specifically, each instance of the SPRP is associated with the following set of parameters:

- $\mathcal{N}$: a set of geographically dispersed patients, where $|\mathcal{N}| = n$,
- $\mathcal{K}$: a set of time stages, where $|\mathcal{K}| = K$,
- $\mathcal{B}$: a set of available ORs, throughout the planning horizon consisting of K stages,
- $\mathcal{B}_k$: a set of available ORs at stage $k \in \mathcal{K}$, $\mathcal{B}_k \in \mathcal{B}$,
- $\mathcal{Q}_k$: a set of available vehicles at stage $k \in \mathcal{K}$,
- L_b^k: the session length of OR $b \in \mathcal{B}_k$ at stage $k \in \mathcal{K}$,
- ℓ_{ij}: the travel time between patients $i, j \in \mathcal{N}$,
- d_i: the surgery duration of patient $i \in \mathcal{N}$ (including pre- and post-incision periods),
- κ: the capacity of each vehicle,

- τ_b^k: the opening time of OR $b \in \mathcal{B}_k$ at stage $k \in \mathcal{K}$ (we assume that $\min_{b,k}\{\tau_b^k\} = 0$),
- c^r: the travel time cost, per patient per unit of travel time,
- c^h: the hospital time cost, per patient per unit of time spent at the hospital.

We define the following set of decision variables:

- $x_{ijb}^{qk} \in \{0,1\}$: 1 if vehicle q at stage k picks up patient i immediately before patient j, and patient i will be operated in OR b, and 0 otherwise,
- $\overline{x}_{ijb}^{qk} \in \{0,1\}$: 1 if vehicle q at stage k drops off patient i immediately before patient j, and patient i will be operated in OR b, and 0 otherwise,
- $z_{ib} \in \{0,1\}$: 1 if patient i's surgery is scheduled in OR b, and 0 otherwise,
- $u_{iqk} \in \{0,1\}$: 1 if vehicle q picks up patient i assigned to stage k, and 0 otherwise,
- $\overline{u}_{iqk} \in \{0,1\}$: 1 if vehicle q drops off patient i assigned to stage k, and 0 otherwise,
- $\eta_{ijb} \in \{0,1\}$: 1 if patients i and j are both assigned to OR b, and the surgery for patient i precedes the surgery of patient j, and 0 otherwise,
- t_{iq}^k: pick-up time of patient i by vehicle q assigned to stage k,
- $\overline{t}_{iq}^k$: drop-off time of patient i by vehicle q assigned to stage k,
- s_{ib}^k: start time of patient i's surgery in OR b at stage k.

Note that t_{iq}^k is defined with respect to the earliest OR's opening time (which is assumed to be time 0); thus, it can possibly take negative values if patients arrive before the earliest OR's opening time.

Let $G = (\widetilde{\mathcal{N}}, \mathcal{A})$ be a directed graph, where $\widetilde{\mathcal{N}}$ and $\mathcal{A}$ are its node and arc sets, respectively. Each node in $\widetilde{\mathcal{N}} = \mathcal{N} \bigcup \{0, n+1\}$ either corresponds to the location of patient $i \in \mathcal{N}$, or denotes the hospital, i.e., $\{0, n+1\}$. Let b_0 be a dummy OR assigned to node 0. The arc set $\mathcal{A}$ defines possible routes between patients' locations. For simplicity of exposition, we assume that $\ell_{ij} = \ell_{ji}$ for all $i, j \in \mathcal{N}$, $i \neq j$ and $\ell_{0i} = \ell_{i(n+1)}$ for all $i \in \mathcal{N}$. Note that for each stage $k \in \mathcal{K}$, variables $t_{(n+1)q}^k$ and $\overline{t}_{0q}^k$ define the arrival and departure times of vehicle $q \in \mathcal{Q}_k$ to and from the hospital, respectively. Without loss of generality, we do not consider multiple patient pick-ups or drop-offs, i.e., there is exactly one patient for each node in $\mathcal{N}$. Furthermore, $\mathcal{A}$ does not contain arcs of the type $(n+1, i)$ and $(i, 0)$ for all $i \in \widetilde{\mathcal{N}}$ as well as $(0, n+1)$.

The following set of constraints determines the pick-up schedule of the patients at each time stage, while simultaneously considering vehicle capacity restrictions and forbidding overtime in ORs:

$$\sum_{b\in B} z_{ib} = 1, \qquad i \in \mathcal{N}, \tag{4.1}$$

$$\sum_{q\in\mathcal{Q}_k} u_{iqk} = \sum_{b\in B_k} z_{ib}, \qquad i \in \mathcal{N}, k \in \mathcal{K}, \tag{4.2}$$

$$\sum_{i\in\mathcal{N}} u_{iqk} \leq \kappa, \qquad q \in \mathcal{Q}_k, k \in \mathcal{K}, \tag{4.3}$$

$$\sum_{j\in\mathcal{N}} x^{qk}_{0jb_0} \leq 1, \qquad q \in \mathcal{Q}_k, k \in \mathcal{K}, \tag{4.4}$$

$$\sum_{j\in\mathcal{N}} \sum_{b\in\mathcal{B}_k} x^{qk}_{j(n+1)b} \leq 1, \qquad q \in \mathcal{Q}_k, k \in \mathcal{K}, \tag{4.5}$$

$$\sum_{b\in\mathcal{B}_k} \sum_{j\in\widetilde{\mathcal{N}}\setminus\{0\}} x^{qk}_{ijb} = \sum_{b\in\mathcal{B}_k} \sum_{j\in\widetilde{\mathcal{N}}\setminus\{n+1\}} x^{qk}_{jib}, \qquad i \in \mathcal{N}, q \in \mathcal{Q}_k, k \in \mathcal{K}, \tag{4.6}$$

$$\sum_{b\in\mathcal{B}_k} \sum_{j\in\widetilde{\mathcal{N}}\setminus\{0\}} x^{qk}_{ijb} = u_{iqk}, \qquad i \in \mathcal{N}, q \in \mathcal{Q}_k, k \in \mathcal{K}, \tag{4.7}$$

$$t^k_{iq} + \ell_{ij} - M(1 - \sum_{b\in\mathcal{B}_k} x^{qk}_{ijb}) \leq t^k_{jq}, \qquad i, j \in \widetilde{\mathcal{N}}, q \in \mathcal{Q}_k, k \in \mathcal{K}, \tag{4.8}$$

$$-Mu_{iqk} \leq t^k_{iq} \leq Mu_{iqk}, \qquad i \in \widetilde{\mathcal{N}}, q \in \mathcal{Q}_k, k \in \mathcal{K}, \tag{4.9}$$

$$\sum_{i\in\mathcal{N}} d_i z_{ib} \leq L^k_b, \qquad b \in \mathcal{B}_k, k \in \mathcal{K}. \tag{4.10}$$

Constraints (4.1) ensure that each patient is assigned to exactly one OR. Constraints (4.2) guarantee that each patient is picked up by a single vehicle, and only the vehicles available at the corresponding time stage are used for transportation. Furthermore, the vehicle capacity restriction is enforced by constraints (4.3). The pick-up routes should satisfy certain properties: constraints (4.4) and (4.5) assure that each vehicle leaves and arrives to the hospital at most once, constraints (4.6) model the flow conservation for the pick-up routes, while constraints (4.7) guarantee that vehicles visit each patient's location in the pick-up routes at most once. The specific pick-up times are defined in constraints (4.8) and (4.9), where the positive constant M is large enough (e.g., $M \geq n(\max_{i,j\in\mathcal{N}} \ell_{ij} + \max_{i\in\mathcal{N}} d_i)$). Note that the former also acts as a sub-tour elimination constraint. Finally, overtimes in ORs are forbidden by (4.10). The following constraints determine the sequence of surgeries in the ORs:

$$\tau_b^k - M(1 - z_{ib}) \leq s_{ib}^k, \qquad i \in \mathcal{N}, k \in \mathcal{K}, b \in \mathcal{B}_k, \tag{4.11}$$

$$t_{(n+1)q}^k - M(1 - \sum_{j \in \widetilde{N} \setminus \{0\}} x_{ijb}^{qk}) \leq s_{ib}^k, \qquad i \in \mathcal{N}, q \in \mathcal{Q}_k, k \in \mathcal{K}, b \in \mathcal{B}_k, \tag{4.12}$$

$$s_{ib}^k + d_i - M(1 - \eta_{ijb}) \leq s_{jb}^k, \qquad i, j \in \mathcal{N}, k \in \mathcal{K}, b \in \mathcal{B}_k, \tag{4.13}$$

$$\eta_{ijb} + \eta_{jib} \leq z_{ib}, \qquad i, j \in \mathcal{N}, b \in \mathcal{B}, \tag{4.14}$$

$$\eta_{ijb} + \eta_{jib} \leq z_{jb}, \qquad i, j \in \mathcal{N}, b \in \mathcal{B}, \tag{4.15}$$

$$\eta_{ijb} + \eta_{jib} \geq z_{ib} + z_{jb} - 1, \qquad i, j \in \mathcal{N}, b \in \mathcal{B}. \tag{4.16}$$

Constraints (4.11) and (4.12) specify that the surgeries start after the OR start times and the arrival of vehicles to the hospital. Constraints (4.13)-(4.16) sequence the surgeries in the ORs and determine their start times.

Finally, the following set of constraints determines the drop-off schedule of the patients after the completion of their surgeries at each time stage:

$$\sum_{q \in \mathcal{Q}_k} \overline{u}_{iqk} = \sum_{b \in B_k} z_{ib}, \qquad i \in \mathcal{N}, k \in \mathcal{K}, \tag{4.17}$$

$$\sum_{i \in \mathcal{N}} \overline{u}_{iqk} \leq \kappa, \qquad q \in \mathcal{Q}_k, k \in \mathcal{K}, \tag{4.18}$$

$$\sum_{j \in \mathcal{N}} \overline{x}_{0jb_0}^{qk} \leq 1, \qquad q \in \mathcal{Q}_k, k \in \mathcal{K}, \tag{4.19}$$

$$\sum_{j \in \mathcal{N}} \sum_{b \in \mathcal{B}_k} \overline{x}_{j(n+1)b}^{qk} \leq 1, \qquad q \in \mathcal{Q}_k, k \in \mathcal{K}, \tag{4.20}$$

$$\sum_{b \in \mathcal{B}_k} \sum_{j \in \widetilde{\mathcal{N}} \setminus \{0\}} \overline{x}_{ijb}^{qk} = \sum_{b \in \mathcal{B}_k} \sum_{j \in \widetilde{\mathcal{N}} \setminus \{n+1\}} \overline{x}_{jib}^{qk}, \qquad i \in \mathcal{N}, q \in \mathcal{Q}_k, k \in \mathcal{K}, \tag{4.21}$$

$$\sum_{b \in \mathcal{B}_k} \sum_{j \in \widetilde{\mathcal{N}} \setminus \{0\}} \overline{x}_{ijb}^{qk} = \overline{u}_{iqk}, \qquad i \in \mathcal{N}, q \in \mathcal{Q}_k, k \in \mathcal{K}, \tag{4.22}$$

$$\overline{t}_{iq}^k + \ell_{ij} - M(1 - \sum_{b \in \mathcal{B}_k} \overline{x}_{ijb}^{qk}) \leq \overline{t}_{jq}^k, \qquad i, j \in \widetilde{\mathcal{N}}, q \in \mathcal{Q}_k, k \in \mathcal{K}, \tag{4.23}$$

$$-M\overline{u}_{iqk} \leq \overline{t}_{iq}^k \leq M\overline{u}_{iqk}, \qquad i \in \widetilde{\mathcal{N}}, q \in \mathcal{Q}_k, k \in \mathcal{K}, \tag{4.24}$$

$$\overline{t}_{0q}^k \geq s_{ib}^k + d_i - M(1 - \sum_{j \in \widetilde{N} \setminus \{0\}} \overline{x}_{ijb}^{qk}), \qquad i \in \mathcal{N}, q \in \mathcal{Q}_k, k \in \mathcal{K}, b \in \mathcal{B}_k. \tag{4.25}$$

Constraints (4.17)-(4.24) are similar to those defining the pick-up routes given above. Observe that in (4.17) we assume that the vehicles used to drop off the patients at each stage are the same as those used to determine patients' pick-up schedule. Constraints (4.25) ensure that each vehicle leaves the hospital once the surgeries of all patients assigned to the vehicle are completed.

The objective function is the weighted sum of the patients' total travel time and the time spent at the hospital (or, simply, hospital time) denoted by Φ^r and Φ^h, respectively:

$$\Phi^r = \sum_{i\in\mathcal{N}}\sum_{k\in\mathcal{K}}\sum_{q\in\mathcal{Q}_k}\left\{u_{iqk}(t^k_{(n+1)q} - t^k_{iq}) + \overline{u}_{iqk}(\overline{t}^k_{iq} - \overline{t}^k_{0q})\right\}, \tag{4.26}$$

$$\Phi^h = \sum_{i\in\mathcal{N}}\sum_{k\in\mathcal{K}}\sum_{q\in\mathcal{Q}_k}\left\{\overline{u}_{iqk}\overline{t}^k_{0q} - u_{iqk}t^k_{(n+1)q}\right\}, \tag{4.27}$$

which results in the following MIP formulation:

$$\begin{aligned}
\text{[SPRP]} \qquad & \min_{\mathcal{X}} \quad c^r\Phi^r + c^h\Phi^h \\
\text{subject to} \quad & (4.1) - (4.25), \\
& x^{qk}_{ijb}, \overline{x}^{qk}_{ijb} \in \{0,1\}, && \forall i,j \in \widetilde{\mathcal{N}}, \forall q \in \mathcal{Q}_k, k \in \mathcal{K}, b \in \mathcal{B}_k, && (4.28)\\
& z_{ib}, u_{iqk}, \overline{u}_{iqk}, \eta_{ijb} \in \{0,1\}, && i,j \in \mathcal{N}, q \in \mathcal{Q}_k, k \in \mathcal{K}, b \in \mathcal{B}_k, && (4.29)\\
& s^k_{ib} \geq 0, && i \in \widetilde{\mathcal{N}}, q \in \mathcal{Q}_k, k \in \mathcal{K}, b \in \mathcal{B}_k, && (4.30)
\end{aligned}$$

where $\mathcal{X}$ defines a joint vector of decision variables, i.e., $\mathcal{X} = (x, \bar{x}, z, u, \bar{u}, \eta, s, t, \bar{t})$. Note that the nonlinear terms in the objective of SPRP can be easily linearized. However, we observe that a large number of decision variables and constraints in the obtained MIP formulation makes its solution rather challenging. For example, even a special case of SPRP described in the next section and referred to as BSPRP, is not solvable for reasonably sized instances by off-the-shelf MIP solvers (see Table 5 and the respective discussion in Section 4.5).

4.3.2 Batch Surgical Patient Routing Problem (BSPRP)

In the considered special case of SPRP referred to as the Batch Surgical Patient Routing Problem (BSPRP), we make the following two assumptions:

A1. There is exactly one vehicle available at each stage, i.e., $|\mathcal{Q}_k| = 1$ for all $k \in \mathcal{K}$. Thus, each patient arrives at and leaves from the hospital using the same vehicle.

A2. Instead of considering a separate time limit constraint for each OR as in (4.10), we only require that the total length of surgeries scheduled for each time stage $k \in \mathcal{K}$ should not exceed $L^k = \sum_{b\in\mathcal{B}_k} L^k_b$.

Assumption **A1** captures the needs of low-volume hospitals, where a rather small number of surgeries is usually scheduled at each time stage. Thus, having a single vehicle for each time stage is typically sufficient to satisfy all transportation requirements with respect to the total available OR time in the hospital.

Assumption **A2** implies that BSPRP does not require the assignment (and sequencing) of each patient's surgery to a specific OR, see constraints (4.11)-(4.16), which could be performed by a separate (optimization) procedure. Admittedly, the optimal solution of BSPRP may violate constraints (4.10). However, we assume that this either does not occur (which should be the case in most scenarios as long as the OR utilization is not too high), or the issue can be handled by the hospital management separately on a case-by-case basis introducing overtime in particular ORs.

While assumptions **A1** and **A2** substantially simplify the original model, BSPRP is still intractable using state-of-the-art commercial MIP solvers (see Section 4.5). However, under **A1** and **A2** we establish some structural properties that we subsequently exploit to develop an efficient branch-and-price solution approach (see Sections 4.4 and 4.5).

The BSPRP formulation uses the following notation:

- $x_{ij}^k \in \{0,1\}$: 1 if at stage k patient i is picked up immediately before patient j, and 0 otherwise,
- $\overline{x}_{ij}^k \in \{0,1\}$: 1 if at stage k patient i is dropped off immediately before patient j, and 0 otherwise,
- $z_i^k \in \{0,1\}$: 1 if patient i is assigned to stage k, and 0 otherwise,
- t_{ik}: pick-up time of patient i if assigned to time stage k,
- $\overline{t}_{ik}$: drop-off time of patient i if assigned to time stage k,
- t_i: pick-up time of patient i,
- $\overline{t}_i$: drop-off time of patient i,

where time variables are defined with respect to 0 (the earliest possible departure time of a vehicle for the patients' pick-up). A feasible pick-up/drop-off route of the vehicle at each time stage should only visit each patient at most once, and the sum of surgery durations of all visited patients should not exceed the session length of the stage.

$$\sum_{j\in\mathcal{N}} x_{0j}^k = 1, \qquad \forall k \in \mathcal{K}, \tag{4.31}$$

$$\sum_{j\in\mathcal{N}} x_{j(n+1)}^k = 1, \qquad \forall k \in \mathcal{K}, \tag{4.32}$$

$$\sum_{j\in\widetilde{\mathcal{N}}\setminus\{0\}} x_{ij}^k = \sum_{j\in\widetilde{\mathcal{N}}\setminus\{n+1\}} x_{ji}^k, \qquad \forall i \in \mathcal{N}, \forall k \in \mathcal{K}, \tag{4.33}$$

$$t_{ik} + \ell_{ij} - M(1 - x_{ij}^k) \leq t_{jk}, \qquad \forall i, j \in \widetilde{\mathcal{N}}, \forall k \in \mathcal{K}, \tag{4.34}$$

$$\sum_{i\in\widetilde{\mathcal{N}}\setminus\{n+1\}} x_{ij}^k = z_j^k, \qquad \forall j \in \mathcal{N}, \forall k \in \mathcal{K}, \tag{4.35}$$

$$\sum_{k\in\mathcal{K}} z_j^k = 1, \qquad \forall j \in \mathcal{N}, \tag{4.36}$$

$$\sum_{i\in\mathcal{N}} d_i z_i^k \leq L^k, \qquad \forall k \in \mathcal{K}, \tag{4.37}$$

In these constraints we use our previous definition as $L^k = \sum_{b\in\mathcal{B}_k} L_b^k$. Similarly, using assumptions **A1** and **A2** and simplifying the corresponding constraints from Section 4.3, we specify the drop-off routes:

$$\sum_{j\in\mathcal{N}} \overline{x}_{0j}^k = 1, \qquad \forall k \in \mathcal{K}, \tag{4.38}$$

$$\sum_{j\in\mathcal{N}} \overline{x}_{j(n+1)}^k = 1, \qquad \forall k \in \mathcal{K}, \tag{4.39}$$

$$\sum_{j\in\widetilde{\mathcal{N}}\setminus\{0\}} \overline{x}_{ij}^k = \sum_{j\in\widetilde{\mathcal{N}}\setminus\{n+1\}} \overline{x}_{ji}^k, \qquad \forall i \in \mathcal{N}, \forall k \in \mathcal{K}, \tag{4.40}$$

$$\bar{t}_{ik} + \ell_{ij} - M(1 - \overline{x}_{ij}^k) \leq \bar{t}_{jk}, \qquad \forall i, j \in \widetilde{\mathcal{N}}, \forall k \in \mathcal{K}, \tag{4.41}$$

$$t_{(n+1)k} + \sum_{i\in\mathcal{N}} d_i z_i^k \leq \bar{t}_{0k}, \qquad \forall i \in \mathcal{N}, \forall k \in \mathcal{K}, \tag{4.42}$$

and the patients' pick-up and drop-off times:

$$t_i - M(1 - z_i^k) \leq t_{ik}, \qquad \forall i \in \mathcal{N}, \forall k \in \mathcal{K}, \tag{4.43}$$

$$\bar{t}_i + M(1 - z_i^k) \geq \bar{t}_{ik}, \qquad \forall i \in \mathcal{N}, \forall k \in \mathcal{K}. \tag{4.44}$$

The objective function is determined using the patients' travel and hospital times given by Φ^r and Φ^h, respectively:

$$\Phi^r = \sum_{i\in\mathcal{N}} \left\{ \bar{t}_i - t_i - \sum_{k\in\mathcal{K}} (\bar{t}_{0k} - t_{(n+1)k}) z_i^k \right\}, \tag{4.45}$$

$$\Phi^h = \sum_{i\in\mathcal{N}} \sum_{k\in\mathcal{K}} \left\{ (\bar{t}_{0k} - t_{(n+1)k}) z_i^k \right\}, \tag{4.46}$$

which are simplified version of (4.26)-(4.27). Using the above constraints, model BSPRP is given by:

$$\begin{aligned}
[\text{BSPRP}] \quad & \min_{x,\bar{x},z,t,\bar{t}} \quad c^r\Phi^r + c^h\Phi^h \\
& \text{subject to} \quad (4.31) - (4.44), \\
& \qquad x_{ij}^k, \overline{x}_{ij}^k, z_i^k \in \{0,1\}, \qquad \forall i,j \in \widetilde{\mathcal{N}}, \forall k \in \mathcal{K}. && (4.47)\\
& \qquad t_{ik}, \bar{t}_{ik}, t_i, \bar{t}_i \geq 0, \qquad \forall i \in \widetilde{\mathcal{N}}, \forall k \in \mathcal{K}. && (4.48)
\end{aligned}$$

Next, Proposition 4.3.1 demonstrates that there always exists an optimal solution to BSPRP such that the pick-up and the drop-off routes are in the reverse order of each other.

Proposition 4.3.1. *There exists an optimal solution $\mathcal{X}^* = (x, \bar{x}, z, t, \bar{t})$ for BSPRP such that if $x^{*k}_{ij} = 1$, then $\overline{x}^{*k}_{ji} = 1$ for all $i, j \in \widetilde{\mathcal{N}}$ and $k \in \mathcal{K}$.*

Proof. For $k \in \mathcal{K}$, let $\mathcal{N}_k = \{u \in \mathcal{N} \mid z_u^k = 1\}$. Let $I_{\mathcal{N}_k} =< 0, i_1, i_2, ..., i_{|\mathcal{N}_k|}, n+1 >$ and $J_{\mathcal{N}_k} =< 0, j_1, j_2, ..., j_{|\mathcal{N}_k|}, n+1 >$ be any arbitrary pick-up and drop-off orderings of patients, respectively, where 0 and $n+1$ denote the hospital. Next, let $Q =< 0, q_1, q_2, ..., q_{|\mathcal{N}_k|}, n+1 >$ be a pick-up ordering of patients such that

$$\ell_{q_1q_2} + 2\ell_{q_2q_3} + ... + |\mathcal{N}_k| \cdot \ell_{q_{|\mathcal{N}_k|}(n+1)} = \min_I \left\{ \ell_{i_1i_2} + 2\ell_{i_2i_3} + ... + |\mathcal{N}_k| \cdot \ell_{i_{|\mathcal{N}_k|}(n+1)} \right\}, \tag{4.49}$$

i.e., Q corresponds a pick-up ordering of patients that results in the smallest value of the total travel time.

Let patient $p \in \mathcal{N}_k$ be the vth patient that is dropped off (in drop-off ordering J) and the uth patient who is picked up (in pick-up ordering I). Then the service time cost of patient p, denoted by $C_p^{(u,v)}$, is

$$C_p^{(u,v)} = c^r \left(\ell_{0j_1} + \ell_{j_1j_2} + ... + \ell_{j_{v-1}j_v} + \ell_{i_ui_{u+1}} + \ell_{i_{u+1}i_{u+2}} + ... + \ell_{i_{(u+1)}i_{|\mathcal{N}_k|}} \right) + c^h \cdot \sum_{u\in\mathcal{N}_k} d_u,$$

and the total service time cost of the patients in $\mathcal{N}_k$ is

$$\begin{aligned}
c^r\Phi^r + c^h\Phi^h &= c^r\Big(|\mathcal{N}_k|\cdot\ell_{0j_1} + (|\mathcal{N}_k|-1)\cdot\ell_{j_1j_2} + \ldots + \ell_{j_{|\mathcal{N}_k|-1}j_{|\mathcal{N}_k|}} + |\mathcal{N}_k|\cdot\ell_{0i_{|\mathcal{N}_k|}} \\
&\quad +(|\mathcal{N}_k|-1)\cdot\ell_{i_{|\mathcal{N}_k|-1}i_{|\mathcal{N}_k|}} + \ldots + \ell_{i_1i_2}\Big) + c^h\cdot|\mathcal{N}_k|\cdot\sum_{u\in\mathcal{N}_k} d_u \\
&= c^r\Big(|\mathcal{N}_k|\cdot(\ell_{0j_1}+\ell_{0i_{|\mathcal{N}_k|}}) + (|\mathcal{N}_k|-1)\cdot(\ell_{j_1j_2}+\ell_{i_{|\mathcal{N}_k|-1}i_{|\mathcal{N}_k|}}) + \ldots + (\ell_{j_{|\mathcal{N}_k|-1}j_{|\mathcal{N}_k|}}+\ell_{i_1i_2})\Big) \\
&\quad + c^h\cdot|\mathcal{N}_k|\cdot\sum_{u\in\mathcal{N}_k} d_u \\
&\geq c^r\Big(|\mathcal{N}_k|\cdot(\ell_{0j_1}+\ell_{0q_{|\mathcal{N}_k|}}) + (|\mathcal{N}_k|-1)\cdot(\ell_{j_1j_2}+\ell_{q_{|\mathcal{N}_k|-1}q_{|\mathcal{N}_k|}}) + \ldots + (\ell_{j_{|\mathcal{N}_k|-1}j_{|\mathcal{N}_k|}}+\ell_{q_1q_2})\Big) \\
&\quad + c^h\cdot|\mathcal{N}_k|\cdot\sum_{u\in\mathcal{N}_k} d_u \\
&\geq 2c^r\Big(\ell_{q_1q_2} + \ldots + (|\mathcal{N}_k|-1)\cdot\ell_{q_{|\mathcal{N}_k|-1}q_{|\mathcal{N}_k|}} + |\mathcal{N}_k|\cdot\ell_{0q_{|\mathcal{N}_k|}}\Big) + c^h\cdot|\mathcal{N}_k|\cdot\sum_{u\in|\mathcal{N}_k|} d_u,
\end{aligned}$$

where the last two inequalities follow by definition of Q in (4.49). Thus,

$$\sum_{p\in\mathcal{N}_k}\Big(c^r\Phi^r + c^h\Phi^h\Big) = 2c^r\Big(\ell_{q_1q_2} + \ldots + (|\mathcal{N}_k|-1)\cdot\ell_{q_{|\mathcal{N}_k|-1}q_{|\mathcal{N}_k|}} + |\mathcal{N}_k|\cdot\ell_{0q_{|\mathcal{N}_k|}}\Big) + c^h\cdot|\mathcal{N}_k|\cdot\sum_{u\in\mathcal{N}_k} d_u. \tag{4.50}$$

□

Due to the presence of the big-M constraints in BSPRP, its linear programming (LP) relaxation is usually weak. However, using Proposition 4.3.1 we reformulate BSPRP to obtain an equivalent MIP model that provides better LP relaxation bounds. Special cases of this formulation with $c^h = 0$, or, equivalently, $d_i = d_j$ for all $i, j \in \mathcal{N}$, are considered by [58] and [85] for the K-Traveling Repairman Problem.

Let binary variable $\gamma_{ij}^k = 1$ if both patients i and j are assigned to the same time stage k. Then using (4.50) we obtain the following equivalent MIP model:

$$[\text{BSPRP}]\ \min_{x,z,\gamma,t}\quad \sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{N}}\left\{2c^r t_{ik} + c^h\left(d_i z_i^k + \sum_{j\in\mathcal{N}\setminus\{i\}} d_j\gamma_{ij}^k\right)\right\} \tag{4.51}$$

subject to

$$(4.31)-(4.37),$$

$$\gamma_{ij}^k = z_i^k z_j^k, \qquad \forall i,j\in\mathcal{N}, \forall k\in\mathcal{K}, \tag{4.52}$$

$$t_i^k \geq 0, \qquad \forall i\in\widetilde{\mathcal{N}}, \forall k\in\mathcal{K}, \tag{4.53}$$

$$x_{ij}^k, z_i^k, \gamma_{ij}^k \in \{0,1\}, \qquad \forall i,j\in\widetilde{\mathcal{N}}, \forall k\in\mathcal{K}. \tag{4.54}$$

Note that nonlinear constraints (4.52) link the patient assignment decisions to the time stages. They can be easily linearized using a standard approach with an additional set of linear constraints. While MIP model (4.51)-(4.54) provides tighter LP relaxation bounds compared to the previous formulation, its solution using commercial MIP solvers is still computationally expensive, e.g., CPLEX 12.4 yields an optimality gap of at least 29% in solving instances of BSPRP with 20 patients and 4 stages after 3 hours. In order to overcome this concern, in the next section we present a set-partitioning reformulation of BSPRP, which we consequently exploit to develop a branch-and-price algorithm.

4.4 A BRANCH-AND-PRICE APPROACH

For ease of exposition, in this section we assume that the time available in ORs is the same for all stages, i.e., $L^{k_1} = L^{k_2} = L$ for all $k_1,\ k_2 \in \mathcal{K}$. This restriction requires only a slight modifications of the proposed approach when adjusting for more general problem instances. Due to the special structure of the objective function in BSPRP, Proposition 4.3.1 indicates that it is always optimal to pick up the patients in the reverse order of their drop-off sequence and vice versa. We refer to an assignment of patients to a time stage, their drop-off and pick-up sequences as a transportation *route* throughout the rest of this section.

Let R be the set of all potential feasible transportation routes. Let binary decision variable θ_r, $r \in R$, equals 1 if route r is chosen in the corresponding solution. Additionally, let parameter $a_{ir} \in \{0, 1\}$ indicate whether route r includes patient i. Let ϕ_r be the total cost of travel and hospital times for all patients visited in route r, which is computed using (4.45)-(4.46). Then BSPRP can be reformulated as a set-partitioning problem (SP):

$$\text{[SP]} \quad \min \quad \sum_{r \in R} \phi_r \theta_r \tag{4.55}$$

$$\text{subject to} \quad \sum_{r \in R} a_{ir} \theta_r = 1, \qquad i \in \mathcal{N}, \tag{4.56}$$

$$\sum_{r \in R} \theta_r \leq K, \tag{4.57}$$

$$\theta_r \in \{0, 1\}, \qquad r \in R, \tag{4.58}$$

where constraints (4.56) ensure that each patient is assigned to exactly one route and constraint (4.57) provides the upper bound on the total number of chosen routes. In general, the LP relaxation of formulation (4.55)-(4.58) provides tight bounds. However, the size of R is exponentially large, which results in an excessive number of decision variables and model parameters in (4.55)-(4.58). Therefore, we propose to employ a branch-and-price framework to generate promising routes on "as needed" basis. In the remainder of the chapter, we refer to the LP relaxation of model (4.55)-(4.58) as the master problem (MP).

4.4.1 Route Generation

Branch and price is a technique that incorporates column generation within a branch-and-bound procedure [10]. Define the restricted master problem, denoted by RMP(R'), as the LP relaxation of SP that consists of a restricted set of columns generated so far, denoted by $R' \subseteq R$. At each node of the search tree, the column generation method iteratively solves the RMP(R') and a pricing problem. The purpose of the pricing problem is either to produce columns with the most negative reduced costs based on the dual solution of current RMP(R'), or to prove that none exists. At each iteration, newly generated columns are introduced to RMP(R'), and the process terminates and a lower bound for the corresponding node is obtained whenever no additional column price out favorably. We refer the reader to [10] and [77] for a more detailed description of the branch-and-price framework.

Consider the following dual variables of RMP(R'):

- π_i: dual variable corresponding to constraint (4.56) for patient i;
- μ: dual variable corresponding to constraint (4.57).

The reduced cost $\bar{\phi}_r$ of a potential route, $r \in R$, is given by

$$\overline{\phi}_r = \phi_r - \sum_{i \in \mathcal{N}} a_{ir}\pi_i - \mu. \tag{4.59}$$

Note that any optimal solution of RMP(R') is a feasible solution to MP. However, it is not (necessarily) an optimal solution, unless there is no column left in $R \setminus R'$ that prices out favorably. This dynamic process of generating columns is called pricing, and the problem itself is referred to as the pricing problem. Given a dual solution π to RMP(R') and an

integer λ, we propose an integer-programming formulation for the pricing problem, referred to as RPP$^{\lambda}(\pi)$. Similar formulations are considered in [17, 54, 65, 91, 95], and [58] when $d_i = 0$ for all $i \in \mathcal{N}$ or $c^h = 0$.

Let $R(\lambda)$ be the set of all potential transportation routes that consist of exactly λ patients, where $\sum_{i \in R(\lambda)} d_i \leq L$ and $\lambda \leq \kappa$, so $R = \bigcup_{\lambda=1}^{\kappa} R(\lambda)$. Our pricing problem determines a column (i.e., drop-off route) with the smallest reduced cost among those in $R(\lambda)$ for each $\lambda \leq \kappa$. We assume that there always exists λ such that the overtime restriction is retained. Specifically, we model this problem as a problem of finding a path with the smallest cost on a multi-layer network, denoted by $\mathcal{F}^{\lambda}$. Each network $\mathcal{F}^{\lambda}$ consists of λ layers, where each layer encompasses n nodes associated with the patients. Moreover, $\mathcal{F}^{\lambda}$ includes nodes 0 and $n+1$ that represent the start and the end location of the each route. Starting from node 0 and ending at $n+1$, each feasible path on $\mathcal{F}^{\lambda}$ is composed of distinct nodes selected at each layer, defining the drop-off sequence of the selected patients, which also provides us with the pick-up sequence according to Proposition 4.3.1.

The integer programming formulation of RPP$^{\lambda}(\pi)$ uses the following decision variables:

- $\zeta_i^t \in \{0, 1\}$ if node i is selected in layer t, for $i \in \mathcal{N}$ and $t = 1, ..., \lambda$;
- $w_{ij}^t \in \{0, 1\}$ if nodes i and j are selected in layers t and $t+1$, for $i, j \in \mathcal{N}(i \neq j)$ and $t = 1, ..., \lambda - 1$.

Following [58], we refer to these variables as the *position* and the *transition* variables, respectively. Given λ as the possible number of patients in a route, the objective function for RPP$^{\lambda}(\pi)$ can be modeled as:

$$\mathcal{Z}_{\lambda}(\zeta_i^t, w_{ij}^t) = 2c^r \left(\lambda \sum_{j \in \mathcal{N}} \ell_{0j} \zeta_j^1 + \sum_{t=1}^{\lambda-1} \sum_{i,j \in \mathcal{N}, i \neq j} (\lambda - t) \ell_{ij} w_{ij}^t \right) + \sum_{t=1}^{\lambda} \sum_{i \in \mathcal{N}} (c^h \lambda d_i - \pi_i) \zeta_i^t.$$

As a result, for a given $\lambda \leq \kappa$ those columns in $R \setminus R'$ for which $\mathcal{Z}_{\lambda}(\zeta_i^t, w_{ij}^t) - \mu \leq 0$, are priced out and introduced to RMP(R'). Therefore, using the dual variable π, the pricing problem can be described as:

$$
\begin{aligned}
[\mathrm{RPP}^{\lambda}(\pi)] \quad \text{minimize} \quad & \mathcal{Z}_{\lambda}(\zeta_i^t, w_{ij}^t) && && (4.60)\\
\text{subject to} \quad & \sum_{t=1}^{\lambda} \zeta_i^t \leq 1, && i \in \mathcal{N}, && (4.61)\\
& \sum_{i \in \mathcal{N}} \zeta_i^t = 1, && t = 1, ..., \lambda, && (4.62)\\
& \sum_{j \in \mathcal{N} \setminus \{i\}} w_{ij}^t = \zeta_i^t, && t = 1, ..., \lambda - 1, i \in \mathcal{N}, && (4.63)\\
& \sum_{j \in \mathcal{N} \setminus \{i\}} w_{ji}^t = \zeta_i^{(t+1)}, && t = 1, ..., \lambda - 1, i \in \mathcal{N}, && (4.64)\\
& \sum_{j \in \mathcal{N}} \sum_{t=1}^{\lambda} d_j \zeta_j^t \leq L, && && (4.65)\\
& \zeta_i^t \in \{0, 1\}, && t = 1, ..., \lambda, i \in \mathcal{N}, && (4.66)\\
& w_{ij}^t \in \{0, 1\}, && t = 1, ..., \lambda - 1, i, j \in \mathcal{N}. && (4.67)
\end{aligned}
$$

The formulation ensures that each node is visited at most once (4.61), exactly one node is selected in each layer (4.62), while constraints (4.63) and (4.64) link the position and transition variables. Lastly, constraint (4.65) prevents overtime at time stage. It can be easily shown that $w_{ij}^t = \zeta_i^t \cdot \zeta_j^{t+1}$ for all $i, j \in \mathcal{N}$ in the an optimal solution. Hence, the binary restrictions for variables w_{ij}^t can be relaxed and replaced by nonnegativity requirements.

Our computational tests in Section 4.5 show that the proposed pricing problem is solvable using commercial solvers. However, in the next section we introduce a heuristic approach to generate multiple potential columns to enhance the performance of the method at the first iterations of the column generation method.

4.4.2 Algorithmic Enhancements

Next, we discuss several computational considerations that are important in implementing our branch-and-price algorithm. In particular, we describe our branching strategy in Section 4.4.2.1 and introduce a greedy column generation heuristic in Section 4.4.2.2.

4.4.2.1 Patient-Pair Branching

Using the traditional branching on variables for solving large-scale set-partitioning problems often results in an unbalanced search tree and may destroy the structure of the pricing problem [82]. To overcome this difficulty, we adapt an alternative branching from [102], which is known to be a very effective strategy for set-partitioning applications including vehicle routing and crew scheduling problems [77].

The branching scheme used in [102] is based on the following proposition. Although the authors do not consider column generation, their branching scheme is effective in this context as well [10].

Proposition 4.4.1. *[10] If A is a 0–1 matrix, and a basic solution to Ax=1 is fractional, i.e., at least one of the components of x is fractional, then there exist two rows s and t of the master problem such that*

$$0 < \sum_{k:\ a_{sk}=1,\ a_{tk}=1} x_k < 1.$$

Note that the matrix associated with the set-partitioning constraints (4.56) in any restricted master problem is a 0–1 matrix. Hence, using the result of Proposition 4.4.1 the pair of branching constraints are:

$$\sum_{k:\ a_{sk}=1,\ a_{tk}=1} \theta_k = 0 \qquad \text{and} \qquad \sum_{k:\ a_{sk}=1,\ a_{tk}=1} \theta_k = 1,$$

i.e., rows s and t are covered by different columns in the first (left) case and by the same column in the second (right) case. In our application each row corresponds to a patient, so in the first branch we force patients s and t to be assigned to different stages, by introducing the following constraint into the pricing problem:

$$\sum_{k=1}^{\lambda} \zeta_s^k + \sum_{k=1}^{\lambda} \zeta_t^k \leq 1. \tag{4.68}$$

We refer to constraint (4.68) as the *decoupling* branching constraint. Similarly, we introduce a *coupling* branching constraint as:

$$\sum_{k=1}^{\lambda} \zeta_s^k = \sum_{k=1}^{\lambda} \zeta_t^k, \tag{4.69}$$

into the pricing problem on the other branch, where (4.69) indicates that patients t and s should be assigned to the same stage. We refer to this specialized branching rule as *Branching on Patient-Pairs*. Since the number of patients is finite, branch and price using this branching scheme terminates finitely.

4.4.2.2 Improving Column Generation.

Often, during initial column generation iterations, large values of dual variables can negatively impact solution times of the subproblems. To avoid this, we initialize the restricted master problem with a set of columns that is sufficient to obtain a feasible solution. The convergence rate of the branch-and-price algorithm can be further improved, if the set of the initial columns is comprised of those that are likely to be in the final optimal set. Therefore, we propose to generate this set heuristically using a greedy approach.

Input: κ
Output: $\mathcal{R}$
set $\mathcal{R} = \emptyset$
foreach $\lambda = \kappa, ..., 2$ **do**
  foreach $i \in \mathcal{N}$ **do**
    set $\mathcal{S} \leftarrow \mathcal{N}$ and $\mathcal{R}^{i,\lambda} = \emptyset$, let $j = 1$
    $\mathcal{L}^j = \lambda(2c^r \ell_{0i} + c^h d_i)$
    $D^j = d_i$
    set $\mathcal{S} \leftarrow \mathcal{S} \setminus \{k^j\}$, $\mathcal{R}^i \leftarrow \mathcal{R}^{j,\lambda} \bigcup \{k^j\}$, $k^j \leftarrow i$
    while $j \leq \lambda$ *and* $\mathcal{S} \neq \emptyset$ **do**
      for $k \in \mathcal{S}$ **do**
        if $D^j + d_k > L$ **then**
          $\mathcal{S} \leftarrow \mathcal{S} \setminus \{k\}$
      end
      if $\mathcal{S} \neq \emptyset$ **then**
        $k^{j+1} \in \text{argmin}_{k \in \mathcal{S}} \{2(\lambda - j)c^r \ell_{kk^j} + \lambda c^h d_k\}$
        $\mathcal{L}^{j+1} = \mathcal{L}^j + 2(\lambda - j)c^r \ell_{k^{j+1}k^j} + \lambda c^h d_{k^{j+1}}$
        $D^{j+1} = D^j + d_{k^{j+1}}$
        $\mathcal{S} \leftarrow \mathcal{S} \setminus \{k^{j+1}\}$, $\mathcal{R}^{i,\lambda} \leftarrow \mathcal{R}^{i,\lambda} \bigcup \{k^{j+1}\}$
        $k^j \leftarrow k^{j+1}$, $j \leftarrow j + 1$
    end
    if $|\mathcal{R}^{i,\lambda}| = \lambda$ **then**
      $\mathcal{R} \leftarrow \mathcal{R} \bigcup \{\mathcal{R}^{i,\lambda}, \mathcal{L}^\lambda\}$
  end
end

Algorithm 1: Greedy Heuristic Algorithm for Initial Transportation Route Generation

The proposed heuristic identifies a set of stage assignments consisting of at most κ patients and their transportation routes (the drop-off sequence), denoted by $\mathcal{R} \subseteq R$. Specifically, the algorithm iteratively constructs a sequence of $\lambda \leq \kappa$ patients, $\mathcal{R}^{i,\lambda}$, starting by first dropping off patient i. When constructing each $\mathcal{R}^{i,\lambda}$, the set of patients who have not been included in $\mathcal{R}^{i,\lambda}$ so far, are denoted by $\mathcal{S}$.

Input: $\kappa, \pi, \mu, \mathcal{N}_p^c = \{(u_1, v_1), ..., (u_{n_p^c}, v_{n_p^c})\}, \mathcal{N}_p^d = \{(\bar{u}_1, \bar{v}_1), ..., (\bar{u}_{n_p^d}, \bar{v}_{n_p^d})\}$
Output: $\mathcal{R}$

set $\mathcal{R} = \emptyset$
foreach $\lambda = \kappa, ..., 2$ **do**
  foreach $i \in \mathcal{N}$, $\mathcal{S} \leftarrow \mathcal{N}$, $j = 1$ *and* $\mathcal{R}^{i,\lambda} = \emptyset$ **do**
    $\mathcal{L}^j = \lambda(2c^r \ell_{0i} + c^h d_i) - \pi_i$, $D^j = d_i$
    $k^j \leftarrow i$, $\mathcal{S} \leftarrow \mathcal{S} \setminus \{k^j\}$, $\mathcal{R}^i \leftarrow \mathcal{R}^{j,\lambda} \bigcup \{k^j\}$
    while $j \leq \lambda$ *and* $\mathcal{S} \neq \emptyset$ **do**
      for $m = 1, ..., n_p^d$ **do**
        if $k^j = \bar{u}_m$ **then**
          $\mathcal{S} \leftarrow \mathcal{S} \setminus \{\bar{v}_m\}$
        if $k^j = \bar{v}_m$ **then**
          $\mathcal{S} \leftarrow \mathcal{S} \setminus \{\bar{u}_m\}$
      end
      for $k \in \mathcal{S}$ **do**
        if $D^j + d_k > L$ **then**
          $\mathcal{S} \leftarrow \mathcal{S} \setminus \{k\}$
      end
      if $\mathcal{S} \neq \emptyset$ **then**
        determine $k^{j+1} \in \text{argmin}_{k \in \mathcal{S}} \{2(\lambda - j)c^r \ell_{kk^j} + \lambda c^h d_k - \pi_k\}$
        $\mathcal{L}^{j+1} = \mathcal{L}^j + 2(\lambda - j)c^r \ell_{k^{j+1}k^j} + \lambda c^h d_{k^{j+1}} - \pi_{k^{j+1}}$
        $D^{j+1} = D^j + d_{k^{j+1}}$
        set $\mathcal{S} \leftarrow \mathcal{S} \setminus \{k^{j+1}\}$, $\mathcal{R}^i \leftarrow \mathcal{R}^{i,\lambda} \bigcup \{k^{j+1}\}$
        $k^j \leftarrow k^{j+1}$, $j \leftarrow j + 1$
    end
    if $|\mathcal{R}^{i,\lambda}| = \lambda$ **then**
      set $feasible = True$
      for $m = 1, ..., n_p^c$ **do**
        if $u_m \in \mathcal{R}^{i,\lambda}$ *and* $v_m \notin \mathcal{R}^{i,\lambda}$ **then**
          $feasible \leftarrow False$
        if $v_m \in \mathcal{R}^{i,\lambda}$ *and* $u_m \notin \mathcal{R}^{i,\lambda}$ **then**
          $feasible \leftarrow False$
      end
      if $\mathcal{L}^\lambda - \mu < 0$ *and* $feasible = True$ **then**
        set $\mathcal{R} \leftarrow \mathcal{R} \bigcup \mathcal{R}^{i,\lambda}$
  end
end

Algorithm 2: Greedy Heuristic Algorithm for the Pricing Problem

Let k^j be the jth patient in $\mathcal{R}^{i,\lambda}$. The algorithm selects a patient k^{j+1} to be visited next from the set of unvisited patients $\mathcal{S}$, in order to minimize the term $2(\lambda - j)c^r \ell_{k^{j+1}k^j} + \lambda c^h d_{k^{j+1}}$, which is consistent with the objective function $\mathcal{Z}_\lambda(\cdot)$ in the pricing problem. Furthermore, condition $D^j + d_{k^{j+1}} \leq L$ imposes the overtime restriction on the sum of surgery durations assigned to the stage when selecting the next patient to visit at iteration j. This process continues until the number of visited patients equals a desired length of the route λ, providing the estimated cost of the route, $\mathcal{L}^\lambda$. These routes are used to populate the initial master problem at the root node.

When generating a feasible column at the column generation stage, one might consider generating multiple columns at once [10]. Sol [107] shows that multiple column generation can be efficient for the set-partitioning problems. Moreover, usually there are more than one column with negative reduced cost at the column generation stage (especially in initial stages), so it may be advantageous to generated some subset of columns heuristically without solving the exact pricing problem, see [36]. To do so, we modify Algorithm 1 to accommodate the dual variables and the branching information available at each node of the branch-and-price tree. However, if the heuristic approach (Algorithm 2) fails to identify appropriate columns with negative reduced cost (i.e., $\mathcal{R} = \emptyset$), we resort to solving the exact pricing problem to ensure the correctness of the overall solution method.

In Algorithm 2, define $\mathcal{N}_p^c = \{(u_1, v_1), ..., (u_{n_p^c}, v_{n_p^c})\}$ and $\mathcal{N}_p^d = \{(\bar{u}_1, \bar{v}_1), ..., (\bar{u}_{n_p^d}, \bar{v}_{n_p^d})\}$ to be the coupling and decoupling constraints at node p of the branch-and-bound tree, respectively. In these sets each coupling pair (u_i, v_i), $i = 1, ..., n_p^c$ corresponds to patients u_i and v_i that are enforced to be at the same stage, while each decoupling pair $(\bar{u}_j, \bar{v}_j)$, $j = 1, ..., n_p^d$ indicates that patients $\bar{u}_j$ and $\bar{v}_j$ should be scheduled at different stages. Algorithm 2 is a greedy algorithm that iteratively constructs a feasible transportation route, consisting of a desired number of patients λ that: (i) retains the branching rule restrictions, (ii) satisfies the no-overtime constraint (4.37) for the given stage, and (iii) calculates the proper modified cost using the dual information. To achieve (i), we dynamically modify the set of feasible unvisited patients $\mathcal{S}$ with respect to decoupling branches whenever a patient is selected. The information about the active coupling branches is exploited when a complete route is constructed at the last step of the inner loop in the algorithm. Similar to our strategy in

Algorithm 1, (ii) is verified by checking the corresponding constraint whenever a new patient is considered. Lastly, (iii) is achieved by considering the dual prices as penalty costs when selecting any particular patient.

4.5 COMPUTATIONAL STUDY

In this section, we first describe test instances considered in our study, followed by the results of our computational experiments. The focus is on evaluating the performance of the developed solution techniques and highlighting the value of integrated surgery scheduling and vehicle routing decisions. We implement our branch-and-price algorithm using BCP, a framework for branch, cut, and price algorithm [96]. All computational experiments are conducted on an Intel Xeon PC with 3 GHz CPU and 3 GB of RAM.

4.5.1 Test Instances

We use the data provided by a VHA hospital in Pittsburgh which includes durations and turnover times for all surgery cases performed between 2006 to 2009. However, this data set does not include patients' location information for privacy reasons. To alleviate this issue, we generate synthetic patients by sampling a residence in Western Pennsylvania from the publicly available data on the veteran population provided by the VHA ([34]), and the operational data of surgery durations. The surgery durations are point estimates of each surgery requests that include the pre- and post-incision times (see [106] for data description). We chose ophthalmology procedures, as these are typically outpatient procedures.

Table 4 provides the characteristics of these problem instances. To be consistent with the historical records on the average number of surgeries in the surgery data set, we restrict our problem instances to those comprised of 4 to 6 possible surgeries to be scheduled at each stage. Throughout our experiments, we consider equal routing and hospital time costs, i.e., $c^r = c^h$.

Table 4: Test instances for computational study

Instance Class	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{K}\rvert$	κ
P1	16	4	4
P2	20	4	5
P3	24	4	6
P4	20	5	4
P5	25	5	5
P6	30	5	6
P7	24	6	4
P8	30	6	5
P9	36	6	6

4.5.2 Solution Method Performance

In Table 5, we report the performance of our branch-and-price algorithm against CPLEX 12.4 for 18 problem instances. Our computational results demonstrate that CPLEX 12.4 can not solve all selected instances within a 3-hour time limit except for those problems comprised of at most 16 patients, 4 time stages and $\kappa = 4$. However, the branch-and-price algorithm solves all instances within 4 minutes.

Next, we evaluate the effect of different computational strategies (discussed in Section 4.4.2.2) on the performance of our branch-and-price algorithm. We consider four algorithmic strategies summarized in Table 6. Each strategy is characterized by the number of columns introduced into the restricted master problem at each column generation iteration (# Columns), and the solution technique used for solving the pricing problem. Under the "Direct" strategy, the pricing problem is solved exactly using CPLEX. Under the "Heuristic" strategy, we first employ a heuristic approach (Algorithm 2); however, we resort to using CPLEX whenever the heuristic method fails. We further consider either single or multiple column generation combined with "Direct" and "Heuristic" approaches. Under the "Single"

Table 5: Performance of the developed branch-and-price approach compared to CPLEX.

Stages	Capacity	Patients	Instance	Branch-and-Price (sec.)	CPLEX (sec.)	CPLEX Optimality Gap (%)
3	4	12	1	1.14	13.37	0.00
			2	1.60	12.37	0.00
			3	0.37	14.31	0.00
3	5	15	1	11.43	151.96	0.00
			2	37.65	143.36	0.00
			3	9.30	144.1	0.00
3	6	18	1	13.09	3038.93	0.00
			2	11.32	1682.19	0.00
			3	245.25	5593.52	0.00
4	4	16	1	1.94	2410.89	0.00
			2	25.72	10645.76	0.00
			3	5.65	6677.41	0.00
4	5	20	1	7.59	8391.75	0.00
			2	7.82	> 10800	29.27
			3	58.98	> 10800	16.87
4	6	24	1	132.76	> 10800	37.99
			2	51.59	> 10800	38.60
			3	64.97	> 10800	29.35

method, a single column with the most negative reduced cost is introduced to the restricted master problem at each iteration, while for the "Multiple" approach, we add a set of columns with negative reduced costs.

In Table 7 we report results for five randomly generated instances of each problem class (Table 4) under each possible strategy. These algorithmic methods are compared based on the solution time (in seconds), the number of explored nodes and the depth of the resulting search trees as shown in Table 7.

Table 6: Description of column generation strategy.

Strategy	# of Columns	Pricing Strategy
SCD	Single	Direct
MCD	Multiple	Direct
SCH	Single	Heuristic
MCH	Multiple	Heuristic

Our computational results show that the solution time is rather sensitive to the algorithmic strategies. As expected, the numerical results also indicate that embedding a heuristic approach for the column generation significantly reduces the overall solution time (SCH and

Table 7: Performance of the branch-and-price method under different algorithmic strategies. The reported solution times are in seconds.

Class	Instance	SCD			MCD			SCH			MCH		
		Nodes	Depth	Time	Nodes	Depth	Time	Nodes	Depth	Time	Nodes	Depth	Time
P1	1	1	1	1.95	1	1	3.47	1	1	3.99	**1**	**1**	**0.85**
	2	32	7	25.57	10	4	12.63	**17**	**5**	**9.61**	21	8	9.87
	3	5	3	5.70	13	7	20.65	7	4	5.54	**5**	**3**	**4.06**
	4	3	2	1.43	3	2	2.08	5	3	3.46	**3**	**2**	**0.70**
	5	1	1	2.11	1	1	5.99	1	1	5.23	**1**	**1**	**1.70**
P2	1	1	1	15.791	1	1	10.35	1	1	6.63	**1**	**1**	**4.38**
	2	27	7	121.93	23	7	58.89	**27**	**7**	**44.42**	65	9	184.37
	3	35	11	153.227	33	7	100.98	**25**	**6**	**41.56**	29	7	84.92
	4	1	1	21.14	1	1	13.15	1	1	9.39	**1**	**1**	**7.94**
	5	1	1	18.04	1	1	9.24	**1**	**1**	**7.38**	1	1	8.15
P3	1	23	7	273.26	**15**	**5**	**151.63**	35	9	200.50	19	7	180.34
	2	16	5	247.18	21	6	231.58	27	7	187.64	**13**	**5**	**138.38**
	3	1	1	34.29	1	1	32.14	1	1	12.52	**1**	**1**	**10.21**
	4	9	4	164.24	13	5	139.10	**9**	**4**	**44.84**	13	5	133.17
	5	61	11	555.59	37	8	275.01	**53**	**9**	**183.46**	49	11	341.22
P4	1	1	1	2.61	1	1	4.42	1	1	3.97	**1**	**1**	**2.06**
	2	75	9	50.84	131	12	129.39	95	10	59.38	**91**	**8**	**50.61**
	3	49	7	35.83	79	14	82.79	**47**	**14**	**21.58**	42	11	26.80
	4	9	5	8.71	43	9	53.29	**15**	**7**	**13.77**	73	10	46.86
	5	55	11	44.72	39	11	64.57	47	10	45.04	**49**	**13**	**21.78**
P5	1	1	1	20.78	1	1	15.92	1	1	7.29	**1**	**1**	**6.26**
	2	1	1	31.19	1	1	24.22	1	1	12.28	**1**	**1**	**12.10**
	3	1	1	18.62	1	1	15.16	1	1	8.19	**1**	**1**	**7.34**
	4	83	12	269.55	215	13	675.60	**33**	**7**	**51.68**	57	7	142.77
	5	267	16	985.47	233	17	816.57	253	17	474.07	**91**	**11**	**271.28**
P6	1	41	7	438.07	55	9	815.41	73	10	522.48	**47**	**8**	**447.29**
	2	93	13	732.345	29	8	334.04	**61**	**9**	**331.95**	93	13	589.90
	3	27	9	242.91	47	8	415.68	28	9	202.65	**21**	**6**	**189.98**
	4	93	10	951.64	75	9	1028.71	**87**	**9**	**588.34**	73	10	621.13
	5	43	8	313.62	179	17	2047.19	49	10	274.96	**27**	**7**	**135.10**
P7	1	45	7	31.37	54	8	75.20	57	8	36.35	**43**	**7**	**26.41**
	2	1	1	3.64	1	1	8.07	1	1	5.27	**1**	**1**	**1.36**
	3	5	3	6.37	5	3	13.16	5	3	7.29	**5**	**3**	**3.32**
	4	5	3	4.30	7	4	12.27	7	4	7.99	**5**	**3**	**2.78**
	5	231	11	177.82	253	11	328.88	**145**	**10**	**96.20**	177	11	117.95
P8	1	443	15	1857.66	453	19	2393.41	**350**	**21**	**1087.66**	391	14	1492.91
	2	3	2	26.49	11	4	70.34	11	6	32.39	**5**	**3**	**24.48**
	3	133	14	570.90	67	11	310.63	**61**	**8**	**160.67**	231	17	952.05
	4	109	12	489.55	55	8	327.99	**49**	**9**	**153.84**	84	10	397.89
	5	1	1	25.16	1	1	39.04	1	1	22.43	**1**	**1**	**14.11**
P9	1	1	1	51.16	1	1	94.92	1	1	29.51	**1**	**1**	**17.93**
	2	5	3	73.06	5	3	164.48	5	3	50.239	**5**	**3**	**44.25**
	3	19	6	133.932	13	4	178.53	17	5	103.98	**7**	**4**	**44.93**
	4	52	11	449.28	69	8	1007.76	129	10	1095.67	**43**	**8**	**376.70**
	5	240	17	1661.62	312	30	1780.65	350	22	1855.94	**230**	**21**	**1459.78**

Table 8: Comparing objective function values of BSPRP when using exact, TSP- and TRP-based methods.

Problem	Instance	Optimal	TSP-based			TRP-based		
			Min	Max	Avg Gap (%)	Min	Max	Avg. Gap (%)
P1	1	4414	5050	5614	21	4976	5570	19
	2	4924	5286	5728	11	5268	5640	9
	3	4776	5092	5784	13	5090	5610	10
	4	5116	5780	6280	17	5604	5912	14
	5	4442	4866	5294	15	4862	5218	13
P2	1	7083	7961	8557	18	7631	8391	14
	2	7662	8612	9596	18	8352	9046	14
	3	8704	9930	10554	19	9428	10338	15
	4	7242	7706	8650	15	7692	8360	11
	5	8147	9229	9875	17	9045	9577	13
P3	1	10726	12036	12888	16	11966	12260	13
	2	11590	12884	13572	14	12840	13262	12
	3	11368	12674	13404	16	12440	13070	13
	4	9688	10746	11540	15	10620	11114	12
	5	11388	12888	14400	18	12796	13230	14
P4	1	5378	6202	6824	21	6116	6492	17
	2	6718	7222	7828	11	7194	7672	10
	3	6066	6956	7564	19	6802	7336	16
	4	6418	7324	8242	21	7290	7918	17
	5	6320	7166	7912	19	7102	7740	17
P5	1	10017	11185	12119	16	11005	11585	13
	2	9793	10937	12105	19	10887	11703	15
	3	9326	10774	12018	23	10476	11296	17
	4	10404	11642	12770	18	11474	12316	15
	5	9945	11771	12185	21	11505	12049	17
P6	1	12846	14706	15600	17	14328	14862	13
	2	12392	13930	14768	15	13470	14116	12
	3	13368	14902	16700	19	14780	15746	14
	4	13520	14778	16568	18	14632	15712	13
	5	14432	17270	18694	25	16650	17692	19
P7	1	7196	8568	9052	22	8360	8712	19
	2	7514	8584	9194	19	8416	8872	15
	3	7228	8528	9390	23	8206	9078	20
	4	7022	7790	8350	15	7736	8158	13
	5	7352	8452	9548	19	8338	8926	17
P8	1	11290	13158	14304	23	13038	13962	19
	2	10529	11921	12855	18	11797	12401	15
	3	11860	13756	15756	25	13602	14710	20
	4	12805	14919	15859	20	14503	15253	16
	5	11150	13440	14106	23	12778	13500	18
P9	1	15816	18430	19762	21	18126	19012	18
	2	15504	17326	18760	16	17102	17954	13
	3	16036	18610	20482	22	18274	19378	17
	4	16064	18324	19410	17	17742	18514	14
	5	14726	16734	18214	20	16490	17198	15

MCH columns), compared to solving the pricing problem exactly. Moreover, we observe that adding multiple columns at each column generation iteration does not necessarily reduce the solution time when compared to a single column approach. However, for the majority of test instances the multiple column generation scheme outperforms the single column generation approach.

4.5.3 Value of Integrating Surgery Scheduling and Vehicle Routing Decisions

In this study we model the integration of surgery scheduling and transportation in order to minimize the total service time cost of the patients. As emphasized earlier, such decisions are typically made independently. For example, in practice surgeries are often assigned to ORs based on some scheduling rule (e.g., first-fit) depending on the OR availability upon the arrival time of the surgery requests and disregarding patient transportation considerations. Subsequently, given the obtained surgery schedule the vehicle routing decisions are made separately, possibly using another optimization approach.

To evaluate the value of integrated decision-making framework, we compare our framework to two heuristic methods, where scheduling and routing decisions are performed sequentially. Under both heuristic methods, given a batch of surgery requests we randomly assign them to the time stages satisfying stage and vehicle capacity constraints. Having assigned all the surgeries to the stages, we exploit the following two routing methods to determine the pick-up and drop-off schedule for the patients at each stage. First, we consider a traveling salesman problem based (TSP-based) approach, where the objective is to determine a set of minimum length tours with respect to the residential locations of the patients at each stage. Second, we consider a traveling repairman problem based (TRP-based) approach that minimizes the sum of the times needed to visit patients locations. The pick-up routes in the TRP-based method are considered to be in the reverse order of the drop-off routes.

In Table 8, we compare the objective function values in BSPRP obtained using solutions of these two heuristics to those obtained by the branch-and-price method. For each problem instance (five in each problem class) the results are reported for 10 randomly generated surgery-to-time stage assignments. When compared to the both heuristic approaches,

the integrated scheduling framework provides a considerable improvement in service costs, ranging between 9% to 25%. These observations show the value of integrating the routing and scheduling decisions and highlight the fact that healthcare providers can substantially improve the quality of their services by considering scheduling and transportation decisions simultaneously. Note also that the TRP-based method performs better than the TSP-based one, which is not surprising as the TSP-based method considers the vehicle travel time instead of the total patients travel time.

4.6 CONCLUSION

We propose an integrated approach that simultaneously considers surgery scheduling and vehicle routing decisions for a given set of elective outpatient surgery requests using available ORs in a hospital. The overall objective is to minimize the total service cost that incorporates transportation and hospital times for all patients. Our main focus is on the special case of the problem. By exploiting the structure of the problem, we develop a branch-and-price algorithm, which is further enhanced with several algorithmic strategies to improve the overall solution efficiency. We test our approach using the historical data from the Veterans Affairs Pittsburgh Healthcare System. The results demonstrate that healthcare providers can substantially improve the quality of care by integrating scheduling and transportation decisions. We view this chapter as the first step in this direction. Future research should relax some of our assumptions, in particular, more general settings with uncertain surgery durations and availability of multiple vehicles.

5.0 CONCLUSIONS AND FUTURE RESEARCH

Rising healthcare expenditures necessitate efficient and equatable health delivery practices. In this dissertation, we aim to improve two healthcare delivery systems: health reporting requirements in liver allocation mechanism and integrating patient transportation and surgery scheduling decisions. First, we resolve to mitigate the inequity in liver allocation due to the information asymmetry between the UNOS and the patients in health reporting requirements. Second, we consider an operational problem faced by healthcare providers when patients transportation decisions are made in coordination with medical appointment scheduling decisions, more specifically surgery requests. Our proposed decision models are practical and quantify the value of integrating decisions in managing complex systems.

5.1 BALANCING EQUITY AND EFFICIENCY IN LIVER ALLOCATION VIA REVISED HEALTH REPORTING FREQUENCIES

The existing literature on organ allocation, more specifically those addressing liver allocation mechanism are mostly based on either a patient's or the societal perspective. However, designing an efficient and equatable organ allocation system needs a joint patient and societal perspective to achieve desired levels. Moreover, these design decision are often made in the face of uncertainty in patient population, donated organs, and usually consider multiple conflicting objectives. In Chapter 3, we propose a multi-objective optimization model to mitigate the existing information asymmetry in liver allocation system due to patients ability to game the system by exploiting the flexibility in health reporting requirements. To do so, we first extend the prior work on patient decision process to accept an organ offer/update health

status or do nothing and formulate this problem as an MDP model. Focusing on a patient's perspective, this model provides optimal accept/update/ do nothing policies to maximize the patient's life expectancy under any updating scheme. Second, under a multi-objective optimization framework and using stochastic programming techniques, we propose a model to determine an updating scheme with minimum expected system inequity as determined by optimal patients' policies while ensuring a minimum desired level of reduction in the data processing burden. Several structural properties of this stochastic programming model and the patients' optimal policies which enable design of efficient solution method are investigated. The proposed stochastic programming model is embedded in an iterative procedure to approximate the efficient frontier of the updating schemes to any given degree of accuracy. Our extensive numerical study using the clinical data shows that a typical patient can exploit the flexibility in the current updating requirements and increase his/her life expectancy up to 1% while the data processing burden can be decreased by up to 46.06%, compared to an updating scheme with perfect information on patient's true health status. We provide a menu of updating schemes that dominate the current updating scheme with respect to an increase in system inequity, a decrease in the data processing burden or both metrics. Our proposed menu captures the uncertainty in the cohort of patients on the liver waiting list in the long run. In future work, our proposed model can be extended in a number of ways. First, a more realistic patient decision-making problem can be achieved by relaxing the assumption that patients are aware of their true health status. This model can be formulated as a partially observable Markov Decision Process (POMDP) [94]. Second, the patient's rank information on the waiting list can also be incorporated into the state space description of the MDP model. Also, the proposed modeling framework can be used to design other organ allocation systems where patients can potentially game the system due to the information asymmetry between the UNOS and the patients. As an example lung transplantation is a potential area of similar research where the candidates on the waiting list are required to update their lung allocation scores at least once every six months [52].

5.2 THE SURGICAL PATIENT ROUTING PROBLEM

Transportation is commonly known as a major barrier to access timely and effective healthcare, especially in rural areas or for those with special needs. Hence, several organizations and medical centers provide transportation services to eligible patients. In Chapter 4, we introduce an integrated model that simultaneously consider the transportation decisions of a set of elective outpatient surgery requests and the surgery scheduling decisions in a hospital. We first propose a general model to determine the sequence of the surgeries in the available ORs as well as patients' transportation plan using a fleet of vehicles in the hospital. The planning horizon of the model consists of a number of stages and the model objective is to minimize the total travel time cost and the waiting time cost at the hospital for all the patients. This model is formulated as a mixed-integer programming problem. Second, we focus our attention on a computationally tractable problem where there is a single vehicle transports the patients at each stage. This model best serves the need of speciality or low-volume hospitals where the number of surgeries performed at each day is not high. Due to the complexity of solving the proposed model using the commercial solvers, we propose a set-partitioning based formulation of the special case model using its structural properties. A branch-and-price algorithm is proposed to solve the set-partitioning model and several algorithmic strategies to enhance the performance of the proposed solution method are discussed. The performance of the proposed branch-and-price method is evaluated through extensive numerical study using clinical data. We show that the healthcare providers can substantially improve the quality of their service by integrating the patient transportation and surgery appointment scheduling decisions. For future work, one can consider the uncertainty in the surgery durations that will both effect the surgery scheduling and the transportation decisions. Another direction of future work is to consider other performance measures such as the OR utilization, OR opening cost or surgery cancelation cost.

BIBLIOGRAPHY

[1] S. Ahmed and N. V. Sahinidis. An approximation scheme for stochastic integer programs arising in capacity expansion. *Operations Research*, 51(3):461–471, 2003.

[2] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Englewood Cliffs, NJ, 1993.

[3] O. Alagoz, C. L. Bryce, S. M. Shechter, A. J. Schaefer, C.-C. H. Chang, D. C. Angus, and M. S. Roberts. Incorporating biological natural history in simulation models: Empiric estimates of the progression of end-stage liver disease. *Medical Decision Making*, 25(6):620–632, 2005.

[4] O. Alagoz, L. M. Maillart, A. J. Schaefer, and M. S. Roberts. The optimal timing of living-donor liver transplantation. *Management Science*, 50(10):1420–1430, 2004.

[5] O. Alagoz, L. M. Maillart, A. J. Schaefer, and M. S. Roberts. Choosing among living-donor and cadaveric livers. *Management Science*, 53(11):1702–1715, 2007.

[6] O. Alagoz, L. M. Maillart, A. J. Schaefer, and M. S. Roberts. Determining the acceptance of cadaveric livers using an implicit model of the waiting list. *Operations Research*, 55(1):24–36, 2007.

[7] T. A. Arcury, J. S. Preisser, W. M. Gesler, and J. M. Powers. Access to transportation and health care utilization in a rural region. *Journal of Rural Health*, 21(1):31–38, 2005.

[8] M. D. Baily, S. M. Shechter, and A. J. Schaefer. SPAR: Stochastic programming with adversarial recourse. *Operations Research Letters*, 34:307–315, 2006.

[9] C. Barnhart, A. Cohn, E. L. Johnson, D. Klabjan, G. L. Nemhauser, and P. H. Vance. Airline crew scheduling. In R.W. Hall, editor, *Handbook of Transportation Science*. Kluwer Academic Publishers, Norwell, MA, 2nd edition, 2002.

[10] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998.

[11] H. Bayrak. *Detection Resource Allocation In Adversarial Problems*. PhD thesis, University of Pittsburgh, 2007.

[12] E. M. L. Beale. On minimizing a convex function subject to linear inequalities. *Journal of the Royal Statistical Society Series B*, 17:173–184, 1955.

[13] R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.

[14] J. F. Benders. Partitioning procedures for solving mixed variables programming problems. *Numerische Mathematics*, 4:238–252, 1962.

[15] D. P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, Belmont, MA, 1995.

[16] D. Bertsimas, V. F. Farias, and N. Trichakis. Fairness, efficiency, and flexibility in organ allocation for kidney transplantation. *Operations Research*, 61(1):73–87, 2013.

[17] L. P. Bigras, M. Gamache, and G. Savard. The time-dependent traveling salesman problem and single machine scheduling problems with sequence dependent setup times. *Discrete Optimization.*, 5(4):685–699, 2008.

[18] J. R. Birge and F. V. Louveaux. *Introduction to Stochastic Programming*. Springer, New York, 1997.

[19] D. Blackwell. Discounted dynamic programming. *Annals of Mathematical Statistics*, 36(1):226–235, 1965.

[20] J. T. Blake and M. W. Carter. Surgical process scheduling: A structured review. *Journal of the Society for Health Systems*, 5(3):17–30, 1997.

[21] C. C. Branas, E. J. MacKenzie, and C. S. ReVelle. A trauma resource allocation model for ambulances and hospitals. *Health Services Research*, 35(2):489–507, 2000.

[22] L. Brotcorne, G. Laporte, and F. Semet. Ambulance location and relocation models. *European Journal of Operational Research*, 147(3):451–463, 2003.

[23] J. Burkhardt. *TCRP Research Report 144: Sharing the costs of human services transportation*. Transportation Research Board, The National Academies., Washington, DC., 2011.

[24] J. Burkhardt, J. M. Rubino, and J. Yum. *TCRP Research Results Digest 99: Improving mobility for veterans*. Transportation Research Board, The National Academies., Washington, DC., 2011.

[25] B. Cardoen, E. Demeulemeester, and J. Belin. Operating room planning and scheduling: A literature review. *European Journal of Operational Research.*, 201(3):921–932, 2010.

[26] V. Chankong and Y. Y. Haimes. *Multiobjective Decision Making Theory and Methodology*. Elsevier Science, New York, 1983.

[27] A. Charnes and W. W. Cooper. *Management Models and Industrial Applications of Linear Programming*. John Wiley and Sons, New York, 1961.

[28] C. Chow and J. Tsitsiklis. The complexity of dynamic programming. *Journal of Complexity*, 5:466–488, 1989.

[29] Mayo Clinic. Available from http://www.mayoclinic.com/. last accessed October 2013.

[30] K. J. Cormicon, D. P. Morton, and K. R. Wood. Stochastic network interdiction. *Operations Research*, 46(2):184–197, 1998.

[31] G. B. Dantzig. Linear programming under uncertainty. *Management Science*, 1(3):197–206, 1955.

[32] G. B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8(1):101–111, 1960.

[33] M.C. Demirci. *Designing the Liver Allocation Hierarchy: Incorporating Equity and Uncertainty.* PhD thesis, University of Pittsburgh, 2008.

[34] Department of Veterans Affairs. The Veteran Population Model (VetPop2007), Accessed 2012. Available from http://www.va.gov/vetdata.

[35] F. D'Epenoux. A probabilistic production and inventory problem. *Management Science*, 10(1):98–108, 1963.

[36] G. Desaulniers. Branch-and-cut-and-price for split-delivery vehicle routing problem with time windows. *Operations Research*, 58(1):179–192, 2010.

[37] M. Desrochers, J. Desrosiers, and M. M. Solomon. A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research*, 40(2):342–354, 1992.

[38] M. Desrochers and F. Soumis. A column generation approach to the urban transit crew scheduling problem. *Transportation Science*, 23(1):1–13, 1989.

[39] F. Dexter, 2013. Available from http://www.FranklinDexter.net/; last accessed October 2013.

[40] F. Dexter, A. Macario, R. D. Traub, M. Hopwood, and D. A. Lubarsky. An operating room scheduling strategy to maximize the use of operating room block time: Computer simulation of patient scheduling and survey of patients' preferences for surgical waiting time. *Anesthesia & Analgesia*, 89(1):2–20, 1999.

[41] F. Dexter, R. D. Traub, and A. Macario. How to release allocated operating room time to increase effciency: Predicting which surgical service will have the most underutilized operating room time. *Anesthesia & Analgesia*, 96(2):507–512, 2003.

[42] K. Doerner, A. Focke, and W. J. Gutjahr. Multicriteria tour planning for mobile healthcare facilities in a developing country. *European Journal of Operational Research*, 179(3):1078–1096, 2007.

[43] Y. Dumas, J. Desrosiers, and F. Soumis. The pickup and delivery problem with time windows. *European Journal of Operational Research*, 54(1):7–22, 1991.

[44] M. Ehrgott. *Multicriteria Optimization*. Springer, Berlin, 2005.

[45] M. Ehrgott. A discussion of scalarization techniques for multiple objective integer programming. *Annals of Operations Research*, 147(1):343–360, 2006.

[46] S. A. Erdogan and B. T. Denton. Surgery planning and scheduling. J. J. Cochran, L. A. Cox Jr., P. Keskinocak, J. P. Kharoufeh, J. C. Smith, eds. 2011.

[47] H. Fei, N. Meskens, and C. Chu. A planning and scheduling problem for an operating theatre using an open scheduling strategy. *Computers and Industrial Engineering*, 58(2):221–230, 2010.

[48] J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer-Verlag, New York, 1997.

[49] M. Fischetti, G. Laporte, and S. Martello. The delivery man problem and cumulative matroids. *Operations Research*, 41(6):1055–1064, 1993.

[50] Center for Disease Control and Prevention. Available from http://www.cdc.gov/. last accessed October 2013.

[51] Centers for Medicare and Medicaid Services. Available from https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trendsand-Reports/NationalHealthExpendData/downloads/proj2010.pdf. Accessed July 17, 2013.

[52] United Network for Organ Sharing. Available from http://www.unos.org.

[53] L. R. Ford and D. R. Fulkerson. A suggested computation for maximal multicommodity network flows. *Management Science*, 5(1):97–101, 1958.

[54] K. R. Fox, B. Gavish, and S. C. Graves. An n-constraint formulation of the (time-dependent) traveling salesman problem. *Operations Research*, 28(4):1018–1021, 1980.

[55] P. C. Gilmore and R. E. Gomory. A linear programming approach to the cutting stock problem. *Operations Research*, 11(6):863–888, 1962.

[56] D. Gupta. Surgical suites operations management. *Production Operations Management*, 16(6):689–700, 2007.

[57] D. Gupta and B. T. Denton. Appointment scheduling in health care: Challenges and opportunities. *IIE Transactions.*, 40(9):800–819, 2008.

[58] G. Heilporn, J. F. Cordeau, and G. Laporte. The delivery man problem with time windows. *Discrete Optimization*, 7(4):269–282, 2010.

[59] J. Hornberger and J. H. Ahn. Deciding eligibility for transplantation when a donor kidney becomes available. *Medical Decision Making*, 17(2):160–170, 1997.

[60] D. H. Howard. Why do transplant surgeons turn down organs? A model of the accept/reject decision. *Journal of Health Economics*, 21(6):957–969, 2002.

[61] R. Howard. *Dynamic Programming and Markov Processes.* MIT Press, Cambridge, 1960.

[62] Z. G. Içten. *Markov Decision Process Models for Improving Equity in Liver Allocation.* PhD thesis, University of Pittsburgh, 2011.

[63] P. Kall and S. W. Wallace. *Stochastic Programming.* John Wiley & Sons, Chichester, U. K., 1994.

[64] E. P. C. Kao and M. Queyranne. Budgeting costs of nursing in a hospital. *Management Science*, 31(5):608–621, 1985.

[65] A. B. Keha, K. Khowala, and J. W. Fowler. Mixed integer programming formulations for single machine scheduling problems. *Computers & Industrial Engineering*, 56(1):357–367, 2009.

[66] A. J. King. Stochastic programming at IBM's T.J. Watson Research Center. *EKKNEWS newsletter*, 1995. Available from http://www.cs.cmu.edu/˜ guyb/real-world/linear/ibm.html.

[67] A. J. Kleywegt, A. Shapiro, and T. Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479502, 2002.

[68] N. Kong. *Optimizing the Efficiency of the United States Organ Allocation System through Region Reorganization.* PhD thesis, University of Pittsburgh, Department of Industrial Engineering, 2006.

[69] N. Kong, A. J. Schaefer, B. K. Hunsaker, and M. S. Roberts. Maximizing the efficiency of the U.S. liver allocation system through region design. *Management Science*, 56(12):2111–2122, 2010.

[70] M. Kurt. *Dynamic Decision Models for Managing the Major Complications of Diabetes.* PhD thesis, University of Pittsburgh, 2012.

[71] G. Laporte. The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(3):345–358, 1992.

[72] G. Laporte and F. V. Louveaux. The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, 13(3):133–142, 1993.

[73] G. Laporte, F. V. Louveaux, and L. Van Hamme. An integer L-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research*, 50(3):415–423, 2002.

[74] E. K. Lee, T. Fox, and I. Crocker. Optimization of radiosurgery treatment planning via mixed-integer programming. *Medical Physics*, 27(5):995–1004, 2000.

[75] E. K. Lee and M. Zaider. Mixed integer programming approaches to treatment planning for brachytherapy - application to permanent prostate implants. *Annals of Operations Research*, 119(1-4):147–163, 2003.

[76] J. Linderoth and S. Wright. Decomposition algorithms for stochastic programming on a computational grid. *Computational Optimization and Applications*, 24(2-3):207–250, 2003.

[77] M. E. Lübbecke and J. Desrosiers. Selected topics in column generation. *Operations Research*, 53(6):1007–1023, 2005.

[78] A. Lucena. Time-dependent traveling salesman problem: The delivery-man case. *Networks*, 20(6):753–763, 1990.

[79] G. Lyberatos and E. M. Abulesz. Analysis of periodic injections in cancer chemotherapy. *International Journal of Systems Science*, 21(8):1659–1671, 1990.

[80] M. Malinchoc, P. S. Kamath, F. D. Gordon, C. J. Peine, J. Rank, and P. C. J. ter Borg. A Model to predict poor survival in patients undergoing transjugular intrahepatic portosystemic shunts. *Hepatology*, 31(4):864–871, 2000.

[81] J. Mattson. Transportation, distance, and health care utilization for older adults in rural and small urban areas. *Transportation Research Record*, 22(65):192–199, 2010.

[82] A. Mehrotra, E. L. Johnson, and G. L. Nemhauser. An optimization based heuristic for political districting. *Management Science*, 44(8):1100–1114, 1998.

[83] H. E. Miller, W. P. Pierskalla, and G. J. Rath. Nurse scheduling using mathematical programming. *Operations Research*, 24(5):857–870, 1976.

[84] T. E. Morton. On the asymptotic convergence rate of cost differences for Markovian decision processes. *Operations Research*, 19(1):244–248, 1971.

[85] S. U. Ngueveu, R. Prins, and R. Wolfler-Calvo. An efficient memetic algorithm for the cumulative capacitated vehicle routing problem. *Computers & Operations Research*, 37(11):1877–1885, 2010.

[86] The University of Tennessee Medical Center (UTMC). Available from:http://www.utmedicalcenter.org/.

[87] C. A. Okoro, T. Strine, S. L. Young, and L. S. Balluz. Access to health care among older adults and receipt of preventative services. results from the behavioral risk factor surveillance system, 2002. *Preventative Medicine*, 40(2):337–343, 2005.

[88] O. Y. Özaltin, O. A. Prokopyev, A. J. Schaefer, and M. S. Roberts. Optimizing the societal benefits of the annual influenza vaccine: A stochastic programming approach. *Operations Research*, 59(5):1131–1143, 2011.

[89] V. Pareto. *Cours D'Economie Politique*. Libraire Drotz, Genve, 1964.

[90] J. Park and B. Kim. The school bus routing problem: A review. *European Journal of Operational Research*, 202(2):311–319, 2010.

[91] J. C. Picard and M. Queyranne. The time-dependent traveling salesman problem and its application to the tardiness problem in one-machine scheduling. *Operations Research*, 26(1):86–110, 1978.

[92] F. D. Preciado-Walters, R. L. Rardin, M. Langer, and V. Thai. A coupled column generation, mixed integer approach to optimal planning of intensity modulated radiation therapy for cancer. *Mathematical Programming*, 101(2):319–338, 2004.

[93] The Organ Procurement and Transplantation Network (OPTN). Available from http://optn.transplant.hrsa.gov/.

[94] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, New York, NY, 1994.

[95] M. Queyranne and A.S. Schulz. Polyhedral approaches to machine scheduling. Technical Report 408, Department of Mathematics, Technical University of Berlin., 1994.

[96] T. K. Ralphs and L. Ladanyi. COIN-OR: Software tools for implementing custom solvers., 2003. Accessed June, 2012, http://www.coin-or .org/Presentations/INFORMSWorkshop03.pdf.

[97] K. J. Rask, M. V. Williams, and R. M. Parker. Obstacles predicting lack of regular provider and delays in seeking care for patients at an urban public hospital. *Journal of the American Medical Association.*, 274(24):1931–1933, 1994.

[98] C. Ribeiro and F. Soumis. A column generation approach to the multiple depot vehicle scheduling problem. *Operations Research*, 42(1):41–53, 1994.

[99] G. M. Ribeiro and G. Laporte. An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. *Computers & Operations Research*, 39(3):728–735, 2012.

[100] J. P. Roberts, B. Nikolai, and S. Tomlanovic. Cost of organ procurement and transplantation network data collection for a large transplant center. *American Journal of Transplantation*, (3):1316–1317, 2003.

[101] M. S. Roberts, D. C. Angus, L. Weissfeld, C. L. Bryce, and Z. Valenta. Survival after liver transplantation in the United States: A disease-specific analysis of the UNOS database. *Liver Transplantation*, 10(7):886–897, 2004.

[102] D. M. Ryan and B. A. Foster. An integer programming approach to scheduling. In A. Wren, editor, *Computer Scheduling of Public Transport Urban Passenger Vehicle and Crew Scheduling*, pages 269–280. North-Holland, 1981.

[103] B. Sandikçi. *Estimating the Price of Privacy in Liver Transplantation.* PhD thesis, University of Pittsburgh, 2008.

[104] B. Sandikçi, O. Alagoz, L. M. Maillart, A. J. Schaefer, and M. S. Roberts. Estimating the patient's price of privacy in liver transplantation. *Operations Research*, 56(6):1393–1410, 2008.

[105] B. Sandikçi, L. M. Maillart, A. J. Schaefer, and M. S. Roberts. Alleviating the patient's price of privacy through a partially observable waiting list. *Management Science*, 59(8):1836–1854.

[106] O. V. Shylo, O. A. Prokopyev, and A. J. Schaefer. Stochastic operating room scheduling for high-volume specialties under block booking. *INFORMS Journal on Computing*, 25(4):682–692, 2013.

[107] M. Sol. *Column Generation Techniques for Pick up and Delivery Problems.* Ph.D. thesis, Eindhoven University of Technology, Eindhoven, The Netherlands., 1994.

[108] M. M. Solomon. Algorithms for the vehicle routing and scheduling problem with time window constraints. *Operations Research*, 35(2):254–265, 1987.

[109] X. Su and S. A. Zenios. Patient choice in kidney allocation: The role of the queueing discipline. *Manufacturing & Service Operations Management*, 6(4):280–301, 2004.

[110] X. Su and S. A. Zenios. Patient choice in kidney allocation: A sequential stochastic assignment model. *Operations Research*, 53(3):443–455, 2005.

[111] X. Su and S. A. Zenios. Recipient choice can address the efficiency-equity trade-off in kidney transplantation: A mechanism design model. *Management Science*, 52(11):1647–1660, 2006.

[112] The Henry J Kaiser Family Foundation. Available from http://www.kff.org/.

[113] P. Toth and D. Vigo. *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications., SIAM, Philadelphia., 2002.

[114] R. Van Slyke and R. J-B. Wets. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17(4):638–663, 1969.

[115] D. M. Warner. Scheduling nursing personnel according to nursing preference: A mathematical programming approach. *Operations Research*, 24(5):842–856, 1976.

[116] World Health Organization (WHO). WHO Statistical Information System. World Health Statistics, 2011. Available from http://www.who.int/whosis/whostat/2011/en/index.html. Accessed July 10, 2013.

[117] R. H. Wiesner, S. V. McDiarmid, P. S. Kamath, E. B. Edwards, M. Malinchoc, W. J. Kremers, R. A. F. Krom, and W. R. Kim. MELD and PELD: Application of survival models to liver allocation. *Liver Transplantation*, 7(7):567–580, 2001.

[118] S. Yoo, M. E. Kowalok, B. R. Thomasden, and D. L. Henderson. Treatment planning for prostate brachytherapy using region of interest adjoint functions and a greedy heuristic. *Physics in Medicine and Biology*, 48(24):4077–4090, 2003.