TWO-STAGE ROBUST OPTIMIZATION WITH APPLICATIONS IN HEALTH CARE AND COMBINATORIAL OPTIMIZATION

by

Saba Neyshabouri

A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
in Partial Fulfillment of
the Requirements for the Degree
of
Doctor of Philosophy
Systems Engineering and Operations Research

Committee:

_____	Dr. Karla Hoffman, Dissertation Director
_____	Dr. Bjorn Berg, Co-Director
_____	Dr. John Shortle, Committee Member
_____	Dr. Fei Li, Committee Member
_____	Dr. Ariela Sofer, Committee Member
_____	Dr. Ariela Sofer, Department Chair
_____	Dr. Kenneth S. Ball, Dean, Volgenau School of Engineering
Date: _____	Fall Semester 2016 George Mason University Fairfax, VA

Two-Stage Robust Optimization with Applications in Health Care and Combinatorial Optimization

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

By

Saba Neyshabouri
Master of Science
George Mason University, 2013
Bachelor of Science
Sharif University of Technology, 2010

Director: Karla Hoffman, Professor
Co-Director: Bjorn Berg, PhD
Department of Systems Engineering and Operations Research

Fall Semester 2016
George Mason University
Fairfax, VA

Dedication

To my lovely parents who made this possible, Safoora, and Ahmad.

Acknowledgments

This dissertation would have not been completed without the guidance, help, and support I received from my mentors, friends, and my family.

I was very fortunate to have Dr. Bjorn Berg as my adviser for this research. Dr. Berg offered me his guidance and time throughout my research, keeping me positive through tough times. This dissertation would not have been complete without his patience and encouragement.

Many thanks goes to the chair of my committee and my academic mentor, Dr. Karla Hoffman. She has played an inspiring role and provided me with a strong backing since the very first day I joined the Ph.D. program at GMU. She taught me the importance of pursuing my interests regardless of how challenging the road may be.

I thank my committee members Dr. Ariela Sofer, Dr. John Shortle, and Dr. Fei Li for their constant guidance throughout my research, which pushed me to set higher standards for my work.

My special thanks to Dr. Lance Sherry for showing me the ropes to the world of research and supporting my decisions throughout my studies.

I would not be able to do any of this work if it was not for the great teachers I have had here at GMU. Dr. Stephen Nash, Dr. Rajesh Ganesan, and Dr. Roman Polyak motivated me to teach and I am grateful for that.

I have been very fortunate to form great friendships at Mason. I thank Ehsan Kouroshfar, Nariman Mirzaei, Pouyan Ahmadi, Ryan O'neal, and Ankit Shah for their constant support during my time here at Mason. I thank my best friend, Zoya Farooq, for being there for me through tough times and encouraging me to continue.

I owe all of my achievements to my family. My two older brothers Emad and Aref never stopped encouraging me and have been a constant source of support to me and my family in my absence. My sister, Safa, has been my best friend and her love has motivated me to continue. I am grateful for my father, Ahmad, for trusting my every decision and supporting me. My mother, Safoora, has been my greatest source of inspiration. None of this would be possible if it was not for her infinite love and selflessness towards me and my family. I am forever grateful for all she has sacrificed for me.

Table of Contents

List of Tables

List of Figures

Abstract

TWO-STAGE ROBUST OPTIMIZATION WITH APPLICATIONS IN HEALTH CARE AND COMBINATORIAL OPTIMIZATION

Saba Neyshabouri, PhD

George Mason University, 2016

Dissertation Director: Dr. Karla Hoffman, Dr. Bjorn Berg

The development of new robust optimization models is motivated by the need for risk-based decision making in health care operations. Surgery scheduling has attracted a great deal of attention due to its importance in health care outcomes and costs. We apply robust optimization theory to the surgery scheduling problem and downstream capacity planning problem to address important questions regarding the impact of uncertainty in surgery duration and length-of-stay (LOS) in the surgical intensive care units on hospital resource planning and scheduling operations.

In this dissertation we focus our research on decision making under uncertainty using the framework of two-stage robust optimization. We develop exact solution methods for optimization problems that include binary variables. We contribute to the theory of robust optimization by addressing special cases when the uncertainty is discrete in nature and depends how decisions made in one period will impact outcomes in future periods.

We propose a novel two-stage robust optimization formulation that models the

discrete nature of the patient's LOS using a column-and-constraint generation approach that is new to the literature. The proposed algorithm successfully handles both the discrete nature of the uncertainty and multi-stage impact. We apply the approach to surgery scheduling where the availability of downstream critical care facilities can seriously impact patient outcomes. Our computational tests show that this methodology can solve, or provide high quality solutions, to realistic problem instances in reasonable time (one hour). The solution structure provides decision makers with insights on the underlying trade-offs between operational performance of the system, such as patient throughput and risk metrics for patients.

Motivated by applications in health care and surgery scheduling, we study a more general problem in greater detail, the robust generalized assignment problem (GAP). The robust GAP is of great importance and appears in many other application domains such as modeling the supply chain and scheduling the manufacturing of multiple items. Often, these decisions are made without concern for the impacts of incorrect specification and variability of the data.

We develop a two-stage robust formulation for the generalized assignment problem where the resource requirements for job-resource pairs are uncertain. We study the structure of this problem and improve the existing solution methods by strengthening the constraints that are passed to the master problem.

While models similar to the robust GAP have received previous attention in the literature, we provide a counter-example for a compact formulation that has been proposed to obtain exact optimal solutions. We improve upon that work by proposing valid inequalities and improve the quality of the solutions obtained by that method.

Our computational study of the robust GAP shows our proposed method can handle instances of medium size problems in reasonable amounts of time (less than an hour). Using simulation we provide better understanding of the impact of uncertainty

and robustness in resource allocation and its implications on load allocations. Solutions tend to distribute *risky* jobs among resources to reduce the chance of going over capacity, which in turn implies fairness in the allocation of loads.

Chapter 1: Introduction

The development of the field of robust optimization models is motivated by the need for risk-based decision making when the impact of the incorrect specification of data can significantly impact the overall success of the endeavor. Health care operations is one area where worst-case scenarios must be considered because the impact of an incorrect decision can be life-threatening. We apply robust optimization theory to the surgery scheduling problem and consider the downstream capacity of the intensive care facilities in order to assure that patients needing these critical care facilities receive them for the period of time required for a proper recovery. Thus, the problem addressed is the important one of how to model the trade-offs between maximizing the use of expensive surgical facilities and the availability of downstream intensive care facilities when there is uncertainty in both surgery duration and in the resultant length-of-stay (LOS) in the surgical intensive care units (SICU). To our knowledge, this is the first attempt at considering these downstream when optimizing surgical schedules using robust optimization.

To solve this problem, our research focuses on decision-making under uncertainty using the framework of two-stage robust optimization. We develop exact solution methods for optimization problems that include binary variables. We also contribute to the theory of robust optimization by addressing special cases when uncertainty is discrete in nature and when decisions impact the definition of uncertainty.

We propose a novel two-stage robust optimization formulation that models the discrete nature of the LOS for patients staying in the SICU using an adapted column-and-constraint generation approach that is new to the literature. The proposed algorithm

successfully handles both the discrete nature of the uncertainty and its dependence on decisions. Our computational tests show that this methodology can solve realistic problem instances of surgical scheduling problems in reasonable amounts of time (an hour). The solution structure provides decision makers with insights on the underlying trade-offs between operational performance of the system, such as patient throughput, and risk metrics for patients.

Motivated by applications in health care and surgery scheduling, we study a more general problem in greater detail, the robust generalized assignment problem (GAP) which has multiple applications.

We first provide an overview of the surgical scheduling problem, we then introduce our robust optimization approach to solving that problem and finally take the results of this modeling effort and show they can be applied to the generalized assignment problem, a problem often encouraged in manufacturing scheduling, inventory optimization, and supply chain management

1.1 Surgery Scheduling

Decision-making in health care has become a very important problem and is being studied extensively. The U.S. health care system is dealing with the major issues of associated skyrocketing costs and patient concerns about outcomes. As in every developed country, health care costs are rising faster than the GDP and there are quality problems such as medical errors, and prevalent overuse and underuse of resources [Green, 2012]. For example, the cost of healthcare in the U.S. in 2011 was over 17.6% of the national gross domestic product; equal to over 2.9 trillion dollars [Martin et al., 2011]. This highlights the importance of improving the efficiency of the health care system and how even minor systematic improvements can translate

into great savings.

Hospital operations, and surgery departments in particular, are important elements in the health care system and offer life-saving services. Surgical suites' operations consume around 10% of hospital's budget. In addition, deferrable surgery procedures may account for up to 52% of all hospital admissions [Gupta, 2007]. In terms of costs, surgeries account for more than 40% of a hospital's total revenues and expenses [Erdogan et al., 2011]. In fact, the operating cost of a surgery department is approximately one-third of the total operating costs of the hospital [Macario et al., 1995]. Other sources mention that surgeries account for approximately two-thirds of the hospital revenues [Jackson, 2002]. This highlights the importance of effective management of surgical suits in hospitals.

An important aspect of high quality health care delivery in a surgery department is the assignment of appropriate post-operative care which is usually provided by specialized units such as Post-Anesthesia Care Unit (PACU), Intensive Care Unit (ICU), or Surgical Intensive Care Unit (SICU). To show the importance of these downstream resources, [Jonnalagadda et al., 2005] show that 15% of the total surgery cancellation is due to the lack of an available recovery bed in the hospital they studied. Similarly, [Sobolev et al., 2005] show that the Length-Of-Stay (LOS) in the ICU and the bed availability in the ICU affect the surgery schedule. This is mainly due to the existence of uncertainty in patients' LOS in such units and limited capacity of aforementioned units.

While a large portion of the literature in surgery scheduling focuses on strategic and tactical planning of surgeries, operational decisions that are made over a short period of time have not received the same attention. Studying and proposing methods to include inherent sources of uncertainty, that may not conform to well-known probability distributions due to the short term decision period, poses great challenges

in planning.

This dissertation applies Robust Optimization (RO) to this context. RO proves to be a flexible method in modeling such uncertainties and allows us to propose plans that address costs and risks simultaneously. This allows decision makers to study the trade-offs between cost and risk and choose plans that conform to their cost-risk preferences.

1.2 Decision-Making Under Uncertainty

Stochastic programming (SP) theory, extends the deterministic optimization techniques to be applied when some or all of the parameters in an optimization model are random and belong to specific distributions. [Birge and Louveaux, 2011] provide a through introduction to the theory and various modeling techniques used in stochastic programming.

Robust optimization (RO) is another methodology designed to address the existence of uncertainty in all or a subset of parameters of an optimization model. The most basic form of a RO model does not assume any distributional information and only assumes that uncertain parameters belong to a known (deterministic) set known as the *uncertainty set*, which makes it inherently different than the SP theory. RO aims to find the best set of decisions with respect to the worst-case realization of the uncertainty. For more in depth treatment on RO theory readers are referred to [Ben-Tal et al., 2009] and references therein.

Recently, the idea of distributionally robust optimization (DRO) is proposed to bridge the gap between the SP and RO theories. DRO assumes that the uncertain parameters in the optimization problem come from a distribution that is unknown to us. Therefore it assumes a set of possible distributions which is called an *ambiguity*

set and optimizes with respect to the worst-case distribution [Bertsimas et al., 2011].

Not considering uncertainty and randomness in optimization applications can generate solutions that are highly sub-optimal and sometimes infeasible to the problem when the data of the problem takes on values considerably different from those assumed in the deterministic formulation. There are numerous application areas where both SP and RO are used to address the issue of randomness and uncertainty.

In this dissertation, we turn our focus on two-stage robust optimization where decisions are broken into two sets. *Here-and-now* decisions that have to be made prior to the realization of uncertainty, and *wait-and-see* decisions which are made after uncertainty is realized, as a corrective action.

The two-stage setting is of particular importance since it allows us to construct a mathematical model for decision making processes. In many cased we do not know the value of parameters with certainty and only have knowldege of the possible values for these parameters. Two-stage SP has been extensively studied and applied to model problems in this setting [Birge and Louveaux, 2011]. Two-stage RO (2RO) is relatively new and has only recently been applied to far fewer applications. While RO offers a natural framework for risk-averse decision making, its use of uncertainty sets to model the uncertain parameters can offer more flexible ways to integrate uncertainty into the decision making process that is suitable for data-driven approaches.

Traditionally, uncertainty sets are modeled as convex sets which has allowed for use of efficient methods and approximations for solving RO and 2RO problems. In reality, uncertainty may not present itself in such form. We present an application where uncertain parameters are discrete and cannot be modeled as convex set.

In addition, due to the complexity of two-stage RO, the literature is heavily focused on finding high quality approximations for these problems [Bertsimas et al.,

2011]. While this is an important stream of research, we believe increased computational power can allow us to employ exact solution methods to solve problem instances of real size. Thus it is important to study and improve exact solutions methods for such problems.

[Wallace and Ziemba, 2005] list applications such as fleet management, production planning and scheduling, supply chain optimization, network resource utilization, and unit commitment problem to be a few of the numerous applications where SP methodology is employed to address the stochastic nature of the problem.

[Gabrel et al., 2014b] present many of the recent applications of RO in decision making problems. To name a few areas where RO is being used extensively applied, the authors count inventory and logistics optimization, facility location, finance, and revenue management. However, few of these applications consider downstream impacts. In this dissertation we focus on applications in health care, specifically in surgery scheduling and consider the downstream impact on ICU availability

1.3 The Generalized Assignment Problem

Finally, we study the characteristics of the generalized assignment problem. GAP tends to show itself in many important applications such as capacitated facility location and supply chain. In addition, it has been used in the surgery scheduling context for assigning surgeries to operating rooms with limited length for each shift. In many applications, such as in surgery scheduling, the resource requirement for a given job (surgery) is not exactly known. Uncertainty in resource requirements can change the way decisions are made by introducing the risk of not having enough capacity on a resource and requiring to purchase more capacity. The structure of the problem is similar to that presented earlier with a few minor changes.

Looking at the capacity of each of surgical facilities and their structures but not considering the downstream units, provides a general framework that applies to a much broader application domain: inventory management, manufacturing scheduling, supply chain distribution assignment, etc., all fall under this domain.

We provide new constraints that tighten the master problem formulation and show how these constraints impact both solution quality and computational speed.

1.4 Thesis Overview

In Chapter 2, we present an introductory overview of the topics in stochastic programming and robust optimization. Next, the literature on surgery scheduling and downstream units are covered. Finally, a brief introduction to the literature of the generalized assignment problem is provided.

Chapter 3 presents a detailed introduction and formulation of the integrated surgery scheduling problem. Using the theory of robust optimization, a two-stage robust formulation for the surgery scheduling and downstream capacity planning is presented. In our setting we consider the uncertainty in both surgery duration and length-of-stay (LOS) for patients. We present a novel formulation to capture the discrete nature of the uncertainty in LOS for patients. A detailed study of the structural properties for this problem allows us to modify and adapt the solution methods in the literature to find exact solutions for this problem. Computational simulation tests are performed to assess the performance of the proposed algorithm as well as the quality of the obtained solutions.

Chapter 4 considers the generalized assignment problem and presents a detailed study of the robust generalized assignment problem (RGAP). We formulate the

two-stage robust generalized assignment problem with resource requirement uncertainty. We propose solution methods based on cutting-plane method [Kelley, 1960] and column-and-constraint generation method [Zeng and Zhao, 2013]. We study the structural properties of the proposed formulation and present results to improve the performance of each solution method. In our investigation, we find a counter example to a formulation which is the only known compact formulation to the two-stage robust bin-packing problem [Denton et al., 2010] and show their formulation produces upper bounds on the problem, but does not guarantee optimal solutions. We propose valid inequalities that improve the results obtained from their formulation. Finally, we present our computational results from our proposed solution methodology and a simulation study to understand the implications of including uncertainty in the model.

Chapter 5 summarizes our contributions in this dissertation and outlines future research avenues.

Chapter 2: Literature Review

In this chapter we introduce a brief summary of decision-making under uncertainty. We provide an overview of the underlying theory of stochastic programming (SP) and robust optimization (RO). Next, we provide an overview of the literature in the applications that are going to be covered in this thesis is presented and what stochastic issues are important to these problem areas.

2.1 Stochastic Programming

Stochastic programming deals with decision-making problems in which all or a subset of the parameters are modeled as random variables. Therefore, there should be a distribution associated with the randam parameters in the problem. The aim is to optimize with respect to a measure that includes the randomness such as the expected value, the variance, or the probability distribution. For example, in a newsvendor problem, the goal is minimize the expected cost while the demand is random. Other problems aims to find a set of decisions that minimizes the total costs constrained to assure that certain constraints are met with high probability. For a detailed treatment and references on the topics in SP that are beyond the scope of this thesis, readers are refered to [Birge and Louveaux, 2011] and the references therein.

SP has been used in many recent applications such as inventory management [Küçükyavuz, 2011], supply chain management [Santoso et al., 2005], and disaster management [Rawls and Turnquist, 2010]. More applications are cited in [Wallace and Ziemba, 2005] and references therein.

2.1.1 Two-Stage Stochastic Programming

Decision-making problems can be broken down into two broad categories of *static* problems and *dynamic* problems. In static models, there is one set of decisions that has to be made prior to the realization of the random variables and the goal is to optimize a probabilistic measure such as expected value. These decisions are called *here-and-now* decisions which corresponds to the fact that they should be made before the uncertainty unravels. In static problems, there are no other decisions to be made.

To explain the case of dynamic decision-making problems, we focus on two-stage stochastic programming (2SSP) where the decisions are divided into two categories: (1) *here-and-now* or *first-stage* decisions which have to be made prior to the realization of the random parameters in the problem, (2) *wait-and-see* or *recourse* decisions which are the decisions that are made after the realization of the random parameters. Consider the case of newsvendor problem. The decision-maker has to make the first-stage decisions of how many papers to order before knowing the exact realization of the random parameter, demand. After first-stage decisions are made, random parameters materialize and the demand is known. In the second-stage the decision-maker (in our case the newsvendor) can take recourse actions such as selling the remaining papers at a lower price or incurring a penalty for unsatisfied demand.

In 2SSP, the goal is to minimize (maximize) the first-stage costs (profit) plus the expected value of recourse costs (profit). The 2SSP problem can be formulated as follows:

$$\min \quad c'x + E_\xi[\min \ \ q(\omega)'y(\omega)] \tag{2.1a}$$

$$s.t.$$

$$Ax = b \tag{2.1b}$$

$$Wy(\omega) + T(\omega)x = h(\omega) \tag{2.1c}$$

$$x \geq 0, y(\omega) \geq 0 \tag{2.1d}$$

where first stage-decisions are shown by vector x and second-stage decisions are based on the random parameters and shown by the vector $y(\omega)$. ξ is a random vector defined on the probability space, (Ω, Ξ, P), and A and W are known matrices of conforming sizes. W is called the recourse matrix. For each ω, $T(\omega), q(\omega)$, and $h(\omega)$ create the stochastic components of the problem, we obtain the vector $\xi(\omega) = (q(\omega), h(\omega), T_1(\omega), .., T_m(\omega))$, where $T_i(\omega)$ is the i-th row of $T(\omega)$. E_ξ represents the mathematical expectation with respect to ξ.

The objective (2.1a) is to minimize the first-stage costs as well as the expected cost of the second-stage. The first set of constraints (2.1b) represent the restrictions on the first-stage decision variables which are not dependent on the random variables. The second set of constraints (2.1c), captures the relation ship between the second-stage decisions y and the first-stage decisions x. It also shows that some of the parameters in the constraints for the second-stage are also dependent on the random variables.

In most cases of 2SSP, random parameters are represented by a finite number of possibilities, labeled as scenarios. Each scenario represents one realization of all the parameters that are random. In this way, the random variables are represented by

a discrete joint distribution and there is a probability associated with each scenario realization which makes the calculation of the expected value possible.

2.1.2 Solution Methods

Deterministic-Equivalent Formulation

There are multiple ways to solve a 2SSP problem. The most trivial way is to solve the *deterministic-equivalent* problem as a large-scale optimization problem. Assume set Ω is the set of all scenarios for the random parameters. The deterministic-equivalent formulation can be written as follows:

$$\min \quad c'x + \sum_{s \in \Omega} p_s q_s' y_s \tag{2.2a}$$

$$s.t.$$

$$Ax = b \tag{2.2b}$$

$$Wy_s + T_s x = h_s \qquad s \in \Omega \tag{2.2c}$$

$$x \geq 0, y_s \geq 0 \qquad s \in \Omega \tag{2.2d}$$

The objective (2.2a) minimizes the sum of first-stage costs as well as the expected cost of the second-stage. Note that p_s is the probability associated to scenario $s \in \Omega$. In addition for each scenario $s \in \Omega$ a set of recourse variables y_s is defined to measure the recourse decisions for each scenario. The constraints in the second-stage (2.2c) are also written for each scenario. This means that the restrictions in the second-stage have to be satisfied for every scenario s. As can be seen, this formulation makes a

copy of second-stage decisions for each scenario and adds $|\Omega|$ copies of the second-stage constraints to the formulation. In case that the number of scenarios are large, the formulation can grow very fast.

2.1.3 Decomposition-based Methods

There are multiple decomposition-based methods to address 2SSP problems that take advantage of the special structure of this class of problems. Direct decomposition methods such as cutting-plane methods aim to construct a sequence of approximations for the objective of the 2SSP by outer linearizing the recourse problem. Dual decomposition methods are another approach to solve multi-stage stochastic programming problems. Readers are referred to [Ruszczyński, 1997] for more detailed discussion of these solution methods.

Here we present the L-shaped method as an example of a solution algorithm for 2SSP. Thanks to the special structure of the 2SSP which has a block-diagonal coefficient matrix, decomposition-based methods can be employed to solve the 2SSP to optimality. Consider the following formulation:

$$\min \quad c'x + \mathcal{Q}(x) \tag{2.3a}$$

$$s.t.$$

$$Ax = b \tag{2.3b}$$

$$x \geq 0 \tag{2.3c}$$

Where $\mathcal{Q}(x) = E[Q(x, \xi_s)]$ in which $Q(x, \xi_s)$ can be written as the following optimization problem:

$$\min \quad q_s' y_s \tag{2.4a}$$

$$s.t.$$

$$W y_s + T_s x = h_s \tag{2.4b}$$

$$y_s \geq 0 \tag{2.4c}$$

Considering the structure of the 2SSP, [Van Slyke and Wets, 1969] proposed the L-shaped method which is based on Bender's decomposition introduced in [Benders, 1962].

Here we present the case with complete recourse, which means that the second-stage problem always has a feasible solution. The general steps of the L-shaped method can be summarized as follows:

- **Step 1-** Solve the master problem which corresponds to the first-stage formulation and obtain the optimal first-stage decisions x and lower bound on the optimal solution.
- **Step 2-** For each scenario $s \in \Omega$ solve the second-stage problem $Q(x, \xi_s)$ and obtain the optimal y_s and update the upper bound on the optimal solution.
- **Step 3-** Check for the optimality criteria:
  - **Step 3.1-** If optimal, return $x, y_s \, \forall s$ as the optimal solution.
  - **Step 3.2-** If not optimal, add an optimality cut to the master and go to Step 1.

Note that the L-shaped algorithm adds one constraint to the master problem at

each iteration of the algorithm. In the case that the second-stage problem is easy to solve, one can improve the performance by solving smaller and easier problems rather than one large-scale optimization.

2.1.4 Multicut L-Shaped Method

[Birge and Louveaux, 1988] introduced the multicut version of the L-shaped method which adds as many as $|\Omega|$ constraints to the master problem at each iteration. The authors prove that the multicut version has superior performance compared to the single cut version of the algorithm. The only downside to this method is that the number of constraints in the master problem grows faster than the single cut version.

There are many other methods to solve the 2SSP problem and its variants. Interested readers are referred to [Birge, 1997], [Birge and Louveaux, 2011], and [Shapiro et al., 2014] and references therein.

2.2 Robust Optimization

Robust optimization is a relatively new approach for decision making under uncertainty which was introduced by [El Ghaoui et al., 1998], [Ben-Tal and Nemirovski, 1998], and [Bertsimas and Sim, 2004]. Unlike the stochastic programming theory which assumes distributional information about uncertain parameters and aims to optimize an expectation measure [Birge and Louveaux, 2011], RO does not need distributional information and produces optimal solutions that are feasible for a defined set of values that uncertain parameters can take. In other words, RO seeks to optimize against the worst-case realization of uncertainty. It can also produce a probabilistic guarantee for the feasibility of the solutions. The conservatism of the solution can be controlled by the means of a defined *budget of uncertainty*. In cases

where obtaining probabilistic information is not possible or infeasibility cannot be tolerated, RO offers a flexible framework for producing good solutions. [Bertsimas et al., 2011] and [Gabrel et al., 2014b] provide surveys of the existing literature on the theory and applications of the robust optimization. [Ben-Tal et al., 2009] provides a comprehensive treatment for the general topics in RO.

RO has been successfully applied to various applications where traditional stochastic methods are not applicable. For example, the use of distributional information for short-term planning of surgeries may not provide appropriate solutions, while better results can be obtained by considering a data-driven approach that requires specific information for each patient. To better understand why distributional information can be misleading in short term planning, consider the case when all patients on a surgery list who need a specific type of surgery (for example heart surgery). In reality, there can be cases that for a week, the number of patients with complicated surgeries that require higher than average length-of-stays. While sampling from the distribution may not reflect on the cases that needs to be scheduled for that week. Using stochastic programming methods or sampling methods will not consider the fact that the week under study has this characteristic. RO on the other hand, can include case-specific information and facilitate such short-term planning complexities. Additionally, RO methodology provides means to obtain solutions that are protected against worst case realization of uncertainty. This characteristic is especially of great interest in the health care settings such as surgery planning, since the outcome of not considering these sources of uncertainty can have serious impacts on the health outcome of patients. Using RO enables the decision makers to provide efficient use of surgical facilities while ensuring against disastrous outcomes. Thus, this methodology is most useful where unlikely results can have extreme consequences.

2.2.1 Static Models

The origins of RO can be traced back to [Soyster, 1973] which considered column-wise uncertainty in the coefficients of the decision variables in the constraints. In his work, the aim was to optimize the decisions such that they are feasible for all realizations of uncertainty. In this article, each parameter could take on values belonging to a specific range which translates into box uncertainty.

Little work is done on this topic until the late 90's where [Ben-Tal and Nemirovski, 1998] introduces RO for general convex optimization problems. In this work they assumed that the uncertainty was enclosed within an ellipsoid. In [Ben-Tal and Nemirovski, 1999], RO for linear programming (LP) problems is introduced while the uncertainty is modeled to belong to an ellipsoidal set. [Ben-Tal and Nemirovski, 2000] study the impact of uncertainty in parameters in linear programming (LP) problems. Here is a generalized formulation for a static RO problem presented in [Ben-Tal et al., 2009]:

$$\min_{x}\{\sup_{(c,A,b)\in\mathcal{U}} c'x : Ax \leq b \; \forall (c,A,b) \in \mathcal{U}\}, \tag{2.5}$$

which can be reformulated as

$$\min_{x,t}\{t : c'x \leq t, Ax \leq b \; \forall (c,A,b) \in \mathcal{U}\}. \tag{2.6}$$

The formulation 2.6 is called the *Robust Counterpart* (RC) of the original problem. It is important to note that in this setting all decisions are *here-and-now* decisions and the goal is to find decisions that are feasible for all realizations of uncertainty. The robust counterpart of an LP with an ellipsoidal uncertainty can be reformulated and solved as a conic quadratic program, which has higher computational cost than solving an LP.

Later, [Bertsimas and Sim, 2004] introduced a modeling approach that is relying on the observation that, naturally not all the uncertain parameters simultaneously take on their worst-case values. The authors suggest defining the uncertainty to be row-wise, meaning that all the uncertain parameters belonging to a constraint are independent of the other constraints. In addition, the number of parameters that can deviate to their worst-case realization is constrained to be less than a predetermined *budget of uncertainty*. The budget of uncertainty can be used to control the level of conservatism in the RO. In addition, since the uncertainty set is in the form of a polytope, the RC of the LP remains an LP with additional variables and constraints.

In [Bertsimas and Sim, 2003] the authors apply the idea of RO with their proposed definition of an uncertainty set to a class of Integer Programs (IP) and network flow problems. They show that the robust counterpart of an IP remains an IP.

The resulting RC formulations can be solved using standard commercial solvers that are capable of handing structured convex optimization problems and integer programming problems.

2.2.2 Adaptive Models

Unlike static models, *adaptive* or *adjustable* models divide the decision-making process into multiple stages. The term "adjustable" is used to point out that decision-makers have the opportunity to adjust their decisions after the realization of uncertain parameters which is similar to recourse decisions. Similar to two-stage stochastic programs ([Birge and Louveaux, 2011]), two-stage (adaptive) robust optimization addresses the cases when the decisions can be split into two different sets. First, *here-and-now* decisions that have to be made prior to the realization of the uncertainty. After the uncertainty is realized, *wait-and-see* decisions are made. These are the recourse decisions that are made to correct for the impact of the uncertainty. In contrast to the

stochastic programming formulation where the realization of uncertainty is a result of a stochastic process (e.g., sample from a distribution, simulation), the realization of uncertainty in the RO methodology is a product of an optimization model.

[Ben-Tal et al., 2004] was the first to introduce the idea for adjustable robust solutions for LP. The authors show that adjustable robust counterparts (ARCs) generally provide solutions that are less conservative than the static model. On the other hand, while solving the static formulation is tractable, ARC is generally NP-hard.

[Atamtürk and Zhang, 2007] propose a two-stage robust optimization approach for a network design problem where the demand is uncertain. Theoretical complexity results and special cases are presented. The methodology is applied to various problems such as lot-sizing, and location-transportation.

[Thiele et al., 2009] present methods for robust LP with recourse when the uncertainty is in the right-hand-side of the recourse constraints. The authors propose a cutting-plane algorithm based on [Kelley, 1960] which operates similar to Bender's decomposition [Benders, 1962] and the L-shaped method [Van Slyke and Wets, 1969].

Here we present a general formulation for the two-stage robust optimization (2SRO):

$$\min_{\{x:Ax\geq b, x\subseteq\mathbb{R}^n_+\}} c^T x + \max_{u\in\mathcal{U}} \min_{y\in S(x,u)} d^T y \tag{2.7}$$

in which $S(x,u) = \{y : Wy \geq h - Ex - Mu, y \subseteq \mathbb{R}^m_+\}$. W, E and M are matrices of appropriate sizes. Vectors are denoted using lowercase letters and assumed to be of conforming dimensions. The first-stage decisions are captured by vector x and recourse decisions are shown using vector y. Note that the uncertainty is captured using the vector u that belongs to the uncertainty set $\mathcal{U}$. The set of feasible solutions for the second-stage depends on both first-stage decisions x and the realization of the uncertain parameter u and is defined by $S(x,u)$. The difference between 2SRO

and 2SSP is apparent in this formulation. In 2SSP we aim to minimize the expected cost for the second-stage. In 2SRO the expectation operator is switched with a maximization operator. Note that the expectation operator is a linear operator while the maximization is another optimization problem.

2.2.3 Solution Methods

Affinely Adjustable Robust Counterpart (AARC)

The idea for AARC was first proposed in [Ben-Tal et al., 2004]. After realizing that the ARC problems are, in general, NP-hard, the authors considered alternative approaches to make the problem tractable. They proposed a more restricted model for ARC formulations. In affinely adjustable robust formulations, it is assumed that the recourse (adjustable) decisions are restricted to be a function of uncertain parameters. More specifically, adjustable variables are restricted to be affine functions of the uncertain parameters. With this restriction, the authors show that the AARC formulation is indeed a large-scale LP and tractable. They apply this framework to an inventory problem and show that AARC outperforms the static formulation in terms of costs.

Finite Adaptability

The idea of *finite adaptability* was first proposed by [Bertsimas and Caramanis, 2010]. The general idea is to restrict the number of possible recourse decisions to be finite. In reality, each realization of uncertainty can have its own set of recourse decisions and for the case of convex uncertainty sets, this can translate into an infinite number of recourse decisions. Finite adaptability restricts the number of possible recourse

decisions, or adjustments, that can be made after the realization of uncertain parameters. The authors show that when recourse costs are known, finite adaptable models have better performances than the static models. In addition they can approximate the fully adaptable problem. In the fully adaptable problem, the decision-maker has a set of recourse decisions for each realization of uncertainty.

[Hanasusanto et al., 2015] extend the idea of finite adaptability to two-stage robust binary problems where the decision-maker pre-commits to K second-stage policies. In this case, the recourse decisions are binary variables.

Constraint Generation or Cutting Plane Methods

As mentioned before [Thiele et al., 2009] proposed the general two-stage robust LP with recourse and uncertainty in the right-hand-side. In order to be able to solve the tri-level optimization of $\min - \max - \min$ form, the authors rely on strong duality and reformulate the inner minimization problem into a maximization using its dual which creates one maximization in the second-stage with bilinear terms. In some cases, thanks to the structure of the uncertainty sets that are defined, one can reformulate the second-stage as a mixed-integer linear program. The second-stage problem serves as the adversarial problem since it aims to find the worst-case realization of uncertainty to incur the highest cost.

[Gabrel et al., 2014a] propose a similar approach for the two-stage robust location transportation problem when the demand is uncertain.

The overview of the proposed constraint generation algorithm is as follows:

- **Step 1-** Solve the master problem which corresponds to the first-stage formulation and obtain the optimal first-stage decisions x and lowerbound LB.
- **Step 2-** For the given first-stage decision x solve the adversarial second-stage

problem $\mathcal{Q}(x,\mathcal{U})$ and obtain the optimal recourse decision y as well as worst-case scenario for the uncertain parameter u and update the upperbound UB.

- **Step 3-** Check for the optimality criteria ($UB - LB \leq \epsilon$):
  - **Step 3.1-** If optimal, return x as the optimal solution.
  - **Step 3.2-** If not optimal, add an optimality cut to the master and go to Step 1.

As can be seen, the constraint generation method is very similar to the L-shaped method for the 2SSP problem, where a Bender's cut based on duality arguments is added to the problem.

Column-and-Constraint Generation Method (C&CG)

[Zeng and Zhao, 2013] propose a new algorithm to solve the 2SRO problem. The algorithm is based on the idea that if one could enumerate all the possible scenarios for the uncertain parameters, a large-scale deterministic equivalent formulation for the 2SRO can be written. In the fully adaptable case, each scenario requires its own set of recourse variables and second-stage constraints. Since the total enumeration of the scenarios can lead to an infinite number of scenarios, a decomposition-based approach is employed to identify the scenarios and add the relating variables and constraints in the form represented by the deterministic equivalent problem. They show that the worst-case performance of their algorithm is better than the constraint generation methods proposed by [Thiele et al., 2009] and [Gabrel et al., 2014a] and they provide computational results for location transportation problems.

Since we have employed a solution methodology that stems from the C&CG, we briefly explain how this algorithm works. [Zeng and Zhao, 2013] mention that the

2SRO problem 2.7 can be written as the following large scale deterministic-equivalent problem:

$$\min_{x} \quad c^T x + \theta \tag{2.8a}$$

$$s.t.$$

$$Ax \geq b \tag{2.8b}$$

$$\theta \geq d^T y^k \qquad k = 1, .., r \tag{2.8c}$$

$$Ex + Wy^k \geq h - Mu^k \qquad k = 1, .., r \tag{2.8d}$$

$$x \subseteq \mathbb{R}^n_+, y^k \subseteq \mathbb{R}^n_+ \qquad k = 1, .., r \tag{2.8e}$$

The recourse problem $\mathcal{Q}(x) = \{\max_{u \in \mathcal{U}} \min d^T y : Wy \geq h - Ex - Mu, y \subseteq \mathbb{R}^m_+\}$ identifies the worst-case scenario for the uncertain parameter u for a given first-stage decision x. Assuming that the second-stage problem has complete recourse, meaning that for every first-stage decision the recourse is feasible, and is bounded for all feasible first-stage decisions, the C&CG algorithm has the following structure:

- **Step 1-** Set $LB = -\infty, UB = +\infty, k = 0$, and $O = \emptyset$.

- **Step 2-** Solve the following master problem:

$$\min_{x} \quad c^T x + \theta \tag{2.9a}$$

$$s.t.$$

$$Ax \geq b \tag{2.9b}$$

$$\theta \geq d^T y^i \qquad \forall i \in O \tag{2.9c}$$

$$Ex + Wy^i \geq h - Mu^i \qquad \forall i \leq k \tag{2.9d}$$

$$x \subseteq \mathbb{R}^n_+, y^i \subseteq \mathbb{R}^n_+ \qquad \forall i \leq k \tag{2.9e}$$

  Obtain the optimal solution $(x^*_{k+1}, \theta^*_{k+1}, y^{1*}, .., y^{k*})$ and set $LB = c^T x^*_{k+1} + \theta^*_{k+1}$.

- **Step 3-** Solve the subproblem $\mathcal{Q}(x^*_{k+1})$ in and update $UB = \min\{UB, c^T x^*_{k+1} + \mathcal{Q}(x^*_{k+1})\}$.

- **Step 4-** If $UB - LB \leq \epsilon$, optimal solution is found, return x^*_{k+1} and terminate. Otherwise do

  - **Step 4.1-** Add the new variables y^{k+1} and the following constraints to the master problem:

$$\theta \geq d^T y^{k+1} \tag{2.10a}$$

$$Ex + Wy^{k+1} \geq h - Mu^{k+1} \tag{2.10b}$$

    where u^{k+1} is the optimal solution (worst-case scenario) obtained solving $\mathcal{Q}(x^*_{k+1})$. Update $k \leftarrow k+1, O \leftarrow O \cup \{k+1\}$ and go to Step 2.

2.3 Surgery and Downstream Capacity Planning

Surgery planning covers a variety of decision-making problems within the health care setting. Decision-making regarding surgery planning can be broken down and studied at three different levels:

- **Strategic Level-** Decisions at this level are aimed to identify the future demand and assign the resources such that the overall demand over a long period is satisfied. For example, the decision of what type of specialties to be offered is a strategic decision. This is also known as case mix planning which is usually done by the hospital leadership.

- **Tactical Level-** Decisions at this level aim to identify a master plan that assigns the surgery blocks to different specialties. These plans are generally cyclic. Minor changes occur throughout the planning horizon, which is in the order of a few months.

- **Operational Level-** Decisions at the operational level deal with day-to-day operations. For example, assigning the time and operating room (OR) for a specific patient is an operational decision.

In this research, we turn our focus to the operational decisions, and specifically those decisions concerned with elective patient scheduling.

Surgical suites' operations consume around 10% of hospital's budget [Gupta, 2007]. In addition, deferrable surgery procedures may account for up to 52% of all hospital admissions [Gupta, 2007]. This shows that efficient management of elective procedures can result in potentially large improvements in the overall performance of hospitals. Operating rooms (ORs) are one of the most expensive resources in

hospitals and require highly skilled staff, expensive resources, and sophisticated technologies. Up to 70% of all hospital admissions involve a stay in the OR department [van Oostrum et al., 2008]. In terms of costs, surgeries account for more than 40% of a hospital's total revenues and expenses [Erdogan et al., 2011]. In fact, the operating cost of a surgery department is approximately one-third of the total operating costs of the hospital [Macario et al., 1995]. On the other hand surgeries account for approximately two-thirds of the hospital revenues [Jackson, 2002].

The quality of care is an especially important factor in managing hospital operations. The inability to deliver high quality care can incur high costs and poor outcomes. In surgery departments, surgery cancellations result in prolonged stays, delayed preoperative treatments and repeated preoperative tests and treatments [Gul et al., 2012]. Cancellations have been found to incur a cost of $1700-$2000 per case [Argo et al., 2009]. The task of efficient surgery planning is complicated by multiple contributing factors such as uncertainty in surgery duration and limited available resources such as operating rooms, surgeons, and OR staff.

An important aspect of high quality health care delivery in a surgery department is the assignment of appropriate post-operative care which is usually provided by specialized units such as Post-Anesthesia Care Unit (PACU), Intensive Care Unit (ICU), or Surgical Intensive Care Unit (SICU). To show the importance of these downstream resources, [Jonnalagadda et al., 2005] shows that 15% of the total surgery cancellation is due to the lack of an available recovery bed in the hospital they studied. Similarly, [Sobolev et al., 2005] show that the Length-Of-Stay (LOS) in the ICU and bed availability in the ICU affect surgery schedules. This is mainly due to the uncertainty in patients' LOS in such units and limited capacity of the aforementioned units. In the case of lack of available capacity for patients in such units, the following policies can be employed:

- **Cancellation** of an already planned surgery which leads to cancellation costs as well as patient discomfort.

- **Premature discharge** or **transfer** of a patient from one of these care units in order to free a bed for another patient.

Each policy affects the system differently. While surgery cancellation costs are estimated in previous studies, it is difficult to quantify the value of patient discomfort. On the other hand, [Utzolino et al., 2010] show that the readmission rate to the SICU for patients with unplanned discharge from the SICU was 25.1% which is almost four times that of those who where discharged electively (8.3%). They also show that the mortality rate for patients who are readmitted to the SICU (13.3%) is almost six times higher than those who are not readmitted (2.28%). These statistics illustrate the need for careful consideration of downstream capacity when determining a surgical schedule. It is important to mention that it is very difficult to quantify the costs related to mentioned risks.

This research focuses on the decision process of assigning elective surgery patients to available surgery blocks under the *block scheduling* policy. We assume that emergency patients have a specialized unit allocated to them and we do not include the emergency surgeries in our study.

Under the *block scheduling* framework, operating room schedules are divided into multiple blocks of defined lengths and each block is assigned to a surgical team or a specific specialty (e.g., Cardiology, ENT, Neurology, etc.). Each specialty is allowed to schedule surgeries in their allocated block. Usually, surgery blocks are planned to be cyclical that repeats itself on a weekly or biweekly basis.

The problem of planning operating/surgery room operations has been well studied in the literature in different categories such as block scheduling, capacity planning,

and surgery sequencing to name a few. Readers are encouraged to refer to [Cardoen et al., 2010], [Gupta, 2007], [Ferrand et al., 2014], [Guerriero and Guido, 2011], [Demeulemeester et al., 2013] for in-depth reviews of the literature related to multiple problems addressed in the previous research efforts.

Thanks to the existing extensive and recent surveys on this topic, we turn our focus to the articles that are recent and are closely related to our subject. [Hsu et al., 2003] set to minimize the number of nurses in the PACU (Post Anesthesia Care Unit) by determining the surgery sequences in a single day setting. They formulate the problem as a deterministic no-wait, two-stage process shop scheduling problem and solve it using a tabu search-based algorithm. Although the proposed algorithm is shown to be effective in finding near optimal solutions, uncertainty is not addressed in their setting. As shown by [Marcon and Dexter, 2006] through a discrete event simulation, different surgery sequencing policies have significant impact on the congestion and resource requirements in the PACU. [Gupta, 2007] employs a dynamic programming formulation for the elective surgery booking problem. While downstream resources are considered in this model, the multi-period nature of the demand for downstream resources has not been addressed. In fact the author states that "a tractable model of the surgery booking control problem is difficult to formulate because, following surgery, patients may require care for several days in a downstream unit and the lengths of stay are not known with certainty. Thus, each booked surgery consumes an unknown and discrete chunk of the downstream unit's resources."

[Bam et al., 2015] provide a mixed-integer linear program (MILP) for surgery scheduling considering PACU resources. They show that the problem is hard to solve using general solvers. They propose a method to generate the parameters for surgery duration and LOS in the PACU such that they are hedged against the uncertainty. Next they propose a two-step heuristic that first assigns patients to ORs and then

finds a sequence to satisfy the limited resources in the PACU. Simulation results show that the proposed solutions from the heuristic are robust to uncertainty in the parameters.

In reality, the LOS for patients in the ICU or SICU can be longer than one day and is not deterministic. [Truong et al., 2013] uses dynamic programming to obtain optimal policies while considering the multi-period demand for downstream units. The decision is to identify the number of elective patients to serve. The authors show that localized decision rules with the focus on a single unit in the hospital can result in up to 60% higher costs. While this formulation includes uncertainty in LOS and emergency arrivals, the model does not explicitly consider the individual capacity requirements for each surgery block at surgery stage.

[Pham and Klinkert, 2008] and [Fei et al., 2008] consider the surgical planning problem with deterministic durations. [Lamiri et al., 2009] propose a stochastic optimization approach for surgery planning problem where arrivals for emergency patients are random. However, the surgery duration is deterministic.

There are multiple sources of uncertainty in surgery planning and decision making under uncertainty is a much more difficult task. For example, uncertain surgery duration, which is case-dependent, can be a cause for overtime. Emergency arrivals to the operating rooms can lead into disruption of the planned schedule. Finally, uncertain LOS in the SICU/ICU can cause cancellation in surgeries and early discharges due to a lack available SICU beds.

Stochastic programming (SP) and robust optimization (RO) techniques have been used to address the uncertainty in surgery durations. [Denton et al., 2010] provide a two-stage stochastic program as well as a robust optimization model to obtain the optimal assignment of surgery blocks to operating rooms. They show that the value of the stochastic solution is highest when the overtime costs are high. They also

show that their robust formulation provides high quality solutions quickly. [Deng et al., 2014] propose a chance-constrained programming and a distributionally robust model to the surgery planning problem under uncertain surgery durations. Their formulation finds the optimal operating rooms to open, as well as the assignment and sequencing of surgeries to the ORs. [Gul et al., 2012] propose a multi-stage mixed-integer programming approach to assign surgeries to operating rooms over a finite horizon. They assume surgery durations, as well as patient demand for surgery are random and aim to minimize expected cost of cancellations, postponements, and OR overtime costs. They propose a progressive hedging algorithm and heuristics to obtain solutions.

[Addis et al., 2014] propose a robust optimization approach for assigning patients to surgery blocks in a block scheduling setting. They assume uncertain duration for surgeries and propose a static robust formulation to minimize a function that penalizes associated waiting time, urgency, and tardiness of patients. Although this modeling effort provides a robust formulation for the surgery planning problem, there is no focus on the downstream resources and their effect of the schedule. In addition, the formulation employs a static formulation that does not include any recourse decisions. [Shylo et al., 2012] propose a model for the batch scheduling of surgeries with uncertain duration to surgery blocks. The goal is to maximize the expected utilization of operating rooms subject to a set of probabilistic capacity constraints. The authors show that their proposed method produces significantly better schedules in terms of performance as compared to simple heuristic scheduling rules. While addressing the high-volume batch scheduling, the proposed methodology does not address the downstream effects of the proposed schedule.

[Min and Yih, 2010] propose a two-stage stochastic programming approach to model the elective surgery planning problem. They consider uncertainty in both

surgery duration and the LOS in the SICU. They employ sample average approximation to obtain optimal solutions to this problem. While this study addresses the uncertainty in both surgery duration and LOS, the two-stage formulation is risk-neutral, i.e., it uses the expectation (mean) as the form of the objective in the second-stage. The use of distributional information may not be appropriate for construction of scenarios for short-term planning. For example the surgery duration and LOS distribution for a small group of heart patients on the waiting list for the upcoming week, may be drastically different than the general distribution obtained from the data collected over years.

[Fügener et al., 2014] consider the master surgery scheduling (MSS) problem and its effects on the downstream units. They propose an analytical approach to calculate the exact distribution for downstream resources for a given MSS. Next they define multiple cost measures resulting from the MSS and methods to minimize these costs. They rely on the existence of empirical data for every specialty such as admission probability, LOS probability, etc., to characterize the probability distributions. Obtaining an accurate estimate for these probabilities is not always possible and requires large number of data points.

As can be seen, the surgery planning problem with downstream resource capacity considerations under uncertainty has received relatively less attention while it is known that not having a holistic planning approach and focusing on isolated units increases the chance of suboptimal or globally infeasibe solutions [Fügener et al., 2014]. In our study, we focus on elective surgery patients with uncertain surgery duration and LOS in the SICU. However, we model these uncertainties within the framework of a robust optimization model. The main reason for employing this approach is that in many cases, obtaining and characterizing a probability distribution can be very difficult. In addition, the existence of distributions will not necessarily ensure tractable

solution methods. The first difficulty is producing the number of scenarios that can be a representative sample of the multi-dimensional uncertainty which can be prohibitively large. Next, the two-stage stochastic programming approach requires the evaluation of the second-stage for each scenario, which if coupled with large number of scenarios can cause tractability issues.

2.4 Combinatorial Optimization

An important area in decision-making is combinatorial optimization (CO). CO is a powerful tool for modeling many complex problems and has enabled the optimization field to tackle very important applications. In this section we present a brief review of literature on two important and well-known classes of CO problems that can be used to model many applications including health care operations.

2.4.1 Generalized Assignment Problem (GAP)

The generalized assignment problem (GAP) is the problem of optimally assigning n jobs to m constrained resources. Each job has to be assigned and the amount of resource on each machine foe a given job is known. The goal is to either minimize the assignment costs or to maximize the assignment profit [Wolsey and Nemhauser, 2014].

The deterministic version of this problem (DGAP) aims to assign jobs to resources in order to optimize an objective function in the form of revenues or costs, while ensuring that the required capacity for each resource does not exceed the available capacity of the machine. In order to formally develop the formulation for DGAP, we

define the parameters as follows:

i	index of resources, $i = 1, ..., m$
j	index for jobs, $j = 1, ..., n$
R_i	amount of resource i available, $i = 1, ..., m$
r_{ij}	amount of resource i needed by job j if assigned, $i = 1, ..., m, j = 1, ..., n$
c_{ij}	cost of assigning resource i to job j, $i = 1, ..., m, j = 1, ..., n$.

The decision variable is defined for all resource-job pairs as follows:

$$x_{ij} = \begin{cases} 1 & \text{if job } j \text{ is assigned to resource } i \\ 0 & \text{otherwise} \end{cases}$$

Considering the definitions, DGAP can be formulated as follows:

$$\min \quad \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \tag{2.11a}$$

$s.t.$

$$\sum_{i=1}^{m} x_{ij} = 1 \qquad \forall j \tag{2.11b}$$

$$\sum_{j=1}^{n} r_{ij}x_{ij} \leq R_i \qquad \forall i \tag{2.11c}$$

$$x_{ij} \in \{0,1\} \qquad \forall i,j \tag{2.11d}$$

Objective function 2.11a is the minimization of the assignment costs. Constraint 2.11b ensures that each job is exactly assigned to one resource. Constraint 2.11c ensures that for each resource, the total required capacity is not exceeding the available capacity. Constraint 2.11d defines the binary domain of the decision variables.

GAP has been extensively studied. It is a classic problem in CO and its structure can be seen as a subproblem in many different applications. [Öncan, 2007] provides an excellent survey of different extensions to GAP and various solutions techniques that are employed. Scheduling applications, transportation and routing applications, telecommunication applications, and location problems are a few applications where GAP appears as a subproblem.

There are two different types of uncertainty that have been considered in the literature. The first type is when the capacity of resources or resource requirements are not exactly known. The second type is if the job or machine availability is uncertain.

[Albareda-Sambola et al., 2006] address the later type of uncertainty in which only a random subset of jobs are required to be processed. A two-stage stochastic programming formulation is developed in which the first-stage decisions are the assignment of jobs to resources and in the second-stage, some jobs may have to be reassigned due to resource overloads.

[Albareda-Sambola and Fernández, 2000] consider the stochastic GAP with Bernoulli demands. Two policies are proposed to handle the infeasibility of the realized demand vector. The authors show that the policy constructed based on the chance-constraints performs the best in terms of the objective function value and avoiding infeasibility.

The literature on using robust optimization formulations of the generalized assignment problem is very sparse. [Fu et al., 2014] present a robust optimization approach for bottleneck GAP under uncertainty in the amount of available resources in each machine. The uncertainty is modeled using scenarios. Robustness is assured by defining the objective function that aims to minimize a linear combination of expected costs, variance of costs, and sum of infeasibilities.

Special cases of GAP can be used to formulate surgery scheduling optimization. The main difference between GAP and surgery scheduling is the assumption that the resource requirements in GAP depend on both the jobs and the machines. In surgery scheduling, surgery durations only depend on the surgery itself and not on the surgery block they are assigned to.

Stochastic programming (SP) techniques have been used to address uncertain parameters in GAP. However, exact and known distributional information is required for the use of stochastic programming techniques. In addition, in the case of two-stage SP, usually a large number of scenarios is required to characterize the uncertainty which can pose intractability due to the large size of the problem or prohibitive number of sub-problems to be solved.

2.4.2 Extensions to Bin Packing (BP) Problem

The bin packing (BP) problem is an important and well-studied class of CO problem (see [Hoffman and Padberg, 2001] and references therein). It can viewed as an extension and generalization to the GAP where the number of resources (machines) that can be used is a decision variable and each resource (machine) has a fixed cost associated with it. The problem parameters for the deterministic bin packing (DBP) problem can be defined as follows:

i	index of resources, $i = 1, ..., m$
j	index for jobs, $j = 1, ..., n$
R_i	amount of resource i available, $i = 1, ..., m$
f_i	amount of fixed cost for utilizing resource i, $i = 1, ..., m$
r_{ij}	amount of resource i needed by job j if assigned, $i = 1, ..., m, j = 1, ..., n$
c_{ij}	cost of assigning resource i to job j, $i = 1, ..., m, j = 1, ..., n$.

The decision variable is defined for all resource-job pairs as follows:

$$x_{ij} = \begin{cases} 1 & \text{if job } j \text{ is assigned to resource } i \\ 0 & \text{otherwise} \end{cases}$$

$$y_i = \begin{cases} 1 & \text{if machine } i \text{ is used} \\ 0 & \text{otherwise} \end{cases}$$

Considering the definitions, DGAP can be formulated as follows:

$$\min \quad \sum_{i=1}^{m} f_i y_i + \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{2.12a}$$

$s.t.$

$$\sum_{i=1}^{m} x_{ij} = 1 \qquad \forall j \tag{2.12b}$$

$$\sum_{j=1}^{n} r_{ij} x_{ij} \leq R_i y_i \qquad \forall i \tag{2.12c}$$

$$y_i, x_{ij} \in \{0, 1\} \qquad \forall i, j \tag{2.12d}$$

In this formulation, if machine i is not used then $y_i = 0$, therefore no job can be assigned to that machine since the second set of constraints 2.12c forces those assignments to be zero. This formulation is used in health care applications where the variable y_i corresponds to opening an operating room and x_{ij} captures the assignment of surgery blocks to operating rooms.

The bin packing formulation for this application is used by [Denton et al., 2010]. The propose a 2SSP and robust optimization approach to address this problem. The robust formulation is proposed as an approximation for the 2SSP formulation. The authors also provide a heuristic to find solutions quickly and show that the robust formulation and their proposed heuristic find high quality solutions quickly. We later

study the robust formulation proposed by [Denton et al., 2010] in greater details in Chapter 4.

[Berg and Denton, 2014] provide fast approximation algorithms for online scheduling of patients in an outpatient setting in which the problem is also modeled as a bin packing problem. The authors show the theoretical performance guarantees for list-based approximation methods as well as an approximation is common in practice. The authors show that the policy of reserving rooms for patient groups in advance can perform very poorly.

[Bam et al., 2015] aim to address the problem of assigning surgeries to ORs as well as surgery sequencing to mitigate downstream capacity issues in the PACU. Due to the complexities, the authors propose a 2-phase approach. First, they propose a bin packing formulation and heuristic for assigning surgeries to ORs. Next, in phase 2 they create the sequence of surgeries. They show that their easy-to-implement heuristics performs very well in both deterministic and stochastic settings.

In next chapter, we provide a detailed description of the surgery scheduling while considering the downstream capacity at the SICU. We propose a two-stage robust formulation for to create schedules that are robust against the uncertainties in the surgery duration and patient's LOS in the SICU.

Chapter 3: Robust Surgery Scheduling Considering Downstream Capacity

3.1 Model Development

In this section, the model development for surgery and downstream capacity planning under uncertainty is presented. We first provide some notation, then a deterministic formulation is presented. Next, we define our model of uncertainty in detail. Finally, the robust formulation to model this problem is presented. We study the structure of the proposed formulation and its characteristics in the subsequent section.

3.1.1 Definitions and Deterministic Formulation

We define set B as the set of surgery blocks in a given decision period (usually a week or two weeks long). Set S defines the set of specialties that are included in the cyclic schedule. Each block, $b \in B$, is dedicated to only one type of specialty while there can be multiple blocks of the same specialty during a cycle in the surgery schedule. B_s is used to denote the set of blocks for specialty s during the planning horizon. Set $I = \{1, ..., n\}$ represents the elective surgery patients. Set $I_s \subseteq I$ represents the set of patients that require specialty s. Note that each patient is assigned to one specialty. Therefore patient i of specialty s can be assigned to any of the blocks $b \in B_s$ during the planning horizon. We assume, without loss of generality, that the length of the planning horizon is T days and is an integer multiple of the surgery schedule cycle length. Each surgery block b has a pre-allocated length of time which is denoted by

Table 3.1: Definition of parameters

Symbol	*Definition*
S	Set of all the specialties $s = 1, .., \|S\|$
B	Set of all the surgery blocks b in the master surgery schedule
I	Set of all the patients $i = 1, .., n$ that are waiting to be assigned for surgery
T	Length of the decision period in days
B_s	Set of surgery blocks that perform surgeries with specialty $s \in S$
I_s	Set of patients that require surgeries with specialty $s \in S$
c_s	Overtime cost for specialty $s \in S$
b'	Index for the dummy surgery block
t_b	Index for the day of surgery block $b \in B$
h_b	Length of surgery block $b \in B$
d_i	Surgery duration for patient $i \in I$ in
l_i	Length-of-stay for patient $i \in I$ in the SICU
r_t	Number of available SICU beds on day $t = 1, .., T$
e_t	Unit cost for not having a SICU bed for patient
a_{ib}	Cost of assigning patient $i \in I$ to surgery block $b \in B$

h_b on a specific day t_b. Due to uncertainty in surgery times, a surgery block may need to use extra time to finish the scheduled surgeries. Therefore, overtime cost c_s for each unit of time is incurred for specialty s. For ease of modeling, a dummy block $b' \in B$, is considered for patients who are not assigned to any surgery blocks during the planning horizon T and are postponed be assigned to surgery during the next planning horizon. The cost of assigning patient i to each block is defined as a_{ib}, where $a_{ib} < a_{ib'} \; \forall b \neq b'$. This represents the admission costs for each patient considering their waiting times and priorities.

Associated with each patient i, there is the length-of-stay l_i which denotes the number of consecutive days that the patient is required to stay in the SICU following surgery. In addition, there is the surgery duration d_i, which represents the time required to perform the surgery for patient i. The SICU has limited number of beds on each day, represented by r_t. In the case of lack of capacity in the SICU, a patient

will be denied admission to the SICU and/or has to be transferred to units with lower level of care. The cost incurred for each day a patient is not receiving the care in the SICU is denoted by e_t. Table 3.1 summarizes the list of sets and parameters used in our model.

To formulate the problem, we define x_{sib} be equal to one if patient i with specialty s is assigned to perform surgery on block b, and is set to zero otherwise. y_{it} is one if patient i is in need of a SICU bed on day t, and is zero otherwise. Continuous decision variable o_b captures the amount of overtime incurred during the surgery in block b. u_t counts the number of patients on day t that are in need of a SICU bed, but cannot receive a bed due to the lack of capacity.

Here we provide the Deterministic Operating Room Planning with Downstream

Capacity (DORP-DC) before introducing its robust counterpart.

$$\min \quad \sum_{s\in S}\sum_{i\in I}\sum_{b\in B} a_{ib}x_{sib} + \sum_{s\in S}\sum_{b\in B_s\setminus\{b'\}} c_s o_b + \sum_{t=1}^{T} e_t u_t \tag{3.1a}$$

$s.t.$

$$\sum_{b\in B_s\cup\{b'\}} x_{sib} = 1 \qquad i \in I_s, s \in S \tag{3.1b}$$

$$\sum_{i\in I_s} d_i x_{sib} \leq h_b + o_b \qquad b \in B_s \setminus \{b'\}, s \in S \tag{3.1c}$$

$$y_{it} \geq x_{sib} \qquad s \in S, i \in I_s, b \in B_s, t = t_b, .., t_b + l_i - 1 \tag{3.1d}$$

$$\sum_{i\in I} y_{it} \leq r_t + u_t \qquad \forall t \tag{3.1e}$$

$$y_{it}, x_{sib} \in \{0,1\}, o_b \geq 0, u_t \geq 0 \qquad \forall s, i, b, t \tag{3.1f}$$

In DORP-DC, the objective (3.1a) is to minimize a measure of total costs. The first term on the left is the sum of the cost of assigning patients to surgery blocks (which includes patient priority and waiting time). The second term calculates the overtime costs in surgery blocks. The third term measures the total cost of lack of SICU capacity which causes premature discharges or transfers. The first constraints, (3.1b), enforce the assignment of patients to blocks, requiring each patient to be assigned exactly once (including the dummy surgery block) to a block within the

required specialty. In cases when patient i has to be assigned for surgery and cannot be postponed to the next decision period, we can add $x_{sib'} = 0$ as a constraint to enforce an assignment during the current period. The second constraints, (3.1c), calculate the value of overtime for each surgery block based on the assignments and surgery duration. Constraints (3.1d) are defined to indicate if a patient is in need of a SICU bed on any given day based on their assignment and LOS. This constraint ensures that patient i stays in the SICU for l_i consecutive days upon performing the surgery on day t_b, which is the day of surgery for block b. Constraints (3.1e) enforces the SICU capacity limitation and calculates the number of patients that need SICU beds for each day, but cannot be accepted to the SICU due to lack of capacity. In this setting, we assume there is another downstream unit with lower level of care (e.g., general ward) with unlimited capacity, where patients that cannot have a SICU bed are transferred. There is a penalty cost incurred for such transfer to model the undesired health risks imposed on patients due to receiving a lower level of care. This model does not select which patients are to be transferred out of the SICU, and assumes when a bed becomes available, patients are transferred back to the SICU. These assumptions are not restrictive since the decision for transferring a patient out of the SICU should be based on his/her medical conditions and our formulation does not consider such information. In addition, it makes sense to bring patients back to SICU as soon as beds are available so patients receive the appropriate level of care. Note that the value for the variable u_t is integer since both y_{it} and r_t are integer valued. Therefore, we can drop the constraint that forces variable u to be integer. The final constraints, (3.1f), define the domain for the decision variables.

3.1.2 Robust Model

In our deterministic formulation, it is assumed that all the parameters of the problem are known with certainty. However in reality, it is very difficult, if not impossible, to predict the values for surgery duration and LOS in the SICU. Therefore, these parameters are assumed to be uncertain while belonging to a known set.

Considering the uncertainty and the decision process, this problem can be viewed as a two-stage process in which decisions to assign patients to surgery blocks (x_{sib}) are made in the first stage. Next, uncertainty in the surgery duration and LOS for each assigned patient is realized. In the second stage, the goal is to minimize the defined worst-case scenario for the overtime and denied SICU admission costs. This is an adaptive process which tries to employ the best recourse decision after the realization of uncertainty. Considering the worst-case realization of uncertainty can be a suitable approach since the risk associated with not satisfying the SICU bed requirements can have very adverse effects on patients safety and health.

We assume that only a subset of uncertain parameters will deviate from their nominal value and try to minimize the the worst-case costs. Let us define $\tilde{d}_i \in [\overline{d}_i, \overline{d}_i + \hat{d}_i]$, in which $\overline{d}_i$ is the nominal value for surgery duration for patient i, while $\hat{d}_i$ is the total deviation from the nominal value that the duration can have. Without loss of generality, for LOS, we define $\tilde{l}_i \in \{\bar{l}_i, \ldots, \bar{l}_i + \hat{l}_i\}$ to represent the uncertainty set for LOS in the SICU for patient i. For simplicity, we further assume that the LOS for each patient is integer-valued and it is corresponding to the number of days; however, to model a finer granularity in time, shorter time periods can be considered (e.g. hours, shifts). Note that the assumption for integer-valued LOS is very close to reality since SICU release decisions are generally made once a day by care providers. In the case of surgery durations, $z_i = \dfrac{\tilde{d}_i - \overline{d}_i}{\hat{d}_i}$ is the normalized deviation from the nominal surgery

duration for patient i and $0 \leq z_i \leq 1$. Following the notation defined by [Bertsimas and Sim, 2004], we define $\Gamma_d = (\Gamma_d^1, \ldots, \Gamma_d^{|S|})$ as the *budget of uncertainty* vector for surgery durations within each specialty. Then we enforce $\sum_{i \in I_s} z_i \leq \Gamma_d^s, \forall s$, which limits the total possible normalized deviation from the nominal value being less than the budget of uncertainty, Γ_d^s. In other words, if Γ_d^s is integer-valued, only Γ_d^s patients of specialty s can have surgery durations equal to their highest possible duration. In a simpler case, we can define Γ_d to be a single parameter that limits the deviations over all specialties.

Different surgery types and specialties have different levels of uncertainty associated to them. For example, the possibility of deviation from the nominal time in a standard joint replacement surgery is expected to be far less than an open-heart surgery due to the inherent uncertainties and possible complicating factors. Therefore, based on the preference of management, different uncertainty budgets can be allocated to different specialties. In case the decision-maker has no preference or information on specialty-related risks, the formulation can be easily adapted by defining a single inequality for the budget of uncertainty rather than $|S|$ inequalities. The same ideas and assumptions can be applied to define the uncertainty set for the LOS in the SICU.

Following the same assumption made for surgery durations, we define $\frac{\tilde{l}_i - \bar{l}_i}{\hat{l}_i}, \forall i$ and enforce the budget of uncertainty $\Gamma_l = (\Gamma_l^1, \ldots, \Gamma_l^{|S|})$ for LOS in the SICU as $\sum_{i \in I_s} \frac{\tilde{l}_i - \bar{l}_i}{\hat{l}_i} \leq \Gamma_l^s, \forall s$. We also assume that the realization of uncertainty in surgery duration is independent from the realization in the LOS.

We define the uncertainty sets for surgery duration and LOS as follows:

$$\mathcal{U}_d = \{d \in R^n : \tilde{d}_i = \bar{d}_i + z_i \hat{d}_i, 0 \leq z_i \leq 1 \ \forall i, \sum_{i \in I_s} z_i \leq \Gamma_d^s \ \forall s\} \tag{3.2}$$

$$\mathcal{U}_l = \{l \in R^n : \tilde{l}_i \in \{\bar{l}_i, \ldots, \bar{l}_i + \hat{l}_i\} \ \forall i, \sum_{i \in I_s} \frac{\tilde{l}_i - \bar{l}_i}{\hat{l}_i} \leq \Gamma_l^s \ \forall s\}. \tag{3.3}$$

The data for the nominal values and worst-case deviations can be obtained from subject matter experts, physicians, or managers that have detailed information about each patient's health and conditions. Depending on the risk-attitude of the decision-maker, a value for the budget of uncertainty is chosen. Higher values of the uncertainty budget allow for larger deviations in uncertain parameters and the resulting schedules are more conservative.

Considering the assumptions and definitions mentioned earlier, the formulation for the Robust Adaptive Surgery Planning with Downstream Capacity (RASP-DC) problem can be written as follows:

$$\min \quad \sum_{s \in S} \sum_{i \in I} \sum_{b \in B} a_{ib} x_{sib} + opt[R(x, \Gamma_d, \Gamma_l)] \tag{3.4a}$$

$s.t.$

$$\sum_{b \in B_s \cup \{b'\}} x_{sib} = 1 \qquad i \in I_s, s \in S \tag{3.4b}$$

$$x_{sib} \in \{0, 1\} \qquad \forall s, i, b \tag{3.4c}$$

where $opt[R(x, \Gamma_d, \Gamma_l)]$ is the optimal solution to the recourse problem, $R[(x, \Gamma_d, \Gamma_l)]$:

$$\max_{\tilde{d}\in\mathcal{U}_d,\tilde{l}\in\mathcal{U}_l} \min \quad \sum_{s\in S}\sum_{b\in B_s\setminus\{b'\}} c_s o_b + \sum_{t=1}^{T} e_t u_t \tag{3.5a}$$

$s.t.$

$$\sum_{i\in I_s} \tilde{d}_i x_{sib} \leq h_b + o_b \qquad b \in B_s \setminus \{b'\}, s \in S \tag{3.5b}$$

$$y_{it} \geq x_{sib} \qquad s \in S, i \in I_s, b \in B_s, t = t_b, .., t_b + \tilde{l}_i - 1 \tag{3.5c}$$

$$\sum_{i\in I} y_{it} \leq r_t + u_t \qquad \forall t \tag{3.5d}$$

$$y_{it} \in \{0,1\}, o_b \geq 0, u_t \geq 0. \qquad \forall s, i, b, t \tag{3.5e}$$

Note that in our formulation we have not considered an upperbound on the value of overtime decision variables $(o_b, \forall b)$ which makes the development of our two-stage formulation simpler due to the complete recourse property. In reality, the overtime cannot exceed a certain amount of time and our formulation can be extended by adding upperbound constraints on variables $o_b, \forall b$. Considering an upperbound for these variables will eliminate the complete recourse property and requires the solution procedure to include feasibility cuts as well as the optimality cuts (see [Birge and Louveaux, 2011] chapters 3 and 5). Another way to address this issue is to create a

convex piece-wise linear cost structure for overtime in surgery durations. Such structure would also require a more complex treatment since it requires binary variables in the second-stage to model the piece-wise linear cost structure. We choose to utilize the simpler formulation with simple recourse since the derivation for the case that requires feasibility cuts is a straight forward extension of this work.

In the first stage, prior to any knowledge of the realization of uncertainty, assignment decisions are made. Patients are assigned to surgery blocks. During the operations, the value for surgery duration will be realized and the costs of overtime for each surgery block is incurred. In addition, the LOS in the SICU is realized which determines the utilization of the SICU capacity and possible denied admissions or transfers. The uncertainty exists in the technology matrix of the first constraints, (3.5b), and the set of indices of the forth set of constraints, (3.5c). In fact, using this formulation, if the LOS is uncertain, the number of constraints of type (3.5c) will be uncertain which poses serious complexity issues for solving this problem. One can reformulate this constraint as $\sum_{t=t_b}^{t_b+\tilde{l}_i-1} y_{it} \geq \tilde{l}_i x_{sib} \quad \forall s, i, b$, however, these constraints cannot solve this complexity issue either, since they have an uncertain number of variables in the summation in the left-hand-side. The goal of the decision-maker is to minimize the costs associated with surgery block overtimes and denied admissions to the SICU. The goal of the second-stage problem is to minimize the worst-case recourse costs based on the definition of the uncertainty sets. It is important to note that this formulation cannot be supplied to any solver in the presented form. An extensive formulation can be obtained by enumerating all the possible realizations of uncertainty. However this is prohibitive even for problems of medium size.

Note that in recourse problem $R[(x, \Gamma_d, \Gamma_l)]$, ((3.5a)-(3.5e)), x_{sib} is not a decision variable. First-stage decision variable values $\boldsymbol{x}$ are passed to the second-stage as parameters. It is clear from the formulation of recourse problem $R[(x, \Gamma_d, \Gamma_l)]$ that

the variables related to the surgery block overtime, o_b, are independent from the variables capturing the status of the SICU bed capacity, y_{it}, and denied admissions, u_t. This observation helps us to decompose the recourse problem further into two different and important problems: (1) **Surgery Block Capacity (SBC)** problem which aims to calculate the worst-case minimum overtime costs due to deviations in surgery durations, (2) **Downstream Capacity (DC)** problem which aims to calculate the worst-case minimum costs of denied admissions to the SICU due to the deviations in the LOS.

Following the observation on the separability of these problems we can reformulate the robust adaptive surgery planning with downstream capacity as follows:

$$\min \quad \sum_{s\in S}\sum_{i\in I}\sum_{b\in B} a_{ib}x_{sib} + opt[R_d(x,\Gamma_d)] + opt[R_l(x,\Gamma_l)] \tag{3.6a}$$

$s.t.$

$$\sum_{b\in B_s\cup\{b'\}} x_{sib} = 1 \qquad i\in I_s, s\in S \tag{3.6b}$$

$$x_{sib}\in\{0,1\} \qquad \forall s,i,b \tag{3.6c}$$

where $opt[R(x,\Gamma_d)]$ is the optimal value of the surgery block capacity recourse problem

$R_d[(x, \Gamma_d)]$:

$$\max_{\tilde{d} \in \mathcal{U}_d} \quad \min \quad \sum_{s \in S} \sum_{b \in B_s \setminus \{b'\}} c_s o_b \tag{3.7a}$$

$s.t.$

$$\sum_{i \in I_s} \tilde{d}_i x_{sib} \leq h_b + o_b \qquad b \in B_s \setminus \{b'\}, s \in S \tag{3.7b}$$

$$o_b \geq 0 \qquad \forall b \tag{3.7c}$$

and $opt[R_l(x, \Gamma_l)]$ is the optimal value of the downstream capacity recourse problem $R_l[(x, \Gamma_l)]$:

$$\max_{\tilde{l} \in \mathcal{U}_l} \quad \min \quad \sum_{t=1}^{T} e_t u_t \tag{3.8a}$$

$s.t.$

$$y_{it} \geq x_{sib} \qquad s \in S, i \in I_s, b \in B_s, t = t_b, .., t_b + \tilde{l}_i - 1 \tag{3.8b}$$

$$\sum_{i \in I} y_{it} \leq r_t + u_t \qquad \forall t \tag{3.8c}$$

$$y_{it} \in \{0, 1\}, u_t \geq 0. \qquad \forall i, t \tag{3.8d}$$

While in our case (having simple recourse) the second-stage problem can be formulated as a maximization problem, we chose to use the $\max - \min$ formulation to allow for a universal treatment of the second-stage subproblems. In addition, the proposed formulation is based on the definition of the two-stage robust optimization problems in the literature which allows for easier extension of more complex recourse

structures for future research.

In the following sections, we study each of these problems in depth and present insightful structural properties that provide insight to each of these problems. The properties are employed to provide a mixed-integer linear programming (MILP) reformulation for each of the sub-problems that can be solved using commercial solvers.

3.2 Structural Properties

In this section, we study the structural properties of surgery block capacity and downstream capacity problems. These insights help us reformulate these problems to more tractable mixed-integer linear programs (MILPs) that can be solved using commercial solvers, which in turn allows us to solve the overall surgery planning problem.

3.2.1 Surgery Block Capacity Problem

In this section, we focus on the surgery block capacity recourse problem defined by (3.7a)-(3.7c). This problem is a two-level optimization problem in which, first, a maximization over the uncertainty set defines the outer-level problem and seeks the worst-case scenario for a given surgery assignment. After the realization of the surgery durations, the inner-minimization problem aims to minimize the overall overtime costs. In the inner-minimization problem, the only decision variables are those that capture the overtime for each surgery block ($o_b, \forall b$), and they are continuous. Using the definition of the uncertainty set $\mathcal{U}_d$, we can substitute for the values of $\tilde{d}_i$ and

have the surgery block capacity recourse problem as follows:

$$\max_{\sum_{i\in I_s} z_i \le \Gamma_d^s, \forall s, 0\le z_i \le 1, \forall i} \quad \min \quad \sum_{s\in S} \sum_{b\in B_s\setminus\{b'\}} c_s o_b \tag{3.9a}$$

$s.t.$

$$\sum_{i\in I_s} (\bar{d}_i + \hat{d}_i z_i) x_{sib} \le h_b + o_b \qquad b \in B_s \setminus \{b'\}, s \in S \tag{3.9b}$$

$$o_b \ge 0. \qquad \forall b \tag{3.9c}$$

Since the inner-minimization is a linear program, strong duality can be used to substitute the inner-minimization problem with its dual, which can be written as a single maximization problem as follows:

$$\max \quad \sum_{s\in S} \sum_{b\in B_s} \left[\sum_{i\in I_s} (\bar{d}_i + \hat{d}_i z_i) x_{sib} - h_b \right] \pi_b^s \tag{3.10a}$$

$s.t.$

$$\sum_{i\in I_s} z_i \le \Gamma_d^s \qquad s \in S \tag{3.10b}$$

$$0 \le \pi_b^s \le c_s \qquad s \in S, b \in B_s \tag{3.10c}$$

$$0 \le z_i \le 1. \qquad \forall i \tag{3.10d}$$

(3.10a)-(3.10d) is the reformulation for the surgery block capacity recourse problem $R_d(x, \Gamma_d)$ that transforms the $\max - \min$ objective into a single maximization problem. Note that variable π_b is the dual variable associated with the capacity constraint (3.9b) for surgery block b. Due to the existence of the bilinear term $z_i \pi_b^s$, the

second-stage problem $R_d(x, \Gamma_d)$ is the maximization of a bilinear function over linear constraints. Bilinear programming is a special case of quadratic programming and the objective function, in general, is neither convex nor concave [Gallo and Ülkücü, 1977]. This is a limiting factor in using standard convex optimization solvers to obtain optimal solutions to the second-stage problem. The following propositions, based on the structure of the recourse problem, enable us to reformulate the second-stage problem $r(x, \Gamma_r)$ as a mixed-integer linear program (MILP).

Proposition 3.2.1. *If the components of the budget of uncertainty Γ_d are integer values, there exists an optimal solution (π^*, z^*) to the second-stage problem $R_d(x, \Gamma_d)$ such that $z_i^* \in \{0, 1\}, \forall i$.*

Proof. Let us define the feasible region for the second-stage problem by the following polyhedra $\Pi = \{\pi \in \mathbb{R}^m | 0 \leq \pi_b^s \leq c_s, \forall s, b \in B_s\}$ and $\mathcal{Z}(\Gamma_d) = \{z \in \mathbb{R}^n | \sum_{i \in I_s} z_i \leq \Gamma_d^s, \forall s, 0 \leq z_i \leq 1, \forall i\}$. Note that both sets Π and $\mathcal{Z}(\Gamma_d)$ are bounded (all variables are bounded) and therefore an optimal solution (π^*, z^*) exists such that π^* is an extreme point of Π and z^* is an extreme point of $\mathcal{Z}(\Gamma_d)$ [Gallo and Ülkücü, 1977]. This implies that when Γ_d is composed of all integer values, there exists an optimal solution such that $z^* \in \{0, 1\}^n$ (also see [Gabrel et al., 2014a]). □

Proposition 3.2.2. *For any $s \in S, b \in B_s$ in the second-stage, the optimal solution is $\pi_b^{s*} \in \{0, c_s\}$.*

Proof. Due to the structure of the objective function one of the following cases is true for any $\pi_b^s, s \in S, b \in B_s$:

- First, consider the case that for a given block b with specialty s, assignment vector x, and deviation vector z, $\sum_{i \in I_s} (\overline{d}_i + \hat{d}_i z_i) x_{sib} - h_b > 0$. In this case, due to the maximization of the objective function, the optimal value for π_b^s is its upperbound c_s.

- Second, consider the case that for a given block b with specialty s, assignment x, and deviation vector z, $\sum_{i \in I_s} (\overline{d}_i + \hat{d}_i z_i) x_{sib} - h_b \leq 0$. This means that given the assignment and the deviations for resource consumption parameters, resource consumption will not exceed the available capacity h_b. In this case, due to the maximization of the objective, $\pi_b^s = 0$.

□

As a result of 3.2.1 and 3.2.2 we can reformulate and solve (3.9a)-(3.9c) as a MILP. Next we consider the downstream capacity problem and its structural properties.

3.2.2 Downstream Capacity Problem

In this section, we turn our focus to the downstream capacity recourse problem. The formulation is presented by (3.8a)-(3.8d). Note that the maximization over the uncertain parameter $\tilde{l}$ is impossible in the current formulation since $\tilde{l}$ is in the set of indexes of the constraint (3.8b) and not in the inequalities. In other words, the maximization has to decide the number of constraints of type (3.8b) as decision variables. There is no existing method to address problems of this structure. This motivates the need for a new formulation such that we can transfer the decision variable $\tilde{l}$ into the equations. Here, we provide a new formulation for DCP by redefining our decision variables.

As explained before, we can divide the process into three different decision-making stages. In the first stage, patients are assigned to surgery blocks. Considering the uncertainty in surgery duration and LOS and the definition of the budget of uncertainty, uncertain parameters assume their value in the second stage. We assume the second stage decisions are made by an adversary. For the third stage, we aim to minimize the cost of recourse for the previous stages. In order to formulate this problem we

need to redefine our variables as $v_{it} = 1$ if patient i enters SICU *by* day t, 0 otherwise, and $w_{it} = 1$ if patient i leaves SICU *by* day t, 0 otherwise.

It is important to note that *"by"* is used in the definition of the variables rather than *"at."* Stemming directly from the definition of the variables, the following inequalities hold:

$$v_{it} \leq v_{i,t+1} \qquad \forall i, t \tag{3.11a}$$

$$w_{it} \leq w_{i,t+1}. \qquad \forall i, t \tag{3.11b}$$

The first inequality indicates that if a patient has arrived to the SICU *by* day t ($v_{it} = 1$), then he/she has arrived by the days after t. Therefore, all the variables for those days must equal 1. The second inequality indicates the same principle for leaving the SICU.

Defining the variables in this way naturally adapts to our problem setup and stages. Variables v_{it} are automatically defined after the first stage decisions are made. Next, the worst-case second-stage recourse costs are calculated by choosing the times that patients will leave the SICU, controlled by variables w_{it}. The LOS for patient i is equal to $\sum_{t=1}^{T}(v_{it} - w_{it})$, and the number of patients in the SICU on day t is equal to $\sum_{i=1}^{n}(v_{it} - w_{it})$. This requires us to redefine the uncertainty set based on the variables that represent the uncertainty as the SICU departure time. Note that this reformulation transforms the parameter for the LOS (l) (which does not depend on the arrival to the SICU) into an arrival/departure process. While using the arrival and departure times we can simply calculate the LOS, the definition of uncertainty changes to be the time that a patient is released from the SICU.

Alternative Representation of Uncertainty

As discussed before, each patient i has a LOS ($\tilde{l}_i$) at the SICU that belongs to the discrete set $\{\bar{l}_i, \ldots, \bar{l}_i + \hat{l}_i\}$. We assume, without the loss of generality, that the LOS is defined to be integer which means that the LOS cannot be a fraction of a day and both $\bar{l}_i$ and $\hat{l}_i$ are also integer. We can consider fractions of a day by further dividing the steps in the time-windows. Note that the definition of our variables naturally adapts to our assumptions on the LOS in the SICU. These assumptions along with the value of first-stage variables help us fix the values of a subset of variables as follows:

$$v_{it} \geq x_{sib} \qquad t = t_b, \forall s, i \in I_s, b \in B_s \tag{3.12a}$$

$$v_{it} \leq 1 - x_{sib} \qquad t = 1, .., t_b - 1, \forall s, i \in I_s, b \in B_s \tag{3.12b}$$

$$w_{it} \geq x_{sib} \qquad t = t_b + \bar{l}_i + \hat{l}_i - 1, .., T, \forall s, i \in I_s, b \in B_s \tag{3.12c}$$

$$w_{it} \leq 1 - x_{sib}. \qquad t = 1, .., t_b + \bar{l}_i - 1, \forall s, i \in I_s, b \in B_s. \tag{3.12d}$$

The first inequalities, (3.12a), ensure that each patient goes to the SICU on the day of surgery, while the second set of inequalities, (3.12b), enforce that patients cannot go to the SICU before the day of surgery. The third set of inequalities, (3.12c), ensure that each patient can only stay in the SICU for at most $\bar{l}_i + \hat{l}_i$ days. The forth set of inequalities, (3.12d), ensure that patients cannot leave the SICU before the minimum LOS in the SICU, which is $\bar{l}_i$ days.

Considering that the LOS for patient i can be written as $\tilde{l}_i = \sum_{t=1}^{T}(v_{it} - w_{it})$, the mathematical representation of the budget of uncertainty constraint can be written as follows:

$$\sum_{i\in I_s,\hat{l}_i>0,x_{ib'}=0}\left[\frac{\sum_{t=1}^{T}(v_{it}-w_{it})-\bar{l}_i}{\hat{l}_i}\right]\leq\Gamma_l^s \qquad \forall s, \tag{3.13}$$

which is defined for the patients that have uncertainty in the LOS and are assigned to have a surgery during the planning horizon.

The set that characterizes the uncertainty in the LOS in the SICU can be redefined based on their departure time from the SICU, while their arrival is an input parameter to define the uncertainty set as follows:

$$\mathcal{U}_l(x,v)=\{w\in\{0,1\}^{n\times T}:(3.11b),(3.12c)-(3.12d),(3.13)\}. \tag{3.14}$$

This novel definition of the variables and reformulation of the uncertainty set allows us to incorporate the random parameter in the problem as a decision variable so it can easily adapt to the robust optimization approach. To the best of our knowledge, this is the first study such that the definition of uncertainty set depends on the first-stage variables.

Resulting Solvable Formulation

Considering the structural properties based on the definition of our variables, the robust adaptive surgery planning with downstream capacity problem can be formulated as a two-stage problem. In the first stage, decisions regarding the assignment of patients to surgery blocks (x_{sib}) are made. The by-product of this stage is the time for each patient to enter the SICU (v_{it}) which can be obtained using inequalities (3.11a), (3.12a), and (3.12b). Variables v_{it} represent the arrival of patients to the SICU based on the assignment of patients to the surgery blocks and the variable definitions.

Variables v_{it} naturally belong to the first stage of the problem and do not have

any impact of the objective value of the first stage nor limit the feasible region for variables x_{ib}.

The downstream capacity recourse problem, $R(\boldsymbol{x}, \boldsymbol{v}, \Gamma_l)$, can be written as follows:

$$\max_{w \in \mathcal{U}_l(x,v)} \quad \min \quad \sum_{t=1}^{T} e_t u_t \tag{3.15a}$$

$$s.t.$$

$$\sum_{i \in I} v_{it} - w_{it} \leq r_t + u_t \qquad \forall t \tag{3.15b}$$

$$u_t \geq 0. \qquad \forall i, t \tag{3.15c}$$

The objective function (3.15a) is the maximization of the denied admission costs over the uncertainty set which controls the the discharge date, and consequently the LOS for each patient. The first constraint (3.15b) calculates the number of denied admissions to the SICU based on the arrivals to the SICU (determined by v_{it}) and the SICU discharges (determined by w_{it}) for each day. The second constraints, (3.15c), define the range for the number of denied admissions.

Note that this problem has two levels. The first level is finding the worst-case realization of LOS for patients. Next, the decision-maker aims to minimize the costs associated with the realized LOS and required transfer costs. In this formulation, the inner-minimization problem is a linear program and all the arrival variables (v_{it}) are decided during the first stage while departure variables (w_{it}) are decided through finding the worst-case (maximization) realization of uncertainty in the second stage. It can be seen that although the variables u_t are defined to be continuous, since r_t is integer and arrivals and departure variables are binary, the optimal value for u_t is always integer.

In order to be able to solve the downstream capacity recourse problem (3.15a)-(3.15c), we apply strong duality to reformulate the inner-minimization as a maximization problem and also substitute for the definition of the uncertainty set $\mathcal{U}_l(\boldsymbol{x}, \boldsymbol{v})$. The downstream capacity recourse problem $R_l(\boldsymbol{x}, \boldsymbol{v}, \Gamma_l)$ is presented as follows:

$$\max \quad \sum_{t=1}^{T} \left[\sum_{i \in I} (v_{it} - w_{it}) - r_t \right] \lambda_t \tag{3.16a}$$

$s.t.$

$$\sum_{i \in I_s, \hat{l}_i > 0, x_{ib'} = 0} \left[\frac{\sum_{t=1}^{T} (v_{it} - w_{it}) - \bar{l}_i}{\hat{l}_i} \right] \leq \Gamma_l^s \qquad \forall s \tag{3.16b}$$

$$w_{it} \geq x_{sib} \qquad t = t_b + \bar{l}_i + \hat{l}_i - 1, .., T, \forall s, i \in I_s, b \in B_s \tag{3.16c}$$

$$w_{it} \leq 1 - x_{ib} \qquad t = 1, .., t_b + \bar{l}_i - 1, \forall s, i \in I_s, b \in B_s \tag{3.16d}$$

$$w_{it} \leq w_{i,t+1} \qquad \forall s, i \in I_s, b \in B_s, \forall t \tag{3.16e}$$

$$0 \leq \lambda_t \leq e_t \qquad \forall t \tag{3.16f}$$

$$w_{it} \in \{0, 1\}. \qquad \forall i, t \tag{3.16g}$$

Note that in the downstream capacity sub-problem, v_{it} are first-stage variables and have known values when solving the second stage. λ_t is the dual variable associated with the SICU capacity constraint (3.15b). In an economical sense, it defines the price of denied admission to the SICU. Therefore, the objective function is the maximization of the cost for denied admissions (3.16a). The first constraints (3.16b) enforce the budget of uncertainty for maximum possible deviations in LOS for each specialty group. These constraints allow the decision-maker to be able to have different risk preferences for different specialties. The second (3.16c) and third (3.16d) set of constraints fix the value for w_{it} such that no patient can leave the SICU before its minimum LOS ($\bar{l}_i$) has passed, and each patient cannot stay in the SICU longer than largest possible LOS ($\bar{l}_i + \hat{l}_i$). The forth constraints, (3.16e), are defined based on the definition of the variable w_{it}. The fifth constraints (3.16f) define the range for the values of the dual variables λ_t. Finally, the last set of constraints (3.16g) define the domain for the variable w_{it}.

This formulation is, in fact, a bilinear program which is generally non-convex. Next we exploit some of the structrural properties of the downstream capacity recourse problem $R_l(x, y, \Gamma_l)$ presented by (3.16a)-(3.16g) and propose a mixed-integer linear programming reformulation that can be solved using traditional methods.

The following proposition characterizes the optimal value for the cost of denied admissions in the downstream capacity recourse problem, which helps us reformulate DC into a MILP.

Proposition 3.2.3. *For any $t = 1, \ldots, T$ in the second-stage, an optimal solution to $R_l(x, v, \Gamma_l)$ can be find such that, $\lambda_t^* \in \{0, e_t\}$.*

Proof. Similar to the proof for the Proposition 3.2.2.

□

In the next section, we outline the steps of our exact solution methodology to solve our problem.

3.3 Solution Technique

The formulations for the SBC and DC subproblems are bilinear programs and, in general, are not convex. In order to be able to address the Robust Adaptive Surgery Planning with Downstream Capacity (RASP-DC) problem, we need to be able to solve each of these problems.

In the light of Propositions 3.2.1 and 3.2.2 we can reformulate the bilinear second-stage problem $R_d(x, \Gamma_d)$ as an MILP, $R_d^{MIP}(x, \Gamma_d)$, by defining $p_{ib}^s = \pi_b^s z_i, \forall s, i \in I_s, b \in B_s$ as follows:

$$\max \quad \sum_{s\in S}\sum_{b\in B_s}\sum_{i\in I_s} \overline{d}_i x_{sib} + \sum_{s\in S}\sum_{b\in B_s}\sum_{i\in I_s} \hat{d}_i x_{sib} p_{ib}^s - \sum_{s\in S}\sum_{b\in B_s} h_b \pi_b \tag{3.17a}$$

$s.t.$

$$\sum_{i\in I_s} z_i \leq \Gamma_d^s \qquad s \in S \tag{3.17b}$$

$$0 \leq \pi_b^s \leq c_s \qquad s \in S, b \in B_s \tag{3.17c}$$

$$p_{ib}^s \leq c_s z_i \qquad s \in S, b \in B_s \tag{3.17d}$$

$$z_i \in \{0,1\}, p_{ib}^s \geq 0. \qquad \forall s, b, i \tag{3.17e}$$

The MILP formulation (3.17a)-(3.17e) can be solved using standard solvers to obtain a solution for the surgery block capacity recourse problem. Next, we explore the structural properties of the SBCP in order to gain deeper insight that can be employed to improve the efficiency of the solution approach.

It can be observed from the formulation proposed for the surgery block capacity recourse problem $R_d^{MIP}(x, \Gamma_d)$ that the problem of calculating the worst-case overtime cost for surgery blocks can be decomposed into $|S|$ separate and independent problems that are connected only through the first-stage decision variables x_{sib}. In other words, the worst-case overtime costs can be calculated separately for each specialty $s \in S$. This property can be used when the size of the recourse problem over all specialties is large and a smaller problem for each specialty can be solved. In addition, this characteristic can be employed to devise a multi-cut approach similar to the well-known multi-cut L-shaped method in stochastic programming (see [Birge and Louveaux, 2011] and references there in).

As for the DC subproblem, the objective function for (3.16a) includes a bilinear term, $w_{it}\lambda_t$, and since w_{it} is defined to be a binary decision variable, we can simply reformulate the problem using the same technique in previous sections by having $q_{it} = w_{it}\lambda_t, \forall i, t$. Keeping in mind the negative coefficient of q_{it} in the objective function, the linearization requires the addition of $q_{it} \leq \lambda_t, \forall i, t$, $q_{it} \leq e_t w_{it}, \forall i, t$, and $q_{it} \geq \lambda_t - e_t(1 - w_{it}), \forall i, t$ as constraints. The first inequality is redundant as a result of the Proposition 3.2.3. The MILP downstream capacity recourse problem $R_l^{MIP}(x, v, \Gamma_l)$, can be written as:

$$\max \quad \sum_{t=1}^{T} \sum_{i \in I} v_{it} \lambda_t - \sum_{t=1}^{T} \sum_{i \in I} q_{it} - \sum_{t=1}^{T} r_t \lambda_t \tag{3.18a}$$

$s.t.$

$$\sum_{i \in I_s, \hat{l}_i > 0} \left[\frac{\sum_{t=1}^{T} (v_{it} - w_{it}) - \bar{l}_i}{\hat{l}_i} \right] \leq \Gamma_l^s \qquad \forall s \tag{3.18b}$$

$$w_{it} \geq x_{sib} \qquad t = t_b + \bar{l}_i + \hat{l}_i - 1, .., T, \forall s, i \in I_s, b \in B_s \tag{3.18c}$$

$$w_{it} \leq 1 - x_{sib} \qquad t = 1, .., t_b + \bar{l}_i - 1, \forall s, i \in I_s, b \in B_s \tag{3.18d}$$

$$w_{it} \leq w_{i,t+1} \qquad \forall s, i \in I_s, b \in B_s, \forall t \tag{3.18e}$$

$$q_{it} \geq \lambda_t - e_t(1 - w_{it}) \qquad \forall s, i \in I_s, b \in B_s, \forall t \tag{3.18f}$$

$$q_{it} \leq e_t w_{it} \qquad \forall i, t \tag{3.18g}$$

$$0 \leq \lambda_t \leq e_t \qquad \forall t \tag{3.18h}$$

$$w_{it} \in \{0, 1\}. \qquad \forall i, t \tag{3.18i}$$

In the recourse formulation (3.18a)-(3.18i), the subset of variables w_{it} are fixed for some values of t. More specifically, the second and third constraints fix the values for w_{it} to either 1, or 0. In fact, for each patient i, the only binary variables that are not fixed are $\{w_{it} : t = t_b + \bar{l}_i - 1, ..., t_b + \bar{l}_i + \hat{l}_i - 1, t_b : x_{ib} = 1\}$. In other words, for each patient i, only $\hat{l}_i$ of the variables w_{it} are not fixed. Therefore, the number of fractional values in the optimal solution of the LP-relaxation of $R_l^{MIP}(x, y, \Gamma_l)$ is bounded by $\sum_{i=1}^{n} \hat{l}_i$. In the case that the deviations in the LOS at the SICU are relatively small compared to the decision making horizon T, only a small percentage of variables can be fractional and the number of variables to be branched on is small compared to the size of the problem.

The linear relaxation of this formulation does not necessarily yield optimal solutions where all the variables w_{it} are binary. The reasons for non-integer values are solely due to the existence of the first constraint (3.18b) (the budget of uncertainty) and (3.18f) despite the fact that all the other constraints are facet defining [Bertsimas and Patterson, 1998].

Unlike the surgery block capacity problem, the downstream capacity recourse problem $R_l^{MIP}(x, y, \Gamma_l)$ cannot be decomposed into independent sub-problems for each specialty. The main reason is that patients from different specialties share common resources in the SICU.

In the next section 3.3.1, we explain why the existing methods in the literature, namely cutting-plane method based on Kelley's cutting-plane (CP) algorithm [Kelley, 1960] or L-shaped method [Van Slyke and Wets, 1969],[Thiele et al., 2009], and the column-and-constraint generation method (C&CG) [Zeng and Zhao, 2013] cannot be directly employed to solve the RASP-DC.

3.3.1 Deficiencies of Previously Developed Methods

The first approach proposed to solve the two-stage robust optimization (2SRO) problems is introduced by [Thiele et al., 2009] and has its roots in the cutting-plane method ([Kelley, 1960] and [Van Slyke and Wets, 1969]). In their setting, the definition of the uncertainty set does not depend on the first-stage variables and the use of the proposed method produces optimal solutions. In the case of the CP method, a 2SRO problem is decomposed into a master and a recourse problem. The master problem generates first-stage decisions (in our case a surgery schedule is the first-stage set of decisions) and the recourse problem identifies the worst-case outcome of uncertainty and its associated cost for a given first-stage decision. In other words, the recourse problem in a 2SRO problem is a scenario generation problem that creates worst-case scenarios of uncertain parameters. This method relies on introducing the dual variables of the recourse problem into the master problem.

In the case of applying the CP method to the RASP-DC, the cut that is passed to the master problem from the DC subproblem has the following form:

$$\theta_l \geq \sum_{t=1}^{T} \left[\sum_{i=1}^{n} (v_{it} - w_{it}^k) - r_t \right] \lambda_t^k, k = 1, .., K, \tag{3.19}$$

where K is the number of constraints generated. Constraints (3.19), generated from the the DC subproblem, are not valid optimality cuts and can potentially cut off the optimal solution. The reason is, in fact, due to the dependency between the first-stage variables v_{it} (arrival to the SICU) and uncertainty variables w_{it} (departure from the SICU). Note that the LOS for each patient is bounded such that $\bar{l}_i \leq \tilde{l}_i \leq \bar{l}_i + \hat{l}_i$. While these restrictions are considered in the DC subproblem using constraints (3.16c) and (3.16d), constraints (3.19) do not explicitly consider the earliest possible arrival time

associated with a given departure time.

As an illustration, consider the case in which after solving the restricted master problem the arrival day for patient i is the second day ($v_{i,1} = 0$ and $v_{it} = 1, t = 2, ...T$) and $\bar{l}_i = 2$ and $\hat{l}_i = 1$. Keep in mind patient i can be scheduled such that his/her arrival to the SICU is on day one, and such a schedule is also feasible but not optimal in the current restricted master problem. The maximum LOS for patient i is three days. Assume a case in which the optimal solution to the DC subproblem chooses the departure day to be day five ($w_{it} = 0, t < 5$ and $w_{it} = 1, t \geq 5$), therefore $l_i = \sum_{t=1}^{T} v_{it} - w_{it} = 3$. In addition, let us assume that $\lambda_5 > 0$, which means that on day five the number of patients in need of a SICU bed has exceeded the SICU capacity. Since there is not enough capacity in the SICU for $t = 5$, adding constraints of the form (3.19) can potentially remove $v_{i1} = 1$ as a costly solution such that variable v_{i1} cannot be equal to one anymore. On the other hand, for a departure on day five for patient i, arrival on day one cannot be considered since it assumes a possible LOS of four days for this patient while the maximum LOS for patient i can be three days. In other words, this cut removes the arrival to the SICU on day one, based on the scenario where the patient leaves the SICU on day five, which clearly is an invalid scenario, considering the maximum LOS for that patient. Because these constraints do not consider the restriction on the arrival times that should be in effect by the definition of the uncertainty set, they can potentially remove solutions that have better objective values.

Column-and-constraint generation (C&CG) is proposed by [Zeng and Zhao, 2013] and introduces a large-scale deterministic equivalent formulation for the 2SRO that relies on identifying all the scenarios for the uncertain parameters. Unlike the CP methods, C&CG does not include the dual variables associated to the recourse problem in the master problem. The master problem includes constraints in the form

of the deterministic problem for every realization of uncertainty. However, as the number of uncertain parameters increases, the number of scenarios for uncertainty increases exponentially. The authors present an iterative approach to address this issue.

Note that if we apply the proposed C&CG algorithm to our problem, specifically considering the downstream capacity subproblem, at each iteration k, we solve the DC subproblem and introduce this invalid constraint into the master problem:

$$\sum_{i=1}^{n} v_{it} - w_{it}^{k} \leq r_t + u_t^k, \ \forall t \tag{3.20}$$

in which w^k corresponds to the vector of worst-case departure times with respect to the first-stage surgery assignments in iteration k. u^k is the vector of recourse variables corresponding to the specific first-stage decisions and realization of uncertainty.

The reason that the constraint (3.20) is not a valid cut is similar to the argument made for the CP method, discussed earlier this section. In essence, this constraint does not consider the restrictions on the LOS for patients when it is passed to the master problem. It is important to note that if we do not consider the downstream capacity and only focus on the uncertainty in surgery duration, both the CP and C&CG method can be applied to solve the problem since the uncertainty set is not dependent on the first-stage decisions.

In the next subsection, we propose an adapted C&CG, such that it addresses the dependence of the uncertainty set and the first-stage decisions.

Next, in Section 3.3.2 we propose an adapted (C&CG) algorithm based on [Zeng and Zhao, 2013] to solve the RASP-DC and two-stage robust problems of similar structure, specifically stage-dependent uncertainty sets.

3.3.2 Adapted-C&CG Method

We adapt the C&CG method so it can address the issue of a first-stage-dependent uncertainty set as defined in our formulation. Our adapted column-and-constraint generation (A-C&CG) algorithm employs the deterministic formulation DORD-DC to address the decision making in the master problem, while the uncertainty is realized through solving the SBC and DC subproblems. By doing so, the uncertainty defined in the master problem for the LOS is an independent parameter. In the DC subproblem, we define the problem such that the definition of the uncertainty changes to the time the patient is released from the SICU given their admission time. Algorithm 1 presents the A-C&CG to solve the RASP-DC.

The proposed A-C&CG algorithm resembles the original C&CG while it allows for the use of more sophisticated uncertainty sets. In addition, it shows great flexibility in modeling a problem as a robust optimization problem. The second-stage formulations serve as a scenario-generation step and our algorithm allows us to use a different formulation, rather than the one based on the deterministic equivalent to generate the scenarios.

In the formulation for the master problem in the A-C&CG method, the original capacity constraints are employed. However, since the value for the uncertain parameters (deviations in surgery duration denoted by $\boldsymbol{z}$ and departure from the SICU denoted by $\boldsymbol{w}$) is not known in advance, at each iteration, we solve the master problem and obtain the optimal first-stage decisions. Then, the worst-case realization of the uncertain parameters for the given first-stage decision variables is obtained by solving the recourse formulations. The information is passed back to the master problem by introducing new variables and constraints and then the master is solved again.

Note that A-C&CG has a better worst-case performance than the CP methods. It can be seen that in the C&CG method, only the extreme points of the uncertainty

Initialization;
Set $LB = -\infty, UB = +\infty, K = 0$, $O = \emptyset$.;
Master: Solve the following master problem.

$$\min \quad \sum_{s\in S}\sum_{i=1}^{n}\sum_{b\in B} a_{ib}x_{sib} + \theta \tag{3.21a}$$

$s.t.$

$$\theta \geq \sum_{s\in S}\sum_{b\in B_s\setminus\{b'\}} c_s o_b^k + \sum_{t=1}^{T} e_t u_t^k \qquad \forall k \in O \tag{3.21b}$$

$$\sum_{b\in B_s\cup\{b'\}} x_{sib} = 1 \qquad i \in I_s, s \in S \tag{3.21c}$$

$$\sum_{i\in I_s} d_i^k x_{sib} \leq h_b + o_b^k \qquad \forall k \leq K, b \in B_s \setminus \{b'\}, s \in S \tag{3.21d}$$

$$y_{it}^k \geq x_{sib} \qquad s \in S, i \in I_s, b \in B_s, t = t_b, .., t_b + l_i^k - 1, \forall k \leq K \tag{3.21e}$$

$$\sum_{i\in I} y_{it}^k \leq r_t + u_t^k \qquad \forall t, \forall k \leq K \tag{3.21f}$$

$$x_{sib}, y_{it}^k \in \{0,1\}, o_b^k \geq 0, u_t^k \geq 0 \qquad \forall s, i, b, t, \forall k \leq K \tag{3.21g}$$

Obtain the optimal solution $(x_{K+1}^*, \theta_{K+1}^*, y^{1*}, .., y^{K*}, o^{1*}, .., o^{K*}, u^{1*}, .., u^{K*})$ and set $LB = a^T x_{K+1}^* + \theta_{K+1}^*$;

Recourse:

- **Step 1-** Create arrival parameter v_{it} such that, if $x_{sib} = 1$, $v_{it} = 1, \forall t \geq t_b$, and $v_{it} = 0, \forall t < t_b$, else if $x^s_{ib'} = 1$ (assignment to the dummy block), set $v_{it} = 0, \forall t$.
- **Step 2-** Use parameter v_{it} to construct the DC subproblem (3.18a)-(3.18i).

Solve the SBC subproblem (3.17a)-(3.17e) with objective value S^*_{K+1}. Solve the DC subproblem (3.18a)-(3.18i) with objective value D^*_{K+1}. Update
$UB = \min\{UB, c^T x^*_{K+1} + S^*_{K+1} + D^*_{K+1}\}.$;
Set $U \leftarrow \min\{U, \sum_{s\in S}\sum_{i\in I_s}\sum_{b\in B_s\cup\{b'\}} a_{ib}x^{sk}_{ib} +$
$\sum_{s\in S}\sum_{b\in B_s}\left[\sum_{i\in I_s}(\bar{d}_i + \hat{d}_i z^k_i)x_{sib} - h_b\right]\pi^{sk}_b + \sum_{t=1}^{T}\left[\sum_{i=1}^{n}(y_{it} - w^k_{it}) - r_t\right]\lambda^k_t\}$;
if $U - L \leq \epsilon$ **then**
  The optimal solution for RASP-DC is found;
else
  go to Add-Cut routine;
end
Add-Cut:

- **Step 1** Using the results of DC subproblem and departure variables $w_{it}, \forall i, t$, calculate the LOS for each patient at iteration $K+1$ such that $l^{K+1}_i = \sum_{t=1}^{T} v_{it} - w^*_{it}, \forall i$.
- **Step 2** Add variables $o^{K+1}_b, \forall b$, $y^{K+1}_{it}, \forall i, t$, and $u^{K+1}_t, \forall t$ and the following constraints to the master problem:

$$\theta \geq \sum_{s\in S}\sum_{b\in B_s\setminus\{b'\}} c_s o^{K+1}_b + \sum_{t=1}^{T} e_t u^{K+1}_t \tag{3.22a}$$

$$\sum_{i\in I_s} d^{K+1}_i x_{sib} \leq h_b + o^{K+1}_b \qquad b \in B_s \setminus \{b'\}, s \in S \tag{3.22b}$$

$$y^{K+1}_{it} \geq x_{sib} \qquad s \in S, i \in I_s, b \in B_s, t = t_b, .., t_b + l^{K+1}_i - 1 \tag{3.22c}$$

$$\sum_{t=1}^{T} y^{K+1}_{it} \leq r_t + u^{K+1}_t \qquad \forall t \tag{3.22d}$$

where d^{k+1} is the optimal solution (worst-case scenario for surgery duration) obtained by solving the SBC subproblem and l^{K+1} is obtained from the optimal solution of the DC subproblem. Update $K \leftarrow K+1, O \leftarrow O \cup \{K+1\}$ and go to Master routine;

Algorithm 1: A-C&CG Algorithm for RASP-DC.

sets are required in the formulation of the problem. However, for the CP method, the extreme points of both the uncertainty sets and the dual variables of the inner-minimization problems are needed (for detailed proof see [Zeng and Zhao, 2013] and its electronic companion).

As it can be seen from the outline of the A-C&CG algorithm, at each iteration $|B| + (n+1)T$ new variables are added to the master problem. Furthermore, at each iteration k, $|B| + T + 1 + \sum_{i \in I} l_i^k$ new constraints are added to the master problem. Note that l_i^k is the realization of the LOS for patient i at iteration k. Therefore, it is likely that for problems with a high number of patients and high levels of uncertainty (large values for $\hat{l}$), the size of the master problem will grow very fast.

3.4 Computational Experiments

3.4.1 Data and Problem Setting

The numerical results in this section are based on the practice configuration presented by [Min and Yih, 2010]. There are 10 ORs and 32 available surgical blocks per week. The assignment of ORs to surgical blocks is shown in Table 3.2 and we use this example to produce a weekly block schedule. There are nine different surgical groups that perform surgeries. Each group has at least one surgical block during the week, while some have multiple blocks. A patient can be scheduled for surgery in one of the available blocks. We assume each block is eight hours long.

Each specialty s has a mean surgery duration μ_d^s and a standard deviation σ_d^s. We use these statistics to create a list of patients such that each patient has a unique set of nominal values and deviations in surgery duration and LOS. Each number is randomly selected from a distribution that is specific to each specialty. The nominal surgery

Table 3.2: Block schedule structure.

OR Room	Monday	Tuesday	Wednesday	Thursday	Friday
OR 1	ENT	ENT	ENT		
OR 2			ENT	ENT	ENT
OR 3	OBGYN		OBGYN		OBGYN
OR 4	ORTHO	ORTHO		ORTHO	ORTHO
OR 5		ORTHO		NEURO	
OR 6	GEN	GEN	GEN	GEN	
OR 7		GEN	GEN	GEN	GEN
OR 8	OPHTH	OPHTH		OPHTH	OPHTH
OR 9	VASCULAR		CARDIAC		VASCULAR
OR 10	UROLOGY		ORTHO		

duration for patient i, $\overline{d}_i$, is randomly generated from a lognormal distribution with the mean and standard deviation for the patient's required specialty [Strum et al., 2000]. The worst case deviation in surgery duration for patient i, $\hat{d}_i$, is chosen as $\hat{d}_i = \alpha\sigma_d^{s_i}$ where $\alpha \sim U(0.5, 1.5)$.

We assume the average LOS in the SICU of ENT (Ear, Nose, and Throat), OBGYN (Obstetrics and Gynecology), ORTHO (Orthopedic), NEURO (Neurosurgery), GEN (General surgery), OPHTH (Ophthalmology), VASCULAR, CARDIAC and UROLOGY are 0.1 day, 2 days, 1.5 days, 2 days, 0.05 day, 0.05 day, 3.5 days, 2 days and 0.8 day, respectively. We further assume that the standard deviation for LOS in the SICU for each specialty is equal to its mean. The nominal LOS for patient i, $\overline{l}_i$, is then generated from a log normal distribution with proposed mean and standard deviations and rounded down to obtain an integer valued. The worst case deviation in LOS for each patient i, $\hat{l}_i$, is generated from a uniform integer distribution $U_{int}(1, 4)$. Note that these numbers may be far from the reality, specifically considering that we are assigning uncertainty to all patients. However, the proposed setting for generating data allows us to test our proposed formulation to understand its behavior.

Each specialty has a relative importance factor which is used as a multiplier to address the relative importance of one specialty over another. For example, heart surgeries are relatively more important than elective knee surgeries. This feature enables the decision-maker to specify the existing relative importance among different specialties. In addition, each patient has a specific multiplier (which in reality can be provided by her/his physician or a subject matter expert) which identifies the importance or the level of emergency of her/his case with respect to other patients in the same specialty. To generate cases, each patient receives a random integer from $U_{int}(1, 10)$. With this assumption, the optimization gives the emergency cases more priority in scheduling surgeries.

The percentage of patients in need of a specific surgery is used to generate lists of patients randomly. Overtime costs incurred to each block are chosen to reflect the relative importance of the specialty. Therefore, for each specialty the overtime cost is chosen to be 100 times the relative importance multiplier per hour. Table 3.3 provides detailed information on the surgery duration statistics, case mix, and the relative importance of each specialty.

Table 3.3: Statistics for surgery duration based on surgery type.

Surgical group	μ_d^s (minute)	σ_d^s (minute)	Percentage (% of surgeries)	Relative importance
ENT	74	37	21.34	1
OBGYN	86	40	9.26	2
ORTHO	107	44	23.26	2
NEURO	160	77	5.04	5
GEN	93	49	22.12	1
OPHTH	38	19	2.98	2
VASCULAR	120	61	8.2	4
CARDIAC	240	103	2.44	5
UROLOGY	64	52	5.36	3

For the patients that are assigned to the dummy block, we assume that they have to wait until the beginning of the next week for the next round of assignments. The waiting time is calculated by subtracting the assignment day from the initial day when the patient is added to the list. The cost for having one denied admission to the SICU or patient transfer (for each day that a patient is out of SICU) is set to be 100.

3.4.2 Performance Analysis

In order to evaluate the proposed solution methodology, tests were generated with different numbers of patients. We used a single budget of uncertainty to limit the deviations in LOS (Γ_l) and one for surgery duration (Γ_d) rather than a vector. In other words, we assume that the decision-maker does not have specialty-specific risk behavior. The value for each of these budgets can vary from zero to the number of patients. Therefore, for a problem with 10 patients, there are 100 combinations to be solved. To reduce the computational burden, we limit the solution time to be 1,000 seconds for each specific choice of parameters Γ_d and Γ_l. All the computational experiments are coded in Python programming language and Gurobi 6.0 solver is used as the optimization solver. All the tests are run on a Windows machine with an Intel Core i5, 3.20GHz CPU, and 4 GB of RAM.

In order to isolate the impact of uncertainty in surgery duration, we keep $\Gamma_l = 0$ and varied the value of Γ_d. The reverse is done to study the impact of uncertainty in LOS. Figure (3.1) illustrates an example with 10 patients and three SICU beds, and another example with 40 patients and 10 SICU beds.

Figure (3.1) illustrates the performance of the solution methodology for different variations of Γ_l and Γ_d. Figure (3.1a) shows an instance with 10 patients and three SICU beds, where $\Gamma_l = 0$, and Γ_d is increasing from zero to 10. It can be seen that

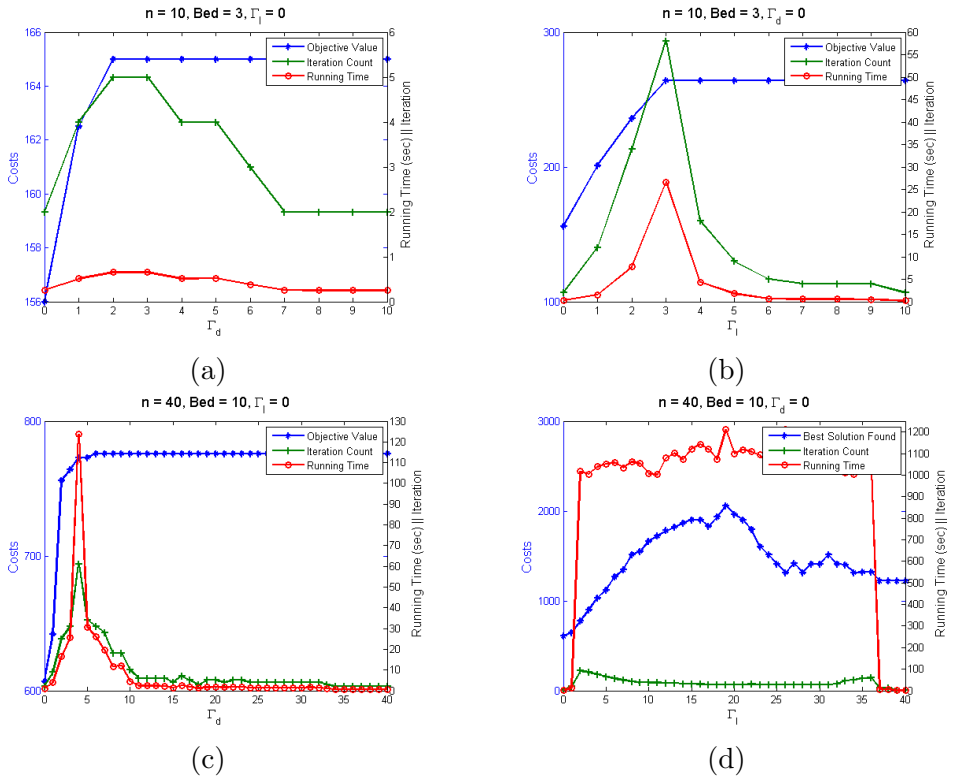


Figure 3.1: Plots for the performance of the algorithm when the source of uncertainty is changed.

for all choices of Γ_d, the instances are all solved to optimality under one second. On the other hand, Figure (3.1b) is the same instance when Γ_d is held constant at zero and Γ_l is increasing. Note that the time required to solve this instance when Γ_l is changing is significantly higher (with a maximum running time close to 30 seconds) than the case when $\Gamma_l = 0$. Note that solution times and iteration counts are shown on the same axis and scale. It can also be seen that for all values of Γ_l the instances are solved to optimality, but the objective value is higher compared to the case when the only source of uncertainty is in the surgery duration. While this behavior is very dependent on the structure of the objective and cost functions, there is a logical reason for this behavior under the given assumptions. Under the assumption of uncertain

LOS, in the case of not having enough capacity in the SICU on a specific case, the optimization either moves the surgery of the patient to another day or assigns the patient to the wait-list for the following weeks. Therefore the uncertainty in the LOS coupled with congestion in the SICU, reduces the congestion in the operating rooms.

For the case of 40 patients, Figure (3.1c) shows that for all values of Γ_d when $\Gamma_l = 0$, the problem is solved to optimality. However, it can be seen in Figure (3.1d) that most of the instances went over the 1,000 second time limit.

As the number of patients increases, the impact of uncertainty in surgery duration on the surgery schedule will increase. To test this, we generated instances with 70 patients and five SICU beds. We changed the values for Γ_l and Γ_d by 10% increments in the number of patients that have deviations associated with them. The results are shown in Figure (3.2). Figure (3.2a) shows the contours for the objective value for different pairs of budget parameters. The section with the highest value (red color) is for the combinations that could not be solved to optimality. This can also be seen in Figure (3.2c) which shows the optimality gap for each combination of budget parameters. It can be seen that instances with parameters $20\% \leq \Gamma_l \leq 40\%$ and $10\% \leq \Gamma_d \leq 30\%$ pose the most computational challenge. An interesting observation is that these combinations took fewer iterations (see Figure (3.2b)), which means that each iteration took longer time to be completed. Figure (3.2a) also shows that as the value for the parameter Γ_l increases, the impact of increasing Γ_d on the objective value is greater. Figure (3.2d) shows that both uncertainty in LOS and surgery duration can result in postponing surgeries. When $\Gamma_d = 0$, an increase in Γ_l results in large increase in the number of postponed patients. When $\Gamma_l = 0$, an increase in Γ_d results in a smaller increase as compared to the pervious case. However, when $\Gamma_l \geq 40\%$, increasing the value of Γ_d results in a fast increase in the number of patients being postponed. It is evidence that both sources of uncertainty can play a critical role in

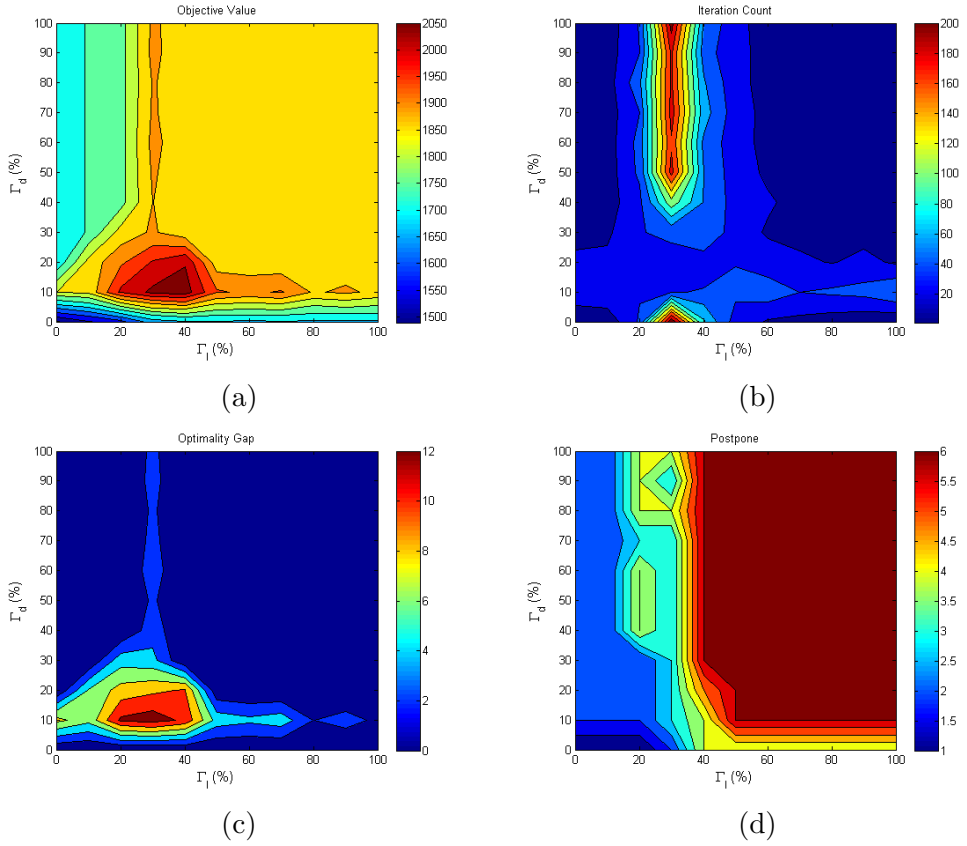


Figure 3.2: Results for an example with 70 patients and five beds, when both Γ_l and Γ_d can change.

creating the optimal surgery schedule.

Depending on the value of Γ and size of the uncertainty set (possible realizations of uncertainty) the number of iterations required to solve each instance can be different. Through testing, we observed that the computational burden of the case when LOS uncertainty is much higher than the case when surgery durations are uncertain. In addition, the case of uncertain LOS has not been studied as extensive as the case of uncertain surgery durations. From here on out, in all of our instances we fix $\Gamma_d = 0$ and only change the value of Γ_l. This helps us turn our focus to the impact of uncertainty in the LOS in downstream units on the surgery schedule, an area of

research which has received far less attention, and reduce the complexity of the results presented.

To evaluate the performance of the algorithm, problem instances with size $n = 5, 10, 15$ patients are generated randomly. The number of beds in the SICU is fixed to three beds. 10 instances of each size are generated and the optimal objective value or best solution found is recorded. The best solution found for the recourse costs is calculated. The number of iterations and running time of the algorithm for each value of Γ_l is also recorded. The number of patients that could not be assigned for a surgery during the planning horizon and have to be postponed are also calculated. Table 3.4 presents detailed results for these test cases.The table shows the average (μ) and standard deviation (σ) of the performance parameters for across the ten instances.

Table 3.4: Aggregate results for 10 instances of problems with $n = 5, 10, 15$.

Setting	Γ_l	Obj-μ	Obj-σ	ω-μ	ω-σ	Iteration-μ	Iteration-σ	Runtime-μ	Runtime-σ	Gap-μ	Gap-σ	Postponed-μ	Postponed-σ
n = 5	0	57.1	16.50	0	0	1.1	0.31	0.09	0.03	0	0	0	0
	1	67.6	17.54	0	0	2.3	0.48	0.22	0.05	0	0	0	0
	2	80.9	22.01	0	0	4.5	2.36	0.43	0.23	0	0	0.1	0.31
	3	87.4	21.7	0	0	4.2	1.54	0.4	0.14	0	0	0.1	0.31
	4	87.6	21.78	0	0	2.8	0.63	0.26	0.06	0	0	0.1	0.31
	5	87.6	21.78	0	0	2.1	0.31	0.19	0.03	0	0	0.1	0.31
n = 10	0	173.9	144.85	0	0	1.8	0.42	0.22	0.05	0	0	0.1	0.31
	1	216.6	166.37	10	31.62	11.1	6.0	1.59	0.9	0	0	0.3	0.48
	2	269.3	182.89	0	0	32.3	21.77	8.55	8.36	0	0	1.4	1.26
	3	293.1	175.59	0	0	31.2	20.94	9.73	9.64	0	0	2.3	0.94
	4	304.7	173.16	0	0	22.4	14.82	6.64	6.62	0	0	2.7	0.67
	5	304.7	173.16	0	0	11.4	4.67	2.17	1.31	0	0	2.8	0.78
	6	304.7	173.16	0	0	7	2.7	1.1	0.59	0	0	2.9	0.73
	7	304.7	173.16	0	0	5.2	0.91	0.7	0.15	0	0	2.8	0.78
	8	304.7	173.16	0	0	5	2.05	0.7	0.4	0	0	2.8	0.78
	9	304.7	173.16	0	0	3.4	1.57	0.44	0.2	0	0	2.8	0.78
	10	304.7	173.16	0	0	2.6	1.34	0.34	0.18	0	0	2.9	0.73
n = 15	0	261.2	100.02	0	0	2	0	0.35	0.06	0	0	0.1	0.31
	1	378.6	115.84	10	31.62	34.8	12.47	13.77	9.09	0	0	1.4	1.5
	2	519	110.29	90	73.78	118.1	38.45	588.06	451.81	5.26	7.27	2.6	1.95
	3	623.2	132.08	120	91.89	112.5	40.66	860.52	348.16	6.1	4.56	4.7	1.56
	4	635.5	131.82	40	51.63	83.2	23.53	609.78	465.99	1.44	2.74	6.7	1.63
	5	624.1	126.31	10	31.62	42.5	19.3	152.55	318.08	0.26	0.85	7.3	1.49
	6	615.9	138.09	0	0	25.9	14.41	32.13	70.03	0	0	7.4	1.26
	7	615.9	138.09	0	0	15	6.73	6.62	7.63	0	0	7.3	1.33
	8	615.9	138.09	0	0	13.2	4.58	4.14	2.37	0	0	7.3	1.33
	9	615.9	138.09	0	0	10.8	3.88	2.91	1.58	0	0	7.4	1.26
	10	615.9	138.09	0	0	9.8	3.52	2.39	0.98	0	0	7.4	1.26
	11	615.9	138.09	0	0	8.5	3.65	1.96	1.05	0	0	7.4	1.26
	12	615.9	138.09	0	0	7.9	3.84	1.65	0.97	0	0	7.4	1.26
	13	615.9	138.09	0	0	8	9.78	2.02	3.01	0	0	7.3	1.3
	14	615.9	138.09	0	0	5.4	5.44	1.04	1.17	0	0	7.3	1.33
	15	615.9	138.09	0	0	3.6	5.05	0.69	1.11	0	0	7.3	1.33

Figure (3.3) compares the average objective value, running time, iteration count and the average number of patients that are postponed as Γ_l increases for these instances. It can be seen that for $n = 5$ and 10 all instances are solved to optimality. When $n = 15$ some of the instances could not be solved to optimality when $\Gamma_l \in \{2, 3, 4, 5\}$ within the 1,000 second time limit. In our setting, as the number of patients (n) increases, the complexity of the problem increases exponentially. Since the uncertainty for LOS is chosen to be uniform random integers between one and four, the average number of possible realizations of uncertainty when $\Gamma_l = n$ is 2.5^n. It can be seen that the running time for the case of $n = 15$ is much larger than the smaller cases. Our assumption in defining the deviations in LOS for patients are for illustrative purposes. In the case of *more routine* surgeries where the uncertainty in LOS is insignificant, many patients will have $\hat{l} = 0$, thus significantly reducing the size of the uncertainty set $\mathcal{U}_l$.

To better understand the quality of the solution provided by the our proposed model, we need to quantify the impact of a proposed surgery schedule on operational efficiency and quality of service.

3.4.3 Analyzing The Solution Quality

In order to understand and quantify the value of the proposed surgery schedule, we focus on the utilization rate of the SICU beds as an operational metric, where the decision-maker's aim is to keep the utilization rate of these expensive resources high to increase the efficiency of the system.

To analyze the quality of service and risks we focus on two important metrics. First, the probability of not having enough SICU capacity (which is equivalent to the probability of having to transfer a patient to a unit with lower level of care) is calculated. This is similar to probabilistic constraints which guarantee a probability

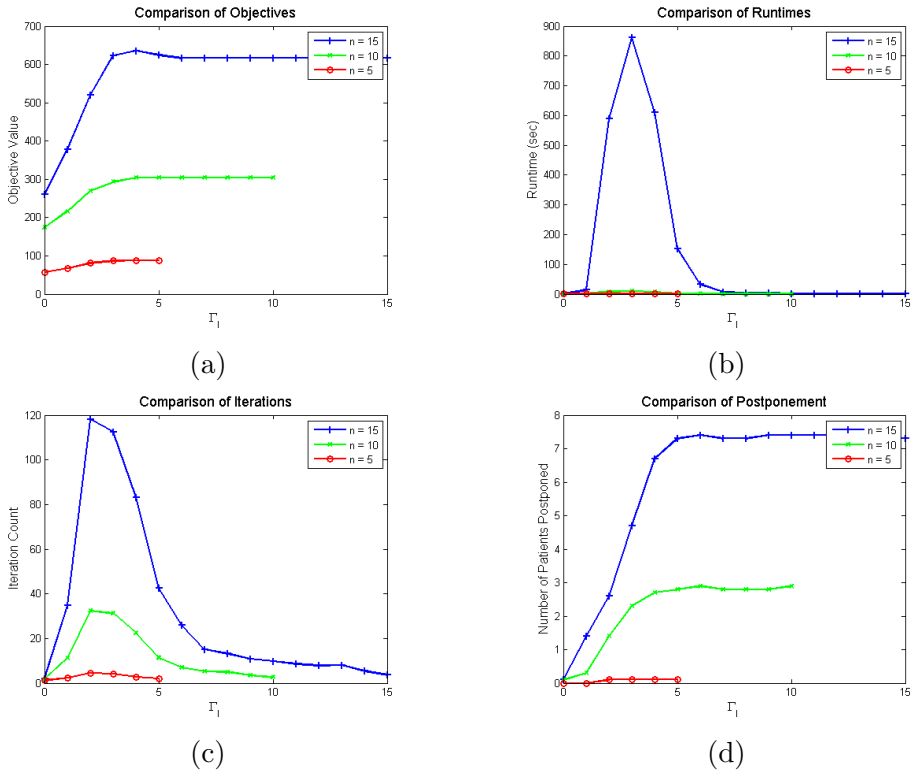


Figure 3.3: Aggregate comparison between the objective value (top left), running time (top right), iterations (bottom left), and the number of postponed patients (bottom right) for $n = 5, 10, 15$.

for feasibility of one or a set of constraints (see [Birge and Louveaux, 2011], chapter 2.7). This is an important metric that helps decision-makers understand the chances of having to reduce the quality of service. In other words, it calculates the *risk* associated with a surgery schedule.

On the other hand, risk alone does not provide enough insight into the recourse actions that are required to be taken. Therefore, it is important to have an idea of how many transfers to lower quality units may be required. In many health care settings, the capacity of the SICU is determined by the number of nurses that are assigned to it. For a decision-maker, adding one nurse to the schedule to cover the expected number of transfers is much more desirable than having to hire five nurses.

This metric calculates the *magnitude of risk* associated with a surgery schedule.

In order to calculate the proposed operational and risk measures, a simulation model is developed. The decision-maker has to decide the value of Γ_l before the realization of uncertainty. In reality, the number of patients whose LOS can deviate from the nominal value can be different from Γ_l. For each fixed value of Γ_l, $n+1$ different cases of deviations can happen.

The simulation model randomly selects $k \in \{0, \ldots, n\}$ patients and generates LOSs that are uniformly distributed between $\bar{l}$ and $\bar{l} + \hat{l}$. Using the proposed surgery schedule from the optimization step, we calculate if a transfer is required and the number of transfers required for that realization. In addition, we calculate the utilization rate for the SICU resources. This process is performed for 200 replications for each value of k (the number of patients that actually deviate from their nominal value). Then, the average probability of transfer, average number of transfers required, and average utilization rate for the SICU is calculated.

Figure (3.4) illustrates the simulation results of previously generated instances of size $n = 5, 10, 15$. All plots show the uncertainty, as the number of patients who deviate from their nominal LOS on the x-axis. The left column of figures, show the probability of having a transfer on the y-axis and each line is associated with a surgery schedule with specific value for Γ_l. It can be seen that as uncertainty increases, meaning that more patients deviate from their nominal LOS, the probability of having a transfer increases. On the other hand, as we increase Γ_l, the probability of not having enough SICU capacity decreases, such that at $\Gamma_l = 3$ for the case of $n = 5$, this probability is almost zero for all cases of deviation. This behavior is repeated for cases with $n = 10$ and 15.

The middle column of Figure (3.4) illustrates the utilization rate of the SICU beds. It can be seen that as the uncertainty increases (moving to the right of the

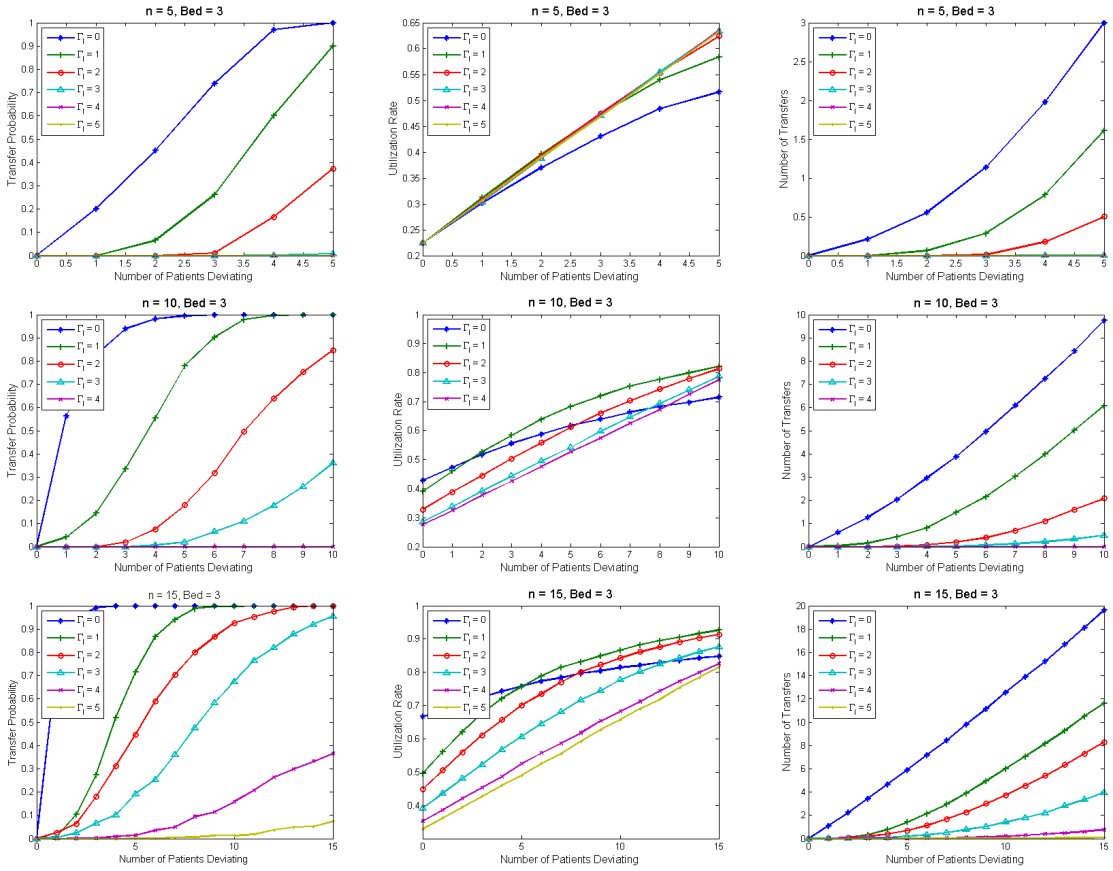


Figure 3.4: Simulation results: Impact of Γ_l and uncertainty on transfer probability, utilization rate, and required transfers.

x-axis), the number of patients who deviate from their nominal LOS and have longer LOSs which translates into higher utilization rate. For $\Gamma_l > 0$, as Γ_l increases, the decision-maker becomes more conservative, thus creating schedules with larger slacks to accommodate uncertainty, which results in lower utilization rate. For the case of $\Gamma_l = 0$, as the uncertainty increases, the increase in the utilization rate is not as fast. This is due to the fact that by choosing $\Gamma_l = 0$, the decision-maker assumes no uncertainty in LOS, thus creating a tight schedule that assigns patients to the days at the beginning of the week while the rest of the week is empty. As uncertainty increases, due to the lack of available beds in the first few days, many transfers are

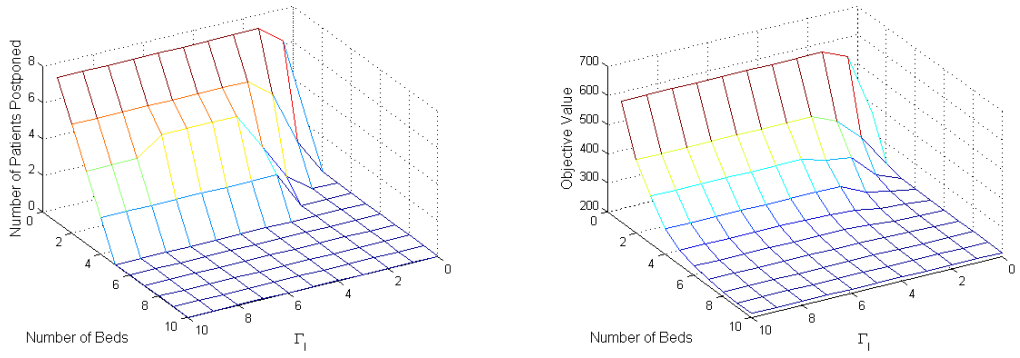
required but the available capacity towards the end of the week remains untouched.

The right column of Figure (3.4) illustrates the average number of transfers that is required for each schedule in the presence of uncertainty. As uncertainty increases, there is greater chance for not having enough SICU beds, thus increasing the number of transfers. However, increasing the value of Γ_l can greatly reduce the number of transfers that are required. For example, in the case of $n = 15$ and $\Gamma_l = 4$, even if all patients deviate from their nominal LOS, there will be, on average, four transfers required.

As previously mentioned, the number of available nurses is an important factor in determining the capacity of the SICU. Note that this can be decided by the manager by designing the nurse schedules. Using our proposed methodology, the decision-maker can determine the staffing levels required to meet the throughput requirement. In other words, the decision-maker can identify the impact of extra resources on scheduling decisions. As an example, we generate an instance with $n = 10$ patients. By changing the number of beds from one to 10, optimal solutions for all values of Γ_l are obtained. To understand the throughput of each schedule, the number of patients that are postponed to have the surgery in future weeks is shown in Figure (3.5).

It can be seen in Figure (3.5a) that as the value of Γ_l increases, more conservative schedules are built, thus more patients get postponed to perform their surgeries later. Increasing the number of beds from one to three will reduce the postponements from eight down to four even in cases where the decision-maker decides to be very conservative (high Γ_l). Looking at the costs of surgery schedules in Figure (3.5b), the decision-maker can decide the best level of staffing for the SICU. In addition, using this method, the decision-maker is capable of understanding the trade-off between the SICU resource level and the throughput of the operating rooms.

We have assumed that the cost of postponing a surgery is equal to the cost of



(a) (b)

Figure 3.5: Impact of SICU capacity on the throughput and cost.

waiting to have a surgery during the next week. A patient that is postponed may not receive a surgery appointment at the beginning of, or during, the next week. Therefore, the decision-maker may require to incur higher costs for patients that are postponed to perform their surgeries later. In order to accommodate this feature, we introduced a multiplier $\gamma \in \{1, \ldots, 10\}$. This multiplier is multiplied by the waiting cost for the patients that are postponed. As the multiplier increases, the cost of postponing a patient increases. Depending on the cost structure, the postponing cost can surpass the cost of having a transfer out of the SICU, in which case, the optimization decides to risk having a transfer. This feature is important for health care providers as postponing a surgery can be very costly at destination medical centers.

As an example, an instance with $n = 10$ patients and three SICU beds is generated. We changed the value of γ from 1 to 10. The case of $\gamma = 1$ is equivalent to our original assumption on the cost structure. We solved this instance for all values of Γ_l and Figure (3.6) shows how the optimization risks having transfers as the cost of

postponing increases. It can be seen that in the case of low downstream capacity, the optimization schedules patients with large values of $\hat{l}$ far away from each other. This is done to minimize the chance of overlapping stays in the SICU which can reduce the number of SICU beds for a long period. This can be used as a rule of thumb for practitioners to mitigate the impact of uncertainty in LOS while scheduling surgeries.

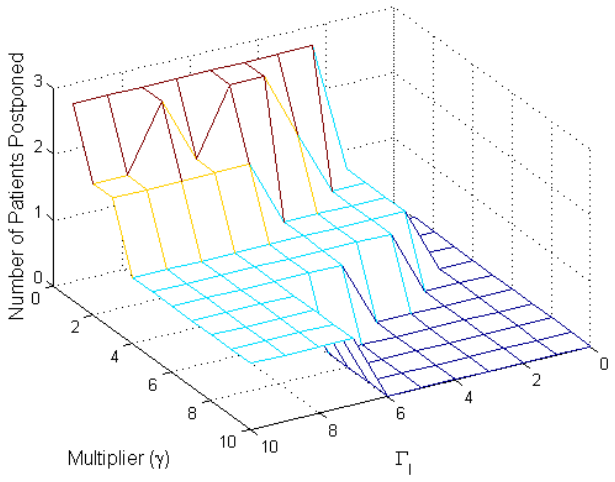


Figure 3.6: Impact of increasing the postponement cost on number of postponements.

We aim to propose a modeling approach that considers the uncertainty in surgery duration and LOS and enables the decision-maker to adjust for her/his risk preferences. In addition, the optimization model, coupled with a simulation model to analyze the quality of solutions (similar to our simulation model), can provide the decision-maker with variety of trade offs that can be made in managing the surgery planning process. Our approach can be used as a dashboard for decision-makers to provide them with different alternatives and the characteristics of each schedule. This will greatly improve the decision making process by assisting the managers with making well-informed and educated decisions.

Figure 3.7 provides an example based on the tests we ran for the case with $n = 15$ patients and three SICU beds.

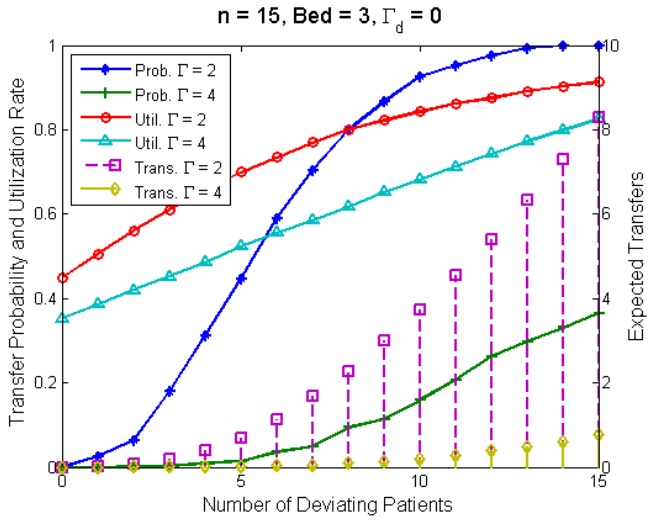


Figure 3.7: Comparing operational and risk metrics between $\Gamma_l = 2$ and $\Gamma_l = 4$.

It can be seen in Figure 3.7, that the probability of having transfers (left y-axis) changes for different risk preferences $\Gamma_l = 2$ and $\Gamma_l = 4$ in the face of uncertainty. If all patients have deviations in their LOS, the probability of requiring a transfer is one at $\Gamma_l = 2$, while it is less than 0.4 if $\Gamma_l = 4$. In terms of the number of transfers required when all patients have deviations (right y-axis), at $\Gamma_l = 2$, more than eight transfers are needed while at $\Gamma = 4$, the required number of transfers is less than one. As for the utilization rate, it can be seen that when all patients have deviation in their LOS, both values for Γ_l have utilization rates greater than 0.8. This is a great example of how our proposed model can help decision-makers choose well-informed alternatives.

3.4.4 Results for larger instances

To test the capabilities of the proposed method, we aimed to solve problems comparable to real cases in terms of size. We modified our data generation scheme based on two observations: (1) not every patient that undergoes surgery requires a stay in the SICU, and (2) not every patient that stays in the SICU has uncertainty associated with his/her LOS. Thus, it is safe to assume that specialties that have smaller average LOS, also have lower chances of having deviations. In other words, uncertainties in the LOS for patients in some specialties are less than others (e.g., ORTHO vs CARDIAC).

In our test generation, the nominal LOS was created using a geometric distribution while the deviation in LOS was randomly generated such that specialties with greater average LOS have higher chance of deviation. In addition, the deviations were perturbed by a random number drawn from $U(0, 2)$ times the standard deviation of the LOS to create different deviations for each patient, and allow for extreme cases where patients in specialties with low average LOS to have high deviations.

Problems were generated with 70 patients and five beds, and 140 patients with 10 SICU beds. For each size, 10 random instances were generated and solved using the proposed method. Run time for each value of Γ_l was restricted to 1000 seconds. In these instances, not every patient has a positive value for $\hat{l}$ and therefore the number of patients that could deviate from their nominal value changes from one instance to another. The budget of uncertainty was changed to consider this fact and a budget of uncertainty $\Gamma_l = 50\%$ means that up to 50% of patients that have positive $\hat{l}$ can have full deviations in their LOS.

In our cases with 70 patients, the average number of patients with positive values for $\hat{l}$ is 22.9 (and maximum of 27), while the same parameter for the case of 140

patients is 46.9 (and maximum of 58). Average values for the objective function, number of iterations, run time (in seconds), optimality gap, and number of patients that are postponed to be scheduled in next decision period is reported in Table 3.5. Optimality gaps are calculated only based on the information from the proposed algorithm, however, tighter gaps can be calculated using the solutions obtained for different values of Γ_l for the same instance. As it can be seen, as the problem size grows the number of instances where optimality is reached decreases ,which is expected, considering the complexity of the problem at hand. On the other hand, the average optimality gap does not exceed 10.5%, which can be considered as good solution quality.

We performed our simulation (1000 replications) analysis to analyze the operational characteristics of the solutions obtained. We have only reported the results where all the patients in the simulation are allowed to deviate from their nominal value. SICU utilization rate, probability of being denied admission to the SICU due to lack of available beds, and average number of beds that is required to cover the lack of capacity are also reported in Table 3.5. Results follow the general trend observed in smaller instances, presented earlier in this section.

3.5 Conclusion

In this chapter, we have proposed a formulation for surgery scheduling while considering the downstream units. We apply two-stage robust optimization to address the inherent uncertainty in surgery duration and length-of-stay in the downstream unit. Since the uncertainty in LOS translates into uncertainty in the number of constraints, a novel modeling approach is proposed to address the challenges in modeling

Table 3.5: Average results for 10 instances of each problem size

Size	Γ_l (%)	Objective	Iter.	Time	Gap (%)	Postpone	Util.	Deny Prob.	Transfer No.
$n = 70$ Bed = 5	0	1776.7	2	1.024	0	2.7	0.613	0.948	4.747
	10	1930.3	56	160.882	0.179	3.8	0.637	0.549	1.283
	20	2135.3	107.4	586.38	2.056	5.3	0.592	0.334	0.625
	30	2284.5	107	759.373	3.819	6.8	0.557	0.166	0.238
	40	2322.8	77.4	451.137	2.595	8.9	0.514	0.045	0.054
	50	2297.4	35.7	135.687	0.82	9.7	0.495	0.009	0.009
	60	2275	18	24.285	0	10.2	0.488	0	0
	70	2275	12.8	7.955	0	10.2	0.488	0	0
	80	2275	15.5	11.903	0	10.2	0.487	0	0
	90	2275	15.9	7.917	0	10.2	0.486	0	0
	100	2275	2.6	1.295	0	10.2	0.486	0	0
$n = 140$ Bed = 10	0	4363.067	2	3.941	0	16.2	0.676	0.77	3.336
	10	4615	46.7	1000	3.095	16.9	0.669	0.291	0.46
	20	4968.933	34.9	1000	7.987	18.4	0.621	0.088	0.113
	30	5163.2	30.5	1000	10.082	19.8	0.592	0.024	0.026
	40	5269.9	30.9	1000	10.455	21	0.571	0.038	0.044
	50	5064.367	34.8	957.336	5.605	22.9	0.533	0.007	0.007
	60	4932	41.7	845.762	2.77	23.2	0.529	0.019	0.021
	70	4886.667	50.1	777.093	1.778	23.7	0.537	0.005	0.005
	80	4877.867	51.1	696.876	1.648	23.8	0.532	0.004	0.004
	90	4849.867	53.4	479.735	0.925	23.9	0.528	0.02	0.022
	100	4810.267	2	2.336	0	24.2	0.522	0	0

this aspect. The proposed formulation can be applied to other domains with similar downstream considerations such as project scheduling. We studied the structural properties of the proposed formulations and reformulated them into solvable MILPs and proposed an exact solution algorithm to solve this problem.

Extensive computational experiments show that this model has the potential of being employed to manage multi-stage care operations. Our simulation model quantifies the impact of our robust model on the utilization of the downstream resources. Our framework, coupled with a simulation model, helps the decision-maker understand the level of risk associated with each proposed surgery schedule and the impact of her/his attitude towards risk. An important insight is that by considering the uncertainty in the LOS, the congestion in the OR can be implicitly alleviated. In addition, the existing trade-offs between different elements in this setting are shown in our computational experiments.

The proposed algorithm may not be efficient for cases with large number of patients with large uncertainty sets. Finding better lower bounds can greatly improve the running time of the proposed algorithm. Effective bounding techniques show promise by using the scenarios generated for different values of Γ can be employed to improve the performance of the algorithm. We hope to continue this development in our future work.

In the next chapter, motivated by the applications in surgery scheduling, we study the two-stage robust generalized assignment problem, formulations, and solution methods.

Chapter 4: The Robust Generalized Assignment Problem

4.1 Introduction

The generalized assignment problem (GAP) is a well-studied subject with many applications in constrained-resource planning problems. The deterministic version of this problem (DGAP) aims to assign jobs to resources in order to optimize an objective function (i.e., either maximizing revenues or minimizing costs), while ensuring that the required capacity for each resource does not exceed the available capacity. This chapter is motivated by the application of scheduling surgeries in surgery blocks or operating rooms. While the duration of the surgeries do not depend on the operating room they are performed in, the generalized assignment problem relaxes this assumption and assumes each job-resource pair has a specific resource requirement. In order to formally develop the formulation for DGAP, we define the parameters as

follows:

i	index of resources, $i = 1, ..., m$
j	index for jobs, $j = 1, ..., n$
R_i	amount of resource i available, $i = 1, ..., m$
r_{ij}	amount of resource i needed by job j if assigned, $i = 1, ..., m, j = 1, ..., n$
c_{ij}	cost of assigning resource i to job j, $i = 1, ..., m, j = 1, ..., n$.

The decision variable is defined for all resource-job pairs as follows:

$$x_{ij} = \begin{cases} 1 & \text{if job } j \text{ is assigned to resource } i \\ 0 & \text{otherwise.} \end{cases}$$

Considering the definitions, DGAP can be formulated as follows:

$$\text{DGAP} : \min \quad \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{4.1a}$$

$s.t.$

$$\sum_{i=1}^{m} x_{ij} = 1 \qquad \forall j \tag{4.1b}$$

$$\sum_{j=1}^{n} r_{ij} x_{ij} \leq R_i \qquad \forall i \tag{4.1c}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j \tag{4.1d}$$

Objective function (4.1a) is the minimization of the assignment costs. Constraint (4.1b) ensures that each job is assigned to exactly one resource. Constraint (4.1c) ensures that for each resource, the total required capacity by the jobs assigned to that machine, does not exceed the available capacity. Constraint (4.1d) defines the binary domain of the decision variables.

DGAP is known to be an NP-hard problem. Iterative approaches, such as Lagrangian Relaxation or branch-and-price are used to solve problems with large numbers of jobs and resources [Savelsbergh, 1997].

All the parameters of DGAP are deterministic and known with certainty. However, in practical cases not all the information is likely to be known with certainty. In order to motivate this modeling approach, consider the case of assignment of surgeries to surgery blocks in a hospital. In many hospitals, blocks of time are assigned to each specialty/surgeon in order to perform their surgeries in their allocated block. A given day typically have multiple blocks scheduled to an OR. There is a cyclic block

schedule for all the specialties that offer surgeries in a hospital. Each block has a specific day and a defined length. This is known as the *block booking policy*. Under the *block booking policy*, patients with specific surgery requirements can be assigned to a subset of these blocks, however, the length of the surgery for each patient is not known with certainty. We assume each patient, based on his/her conditions, can have a nominal surgery duration that can be provided by the physician or historical data. Also, considering the known complications that can occur during the surgery, the physician can also provide a maximum length of time that may be needed to perform the surgery. The goal is to assign surgeries to blocks in order to minimize the costs of operations and increase utilization of the resources. On the other hand, basing assignment decisions only on the nominal durations can push the surgery blocks into overtime or surgery cancellations. In other words, if the uncertainty is not considered to make assignment decisions, the required length of time to perform surgeries within a block can exceed the block length (capacity) which can cause delays and cancellations in the subsequent surgeries and can possibly cause patient discomfort or health issues. Further, surgery rooms are staffed with highly trained workers and the overtime costs can be very high.

The goal is to generate a surgery plan that considers the uncertainties in the surgery durations while taking into account the level of conservatism of the decision makers to minimize the costs of patient admission and improve the care quality by reducing delays or cancellations.

Stochastic programming (SP) techniques have been used to address uncertain parameters in GAP. However, exact and known distributional information is required for the use of stochastic programming techniques. In addition, in the case of two-stage SP,usually a large number of scenarios is required to characterize the uncertainty which can pose intractability due to the large size of the problem or prohibitive

number of sub-problems to be solved.

We choose to model the generalized assignment problem with uncertain resource requirements as a two-stage robust optimization model. Our model does not require any information on the distribution of the random parameters, but rather only requires lower and upper bounds on the possible values for the uncertain parameters. In other words, each parameter is required to belong to a known set or range of values.

4.2 Modeling Two-Stage Robust Generalized Assignment Problem

In the two-stage robust generalized assignment model (2RoGAP) the values for the resource required by each job r_{ij} to be uncertain while belonging to a known set. The uncertainty exists in the technology matrix of the second constraint in DGAP.

This problem can be viewed as a two-stage process in which decisions to assign jobs to resources (or patients to surgery blocks) (x_{ij}) are made in the first stage. Next, uncertainty in the resource requirement (surgery duration) is realized. In the second stage, the goal is to minimize the defined worst-case scenario for the costs of not having enough resources. We define o_i as the amount of resource i required beyond the available capacity. Also, b_i is the penalty for each unit of the resource i that cannot be allocated to a job. In practical cases, o_i represents the amount of extra resource required and in the case of surgery planning, it is representing the overtime needed by a surgery block.

We assume that only a subset of uncertain parameters will deviate from their nominal value and try to minimize the worst-case costs when some subset of jobs (surgeries) deviate to their maximum value in resource requirements. Let us define $\tilde{r}_{ij} \in [\overline{r}_{ij}, \overline{r}_{ij} + \hat{r}_{ij}]$, in which $\overline{r}_{ij}$ is the nominal value for the resource required for

job j if it is assigned to resource i, while $\hat{r}_{ij}$ is the total deviation from the nominal value that the resource requirement can have. Next, we define $z_{ij} = \frac{\tilde{r}_{ij} - \overline{r}_{ij}}{\hat{r}_{ij}}$ as the normalized deviation of the resource requirement for job j when assigned to resource i. Note that $0 \leq z_{ij} \leq 1$ holds. Following the notation defined by [Bertsimas and Sim, 2004], we define Γ as the *budget of uncertainty* for surgery resource requirements. By enforcing $\sum_{i=1}^{m} \sum_{j=1}^{n} z_{ij} \leq \Gamma$, we limit the total possible normalized deviation from the nominal value being less than the budget of uncertainty Γ. In other words, if Γ is integer-valued, only Γ of jobs can have resource requirements equal to their highest possible resource usage, $\overline{r}_{ij} + \hat{r}_{ij}$. Note that the resource requirement r_{ij} can only be positive if job j is assigned to the resource i. Therefore following inequality should is required to enforce this condition:

$$z_{ij} \leq x_{ij}. \qquad \forall i, j \tag{4.2}$$

We define the uncertainty set $\mathcal{U}$ which is the set of all the possible realizations for resource requirements as follows:

$$\mathcal{U} = \{r \in \mathbb{R}^n : \tilde{r}_{ij} = \overline{r}_{ij} + z_{ij}\hat{r}_{ij}, 0 \leq z_{ij} \leq 1, z_{ij} \leq x_{ij}, \sum_{i=1}^{m} \sum_{j=1}^{n} z_{ij} \leq \Gamma\}. \tag{4.3}$$

Considering the definitions for the uncertainty model presented above, the formulation for the Two-Stage Robust Generalized-Assignment Problem (2RoGAP) can be written as follows:

$$\min \quad \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + opt[R(x,\Gamma)] \tag{4.4a}$$

$s.t.$

$$\sum_{i=1}^{m} x_{ij} = 1 \qquad \forall j \tag{4.4b}$$

$$x_{ij} \in \{0,1\} \qquad \forall i,j \tag{4.4c}$$

in which $opt[R(x,\Gamma)]$ is the optimal solution of the recourse problem $R(x,\Gamma)$ and can be written as:

$$\max_{r\in\mathcal{U}} \quad \min \sum_{i=1}^{m} b_i o_i \tag{4.5a}$$

$s.t.$

$$\sum_{j=1}^{n} \tilde{r}_{ij}x_{ij} \leq R_i + o_i \qquad \forall i \tag{4.5b}$$

$$o_i \geq 0 \qquad \forall i \tag{4.5c}$$

Objective function (4.4a) is to minimize the cost of assignment as well as the second-stage cost which is the minimization of the worst-case recourse costs. Constraint (4.4b) makes sure that each job is assigned to exactly one resource. The second-stage objective function (4.5a) is the minimization of the worst-case recourse costs. Note that b_i is the unit penalty/cost for a lack of resource i and o_i is the amount of shortage of resource i. Constraint (4.5b) calculates the amount of shortage

based on the realization of the usage parameter $\tilde{r}_{ij}$ and assignment decision x_{ij}.

Based on the definition of the uncertainty set $\mathcal{U}$, we have:

$$\max_{\sum_{i=1}^{m}\sum_{j=1}^{n} z_{ij} \leq \Gamma, z_{ij} \leq x_{ij}} \quad \min \sum_{i=1}^{m} b_i o_i \tag{4.6a}$$

$$s.t.$$

$$\sum_{j=1}^{n} (\overline{r}_{ij} + z_{ij}\hat{r}_{ij}) x_{ij} \leq R_i + o_i \qquad \forall i \tag{4.6b}$$

$$o_i \geq 0 \qquad \forall i \tag{4.6c}$$

Note that we replaced the uncertain parameter $\tilde{r}_{ij}$ by its definition from $\mathcal{U}$. Variables z_{ij} are in the outer-maximization part of the second-stage and are only known parameters to the inner-minimization. x_{ij} are variables in the first stage problem (4.4a)-(4.4c), and they become known parameters to the second-stage problem.

It can be seen that the inner-minimization problem (4.6a)-(4.6c) is a linear program and we can replace the inner-minimization problem by its dual problem and

combine the objective functions and rewrite the recourse problem $R(x, \Gamma)$ as follows:

$$\max \quad \sum_{i=1}^{m} \left[\sum_{j=1}^{n} (\overline{r}_{ij} + z_{ij}\hat{r}_{ij}) x_{ij} - R_i \right] \pi_i \tag{4.7a}$$

s.t.

$$\sum_{i=1}^{m} \sum_{j=1}^{n} z_{ij} \leq \Gamma \tag{4.7b}$$

$$z_{ij} \leq x_{ij} \qquad \forall i, j \tag{4.7c}$$

$$0 \leq \pi_i \leq b_i \qquad \forall i \tag{4.7d}$$

$$0 \leq z_{ij} \leq 1. \qquad \forall i, j \tag{4.7e}$$

Note that variable π_i is the dual variable associated with capacity constraint for the resource i. Due to the existence of the bilinear term $\sum_{i=1}^{m} \sum_{j=1}^{n} \hat{r}_{ij} x_{ij} z_{ij} \pi_i$ the second-stage problem $R(x, \Gamma)$ is the maximization of a bilinear function over a linear constraint. Since $x_{ij} \in \{0, 1\}$, the constraint (4.7e) is redundant. Bilinear programming is a special case of quadratic programming problems and the objective function, in general, is neither convex or concave [Gallo and Ülkücü, 1977]. This is a limiting factor in using standard convex optimization solvers to obtain optimal solutions to the second-stage problem. Proposition 4.2.1, enables us to reformulate the second-stage problem $r(x, \Gamma_r)$ as a mixed-integer linear program (MILP).

Proposition 4.2.1. *If the budget of uncertainty* Γ *is an integer number, there exists and optimal solution* (π^*, z^*) *to the second-stage problem* $R(x, \Gamma)$ *such that* $z_{ij}^* \in \{0, 1\}, \forall i, j$.

Proof. Let us define the feasible region for the second-stage problem by the following polyhedra $\Pi = \{\pi \in \mathbb{R}^m | 0 \leq \pi_i \leq b_i, \forall i\}$ and $\mathcal{Z}(\Gamma, x) = \{z \in \mathbb{R}^m \times$

$\mathbb{R}^n | \sum_{i=1}^{m} \sum_{j=1}^{n} z_{ij} \leq \Gamma, 0 \leq z_{ij} \leq x_{ij}, \forall i, j\}$. Note that both sets Π and $\mathcal{Z}(\Gamma, x)$ are clearly bounded (all variables are bounded) and therefore and optimal solution (π^*, z^*) exists such that π^* is an extreme point of Π and z^* is an extreme point of $\mathcal{Z}(\Gamma, x)$ (see [Gallo and Ülkücü, 1977] for detailed discussion). On the other hand, the vector for variable x is composed of binary elements defined in the first stage of the problem. This implies when Γ is an integer number, then $z^* \in \{0,1\}^{m \times n}$ (see [Gabrel et al., 2014a]).

□

In the light of 4.2.1 we can reformulate the bilinear second-stage problem $R(x, \Gamma)$ as an MILP $R_L(x, \Gamma)$ by defining $p_{ij} = \pi_i z_{ij}, \forall i, j$ as follows:

$$\max \quad \sum_{i=1}^{m} \sum_{j=1}^{n} \overline{r}_{ij} x_{ij} \pi_i + \sum_{i=1}^{m} \sum_{j=1}^{n} \hat{r}_{ij} x_{ij} p_{ij} - \sum_{i=1}^{m} R_i \pi_i \tag{4.8a}$$

$s.t.$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} z_{ij} \leq \Gamma \tag{4.8b}$$

$$z_{ij} \leq x_{ij} \qquad \forall i, j \tag{4.8c}$$

$$0 \leq \pi_i \leq b_i \qquad \forall i \tag{4.8d}$$

$$p_{ij} \leq \pi_i \qquad \forall i, j \tag{4.8e}$$

$$p_{ij} \leq b_i z_{ij} \qquad \forall i, j \tag{4.8f}$$

$$p_{ij} \geq 0, z_{ij} \in \{0,1\}. \qquad \forall i, j \tag{4.8g}$$

In the next section we present Kelley's cutting plane algorithm in order to solve 2RoGAP as well as some results based on the structural properties of the second-stage

problem $R_L(x, \Gamma)$.

4.2.1 Alternate Formulation

In this section, we propose another formulation for the second-stage problem that has fewer number of variables which can increase the efficiency of the solution time for second-stage problem.

Proposition 4.2.2. *The following formulation is an equivalent formulation for the second-stage problem $R_L(x, \Gamma)$.*

$$\max \quad \sum_{i=1}^{m}\sum_{j=1}^{n} \overline{r}_{ij} x_{ij} \pi_i + \sum_{i=1}^{m}\sum_{j=1}^{n} \hat{r}_{ij} x_{ij} p_{ij} - \sum_{i=1}^{m} R_i \pi_i \tag{4.9a}$$

s.t.

$$\sum_{j=1}^{n} z_j \leq \Gamma \tag{4.9b}$$

$$0 \leq \pi_i \leq b_i \qquad \forall i \tag{4.9c}$$

$$p_{ij} \leq \pi_i \qquad \forall i, j \tag{4.9d}$$

$$p_{ij} \leq b_i z_j \qquad \forall i, j \tag{4.9e}$$

$$p_{ij} \geq 0, z_j \in \{0, 1\} \qquad \forall i, j \tag{4.9f}$$

Proof. In the original formulation, due to the inequalities $z_{ij} \leq x_{ij}, \forall i, j$ and $\sum_{i=1}^{m} x_{ij} = 1, \forall j$, for any given resource i, only one variable z_{ij} can be equal to one. In other words, since the assignment of jobs to resources are already known, we can reduce the second-stage to finding which jobs are going to deviate to cause the largest over-time cost. Therefore, we can reduce the formulation to include only z_j as decision variables. □

4.3 Solution and Structural Properties

Considering the development of the second-stage resource capacity sub-problem, the information in the form of worst-case realization of the resource requirements for a given assignment from the sub-problem can be passed to the master problem. The restricted master problem that does not include all the possible realizations of the uncertain parameters, provide a lowerbound on the optimal solution of 2RoGAP. . Let $\mathcal{W}$ be the set of all the feasible solutions to the MILP recourse problem $R_L(x, \Gamma)$. Since the objective function is linear, the optimal solution is an extreme point of $conv(\mathcal{W})$ which is the convex hull for the set of feasible solutions. This allows us to formulate the two-stage robust generalized-assignment master problem $(2RoGap-m)$ as follows:

$$\min \quad \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \theta \tag{4.10a}$$

$s.t.$

$$\theta \geq \sum_{i=1}^{m}\sum_{j=1}^{n} \overline{r}_{ij}x_{ij}\pi_i^k + \sum_{i=1}^{m}\sum_{j=1}^{n} \hat{r}_{ij}x_{ij}\pi_i^k z_{ij}^k - \sum_{i=1}^{m} R_i\pi_i^k \qquad k = 1, .., K \tag{4.10b}$$

$$\sum_{i=1}^{m} x_{ij} = 1 \qquad \forall j \tag{4.10c}$$

$$x_{ij} \in \{0, 1\}, \theta \geq 0 \qquad \forall i, j, k \tag{4.10d}$$

where $(p_{ij}^k, \pi_i^k, z_{ij}^k)$ and $p_{ij}^k = \pi_i^k z_{ij}^k$ with $k = 1, ..., K$ are the extreme points of the $conv(\mathcal{W})$ for fixed values of first-stage decisions x, y and Γ. Therefore, we have an optimization problem with a linear objective function and an exponential number of constraints. On the other hand, in order to obtain an extreme point of $conv(\mathcal{W})$ we

need to solve $R_L(x, \Gamma)$ which is an MILP that in general, is an NP-hard problem and requires Branch-and-Bound based($B\&B$) method to obtain a solution.

Now, to employ Kelley's algorithm, the general idea is to iteratively generate new extreme points of $conv(\mathcal{W})$ by solving the recourse problem $R_L(x, \Gamma)$, and add them to the master problem ($2RoGAP - m$) until the optimality conditions are satisfied.

The optimal value for $2RoGAP - m$ is a lower bound for the the optimal value of the robust problem (2RoGAP), since it only contains a subset of the constraints (cuts). Therefore, as the constraints are added to the master problem, the value for the lower bound, L, is going to be non decreasing. As for the upper bound, at each iteration k, since the solution vector (x^k, π^k, p^k) is feasible for $2RoGAP - m$, the upper bound value U, is the minimum solution value over all generated solutions up to iteration k. In addition, as the number of extreme points to the $conv(\mathcal{W})$ is bounded (although it has exponential number of extreme points), the algorithm stops within a finite number of iterations.

We rely on the strength of the formulation and the efficiency of the solver in handling the second-stage problem. The cutting-plane algorithm for solving 2RoGAP is presented by Algorithm 2.

As it can be seen in Algorithm 2, at each iteration the second stage problem $R_L(x, \Gamma)$, which is an MILP, has to be solved to optimality in order to obtain an extreme point of the convex hull of its feasible region. We can explore the structural properties of the second-stage problem $R(x, \Gamma)$ in order to increase the efficiency of the algorithm.

Structure of the objective and valid inequalities Considering the objective function (4.7a) has the maximization of $\sum_{i=1}^{m} \left[\sum_{j=1}^{n} (\overline{r}_{ij} + z_{ij}\hat{r}_{ij})x_{ij} - R_i\right] \pi_i$, two different conditions can happen for each choice of variable x which is captured by the following proposition.

Algorithm 2: Kelly's Algorithm for 2RoGAP

Data: $c, \Gamma, m, n, b, \epsilon$

Initialization;

define $2RoGAP - m$ containing one extreme point (p^0, π^0) such that $p^0 = \pi^0 = 0$;

set $L \leftarrow 0, U \leftarrow +\infty, k \leftarrow 1$ and go to Master routine;

Master: Solve master problem $2RoGAP - m$;

$$2RoGAP - m : \min \quad \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \theta$$

subject to

$$\theta \geq \sum_{i=1}^{m}\sum_{j=1}^{n} \overline{r}_{ij}x_{ij}\pi_i^l + \sum_{i=1}^{m}\sum_{j=1}^{n} \hat{r}_{ij}x_{ij}p_{ij}^l - \sum_{i=1}^{m} R_i\pi_i^l \qquad l = 1, ..., k-1$$

$$\sum_{i=1}^{m} x_{ij} = 1 \qquad \forall j$$

$$x_{ij} \in \{0, 1\}, \theta \geq 0 \qquad \forall i, b, t$$

and have (x^k, θ^k) as its optimal solution;

Update $L \leftarrow \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}^k + \theta^k$ and go to Recourse routine;

Recourse: For fixed values x^k, solve the recourse problem $R_L(x^k, \Gamma)$ and have (p^k, π^k, z^k) as its optimal solution;

Set $U \leftarrow \min\{U, \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}^k + \sum_{i=1}^{m}\sum_{j=1}^{n} \overline{r}_{ij}x_{ij}\pi_i^k + \sum_{i=1}^{m}\sum_{j=1}^{n} \hat{r}_{ij}x_{ij}p_{ij}^k - \sum_{i=1}^{m} R_i\pi_i^k\}$;

if $U - L \leq \epsilon$ **then**

  (x^k, θ^k) is the optimal solution for 2RoGAP;

else

  go to Add-Cut routine;

end

Add-Cut: Add the following constraint using (p^k, π^k, z^k) obtained from Recourse step:

$$\theta \geq \sum_{i=1}^{m}\sum_{j=1}^{n} \overline{r}_{ij}x_{ij}\pi_i^k + \sum_{i=1}^{m}\sum_{j=1}^{n} \hat{r}_{ij}x_{ij}\pi_i^k z_{ij}^k - \sum_{i=1}^{m} R_i\pi_i^k$$

to the master problem $2RoGAP - m$ and $k \leftarrow k + 1$ and go to Master Routine;

Proposition 4.3.1. *For any $i = 1, .., m$ in the second-stage optimal solution, $\pi_i \in \{0, b_i\}$.*

Proof. Due to the structure of the objective function one of the following cases is true for any $\pi_i, i = 1, .., m$:

- The case that for a given resource i and assignment vector x and deviation vector z, $\sum_{j=1}^{n}(\overline{r}_{ij} + \hat{r}_{ij}z_{ij})x_{ij} - R_i > 0$. In this case, due to maximization of the objective, π_i will assume its upperbound b_i.

- The case that for a given resource i and assignment x and deviation vector z, $\sum_{j=1}^{n}(\overline{r}_{ij} + \hat{r}_{ij}z_{ij})x_{ij} - R_i \leq 0$. This means that given the assignment and the deviations for resource consumption parameters, resource consumption will not exceed the available capacity R_i. In this case, due to maximization of the objective, $\pi_i = 0$.

□

The second-stage problem, $R(x, \Gamma)$, seeks to maximize the cost of not having enough capacity for each resource, considering that the number of jobs that can assume their maximum resource-consumption parameter is bounded by the budget of uncertainty, Γ. Note that Proposition 4.2.1 states that at optimality, each resource consumption parameter can only deviate fully to its maximum value and partial deviations potentially produce sub-optimal solutions. In other words, the second-stage problem is to find a subset of assigned jobs, J_s, such that $|J_s| \leq \Gamma$, and their resource consumption maximizes the cost of not having enough resources. The following proposition presents the valid inequalities for the second-stage problem in the forms of optimality-cuts. These cuts are used to cut off sub-optimal solutions from the feasible region of the problem.

Proposition 4.3.2. *For any given pair $(j', j) \in J \times J$ and resource $i \in I$, such that $x_{ij'} = x_{ij} = 1$ and $\hat{r}_{ij'} \leq \hat{r}_{ij}$, the inequality $z_{ij'} \leq z_{ij}$ is a valid inequality for the optimal solution to the second-stage problem.*

Proof. Consider the simple case of single resource and two jobs $j = 1, 2$, and the budget of uncertainty $\Gamma = 1$. Assume $\hat{r}_1 \leq \hat{r}_2$ and $\overline{r}_1 + \overline{r}_2 \geq R$, i.e., there is lack of resource. Obviously, both jobs are assigned to the only existing resource and since there is lack of resource we have $\pi = b$. The second-stage problem can be written as follows:

$$\max \quad [(\overline{r}_1 + \hat{r}_1 z_1) + (\overline{r}_2 + \hat{r}_2 z_2) - R]\, \pi$$

subject to

$$z_1 + z_2 \leq \Gamma$$

$$0 \leq \pi \leq b$$

$$z_1, z_2 \in \{0, 1\}$$

in which the objective function can be rewritten as $[\overline{r}_1 + \overline{r}_2 + \hat{r}_1 z_1 + \hat{r}_2 z_2 - R]\, \pi$. Since $\Gamma_r = 1$ and $\hat{r}_1 \leq \hat{r}_2$, the inequality $f_1 = [\overline{r}_1 + \overline{r}_2 + \hat{r}_1 - R]b < [\overline{r}_1 + \overline{r}_2 + \hat{r}_2 - R]b = f_2$ holds. f_1 is the objective value when $z_1 = 1$ and $z_2 = 0$, while f_2 is the objective value for $z_1 = 0, z_2 = 1$. It can be seen that inequality $z_1 \leq z_2$ is a valid optimality-cut for this problem since it does remove the sub-optimal solution $(z_1 = 1, z_2 = 0)$ and does not cut the optimal value $(z_1 = 0, z_2 = 1)$.

In order to prove the proposition, we need to show that the feasible solutions in which the inequality does not hold, are not optimal. Consider the set J_i as the set of jobs that are assigned to the resource $i = 1, .., m$, in a given feasible solution (x, z, π). Consider $(j_1, j_2) \in J_i$ such that $\hat{r}_{ij_1} \leq \hat{r}_{ij_2}$ and $z_{ij_1} = 1$ and $z_{ij_2} = 0$, Thus, $z_{ij_1} \nleq z_{ij_2}$.

The proposed inequality does not hold for this feasible solution. The value of the objective function can be written as $f_1 = \sum_{i=1}^{m} \sum_{j \in J_i \setminus \{j_1, j_2\}} \left[(\overline{r}_{ij} + \hat{t}_{ij} z_{ij}) + \hat{r}_{ij_1} - R_i \right] \pi_i$. Since $\hat{r}_{ij_1} \leq \hat{r}_{ij_2}$, the inequality $f_1 \leq \sum_{i=1}^{m} \sum_{j \in J_i \setminus \{j_1, j_2\}} \left[(\overline{r}_{ij} + \hat{t}_{ij} z_{ij}) + \hat{r}_{ij_2} - R_i \right] \pi_i = f_2$ holds. f_2 is the objective value for a feasible solution (x, z', π) in which all the values for z' is equal to the components of z with the exception of $z'_{ij_1} = 0$ and $z'_{ij_2} = 1$. Clearly this solution is still feasible since the sum of all deviations has not changed. However, this new solution improves the objective function and therefore $z_{ij_1} \leq z_{ij_2}$ is valid inequality for the optimal solution. □

It can be seen from the Proposition 4.3.2, that the cuts should be added to the second-stage problem dynamically. This means that after each time of solving the master problem $2RoGAP - m$, we need to identify the set of jobs that are assigned to each resource $i = 1, .., m$, namely J_i, based on the assignment decision variables x. The jobs assigned to resource i need to be sorted based on the value for $\hat{r}_{ij}$, and the valid inequality can be added to increase the efficiency of the solution of second-stage problem. At the next iteration, upon obtaining new solution for x, cuts that were added in the previous iteration have to be removed and new cuts based on the parameters have to be generated. This process can be tedious and in each iteration we need to sort the values for the deviation parameters that are assigned to each resource, which can adversely affect the performance of our algorithm. The following proposition introduces valid inequalities for the optimal solution of the second-stage problem that can be introduced in the original formulation of $R_L(x, \Gamma)$.

Proposition 4.3.3. *For any given pair $(j', j) \in J \times J$ and resource $i \in I$, such that $\hat{r}_{ij'} \leq \hat{r}_{ij}$, the inequality $z_{ij'} \leq z_{ij} + (1 - x_{ij})$ is a valid inequality for the optimal solution to the second-stage problem.*

Proof. For a given resource i and pair of jobs (j', j) such that $\hat{r}_{ij'} \leq \hat{r}_{ij}$, one of the following cases is true:

- **Case 1-** If both jobs are assigned to resource i which means $x_{ij'} = x_{ij} = 1$. In this case the proposed inequality is reduced to $z_{ij'} \leq z_{ij}$.

- **Case 2-** If j' is assigned to i while job j is not assigned to i, we have $x_{ij'} = 1, x_{ij} = 0$. In this case the proposed inequality reduces to $z_{ij'} \leq 1$, which is true for all z.

- **Case 3-** If j' is not assigned to i while job j is assigned to i, we have $x_{ij'} = 0, x_{ij} = 1$. Based on the formulation of the second-stage problem, $z_{ij'} \leq x_{ij'}$, which fixes the value of $z_{oj'}$ to 0. The proposed inequality reduces to $z_{ij} \geq 0$, which is true for all z.

□

Proposition 4.3.3 can be used to reformulate the second-stage problem to include the optimality cuts in the original formulation. This formulation requires a slightly different definition on the problem parameters. In order to present the formulation, we need to sort the deviation parameter $\hat{r}_{ij}$ for each resource $i = 1, .., m$ in ascending

order. The formulation is as follows:

$$\max \quad \sum_{i=1}^{m}\sum_{j=1}^{n}\overline{r}_{ij}x_{ij}\pi_i + \sum_{i=1}^{m}\sum_{j=1}^{n}\hat{r}_{ij}x_{ij}p_{ij} - \sum_{i=1}^{m}R_i\pi_i \tag{4.11a}$$

s.t.

$$\sum_{i=1}^{m}\sum_{j=1}^{n}z_{ij} \leq \Gamma_r \tag{4.11b}$$

$$z_{ij} \leq x_{ij} \qquad \forall i,j \tag{4.11c}$$

$$z_{ij} \leq z_{ik} + (1 - x_{ik}) \qquad \forall i,j,k > j \tag{4.11d}$$

$$0 \leq \pi_i \leq b_i \qquad \forall i \tag{4.11e}$$

$$p_{ij} \leq \pi_i \qquad \forall i,j \tag{4.11f}$$

$$p_{ij} \leq b_i z_{ij} \qquad \forall i,j \tag{4.11g}$$

$$p_{ij} \geq 0, z_{ij} \in \{0,1\} \qquad \forall i,j \tag{4.11h}$$

in which the only difference is adding constraint (4.11d) which is derived in Proposition 4.3.3. Note that the new formulation has $m\left[\frac{n(n-1)}{2}\right]$ more constraints than the original second-stage problem $R_L(x,\Gamma)$. The hope is that by adding these constraints, the second-stage problem solution will not require large number of branch-and-bound iterations and can be more effectively handled especially when we are using a standard branch-and-bound algorithm. For example, in problems with the introduced optimality cuts, at each node of the branch-and-bound tree, fixing any z_{ij} variable will fix all other variables $z_{ij'}$ such that $x_{ij'} = 1$.

4.3.1 Column-&-Constraint Generation (C&CG) Method

In this section we present the C&CG method introduced by [Zeng and Zhao, 2013] to solve the 2RoGAP. The idea behind the C&CG method stems from the fact that any robust optimization problem can be formulated as a large-scale optimization problem. Consider set $\mathcal{Z}$ as the set of all the possible realizations for the variable z for resource consumption from the uncertainty set $\mathcal{U}_r$. The two-stage robust generalized assignment problem can be formulated as follows:

$$\min \quad \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \theta \tag{4.12a}$$

$s.t.$

$$\sum_{i=1}^{m} x_{ij} = 1 \qquad \forall j \tag{4.12b}$$

$$\sum_{j=1}^{n} (\overline{r}_{ij} + z_{ij}^{k}\hat{r}_{ij})x_{ij} \leq R_i + o_i^k \qquad \forall i,k \tag{4.12c}$$

$$\theta \geq \sum_{i=1}^{m} b_i o_i^k \qquad \forall k \tag{4.12d}$$

$$x_{ij} \in \{0,1\}, \theta, o_i^k \geq 0. \qquad \forall i,j,k \tag{4.12e}$$

The objective function (4.12a) aims to minimize the assignment cost plus the worst-case overage costs. The first constraints (4.12b) make sure each job is assigned to a machine. The second constraints (4.12c) makes sure that for each resource i, and under specific scenario k the resource consumed is below the available amount or the overage is captured. Note that $0 \leq z_{ij}^k \leq 1$ will determine what value for resource consumption for job j on resource i, which is bounded by $\overline{r}_{ij}$ and $\overline{r}_{ij} + \hat{r}_{ij}$. The third constraints (4.12d) capture the overage cost under each scenario. The last constraints

Algorithm 3: C&CG Algorithm for 2RoGAP

Data: $c, \Gamma, m, n, b, \epsilon$

Initialization;

define $2RoGAP - CCG$ containing one extreme point (p^0, π^0) such that $p^0 = \pi^0 = 0$;

set $L \leftarrow 0, U \leftarrow +\infty, K = \{1\}, iter \leftarrow 1, z^k = 0$ and go to Master routine;

Master: Solve master problem $2RoGAP - CCG$;

$$\min \quad \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \theta$$

$$s.t.$$

$$\sum_{i=1}^{m} x_{ij} = 1 \qquad \forall j$$

$$x_{ij} \in \{0,1\}, \theta \geq 0 \qquad \forall i, j, k$$

and have (x^l, θ^l) as its optimal solution;

Update $L \leftarrow \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}^l + \theta^l$ and go to Recourse routine;

Recourse: For fixed values x^l, solve the recourse problem (4.8a)-(4.8g) $R_L(x^l, \Gamma)$ and have (z^l, p^l, π^l) as its optimal solution;

Set $U \leftarrow$ $\min\{U, \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}^l + \sum_{i=1}^{m}\sum_{j=1}^{n} \overline{r}_{ij}x_{ij}^l\pi_i^l + \sum_{i=1}^{m}\sum_{j=1}^{n} \hat{r}_{ij}x_{ij}^l p_{ij}^l - \sum_{i=1}^{m} R_i\pi_i^l\}$;

if $U - L \leq \epsilon$ **then**

  (x^l, θ^l) is the optimal solution for 2RoGAP;

else

  go to Add Columns-and-Constraints routine;

end

Add Columns-and-Constraints: Add variables o_i^k and constraints

$$\sum_{j=1}^{n}(\overline{r}_{ij} + z_{ij}^l\hat{r}_{ij})x_{ij} \leq R_i + o_i^k \qquad \forall i$$

$$\theta \geq \sum_{i=1}^{m} b_i o_i^k$$

$$o_i^k \geq 0 \qquad \forall i$$

to the master problem $2RoGAP - m$, set $k \leftarrow k + 1$ and go to Master Routine;

(4.12e), define the domain for decision variables.

This is a large-scale optimization problem since the number of possible realizations for the vectors of deviations is very large. Therefore, a two-stage formulation is proposed where the second-stage acts as the scenario-generation step and provide upper bounds on the solution. The main difference between the cutting-plane (CP) methods such as the L-shaped method and C&CG is the form of the master problem. In CP methods, as shown in the previous section, at each iteration one constraint in the form of the objective function of the dual of the second-stage is added to the master problem. In C&CG, after realization of uncertain parameters, a set of constraints in the form of the original deterministic formulation of the problem is added to the master. Algorithm 3 presents the steps required to solve this problem.

Stronger Constraints for C&CG

As described previously, when the assignment of jobs to resources is known, the second-stage problem aims to find out which jobs have deviations, rather than finding which job-resource pair has a deviations. Thus, we can redefine the second-stage to include deviation variables that only depend on jobs.

In 2RoGAP, the realization of uncertainty depends on the assignment of the job to the resources. In other words, the second-stage finds the worst case deviations in resource consumption for specific job-resource assignments, and as the assignment changes, the worst case deviations may change. This definition of uncertainty is also used in [Denton et al., 2010] to model the length of the surgery blocks assigned to different operating rooms.

We previously showed that for a given assignment, the problem of finding the worst case deviations for job-resource pairs boils down to finding out which jobs are deviating. Here, we claim that the scenario that only contains the deviations for jobs regardless of the resources they are assigned to is a valid scenario and creates a valid

set of constraints, that are stronger than the original formulation.

Proposition 4.3.4. *Assume z^l is the optimal solution to the second-stage problem (4.9a)-(4.9f) for specific assignment x^l. Constraints (4.15) are valid and stronger for the master problem than (4.12c)*

$$\sum_{j=1}^{n}(\overline{r}_{ij} + z_j^l \hat{r}_{ij})x_{ij} \leq R_i + o_i^k \qquad \forall i. \tag{4.15}$$

Proof. First, we show that these constraints capture the worst-case cost for the assignment x^l that resulted in the deviations. In other words, if instead of (4.9a)-(4.9f), we solved (4.8a)-(4.8g) and obtained optimal z_{ij}^l for each job-resource pair, we have:

$$\sum_{j=1}^{n}(\overline{r}_{ij} + z_j^l \hat{r}_{ij})x_{ij}^l = \sum_{j=1}^{n}(\overline{r}_{ij} + z_{ij}^l \hat{r}_{ij})x_{ij}^l.$$

This is true because $z_{ij}^l = z_j^l x_{ij}^l$. In other words, for the assignment $x_{ij}^l = 1$ the resource consumptions are calculated the same. Note that if job j has a deviations then $z_j^l = 1$ and $z_{ij}^l = 1$. Therefore, the resource consumption for resource i, given that assignment, is correctly calculated.

Next, we need to show that these constraints do not impose overage costs incorrectly. Thus we need to prove that no matter the assignment, the proposed set of deviating jobs is a valid scenario. Note that no matter what the assignment is, all the jobs are assigned to resources and the uncertainty set is restricted by the value of the budget of uncertainty Γ which restricts the sum of normalized deviations in resource consumption for the assigned jobs. Therefore, the scenario $z_j^l, \forall j$ does not violate the budget of uncertainty.

For any other assignment x', the $z_j^l x'_{ij}$ creates a unique job-resource pair deviation

$(z'_{ij} = z^l_j x'_{ij})$ that still satisfies the definition of a valid scenario. If this scenario z'_{ij} is the worst case scenario for x'_{ij}, it imposes the overage costs correctly. In the case this is not the worst case scenario, the imposed cost is less than the worst case. Thus, the second-stage will produce a different scenario z''_j to calculate the worst case overage costs and be added to the master.

The scenario $z_j, \forall j$ is the aggregated version of $z_{ij}, \forall i, j$ since $z_j = \sum_{i=1}^{m} z_{ij}$. Thus it includes the overage costs (not necessarily the worst case costs) for all the assignments where a certain set of jobs are deviating. □

To clarify on the strength of the constraints, consider the case of assigning jobs to resources where $\Gamma = n$. This means that all the jobs will deviate to their worst case resource consumption and $\tilde{r}_{ij} = \bar{r}_{ij} + \hat{r}_{ij}\ \forall i, j$, and the problem becomes deterministic. However, if we choose to solve the problem using job-resource deviation scenarios z_{ij} (which will have n ones and the rest zero), the constraints only consider the overcost for jobs deviating from their nominal resource consumption in a specific assignment (in a given iteration). Therefore, the master problem finds a different assignment where the values for deviation $z_{ij} = 0$ which does not impose overage costs. However, we already know that no matter what the assignments are, the worst-case scenario is when all of the jobs jobs deviate to their maximum resource consumption. Aggregated scenario $z_j, \forall j$ is a vector of ones of size n, and imposes the deviation in all jobs no matter what job-resource assignment is chosen. Therefore, the constraints create stronger lower bounds to the master by combining multiple scenarios into one aggregated scenario that is independent of the assignment.

Similar to the C&CG, we can construct stronger inequalities for the CP method in order to improve the computational performance of Algorithm 2. To do so, at each iteration k we solve the second stage problem with variables z_j. We then replace the

constraints (4.10b) with the following constraints in the master problem:

$$\theta \geq \sum_{i=1}^{m}\sum_{j=1}^{n} \overline{r}_{ij} x_{ij} \pi_i^k + \sum_{i=1}^{m}\sum_{j=1}^{n} \hat{r}_{ij} x_{ij} \pi_i^k z_j^k - \sum_{i=1}^{m} R_i \pi_i^k.$$

In the next section we present another formulation based on [Denton et al., 2010], which presented a formulation for two-stage robust extensible bin-packing with an application to assigning surgery blocks to operating rooms. We follow their steps and adapt their formulation to two-stage robust GAP.

4.3.2 Previous Robust Extensible Bin-Packing Model

The steps taken in this section to derive the deterministic equivalent formulation for the two-stage robust GAP closely follow the steps presented in [Denton et al., 2010], and the model is named DMBH, after the authors (Denton-Miller-Balasubramanian-Huschka) of the paper.

Consider the DGAP (4.1a)-(4.1d) in which the resource consumption coefficient for job-resource pairs belongs to the following uncertainty set

$$\mathcal{U}(\Gamma) = \{\tilde{r} \subset \mathbb{R}^{m \times n} | \tilde{r}_{ij} \in [\overline{r}_{ij}, \overline{r}_{ij} + \hat{r}_{ij}], \sum_{(i,j):x_{ij}=1} \frac{\tilde{r}_{ij} - \overline{r}_{ij}}{\hat{r}_{ij}} \leq \Gamma \qquad \forall i,j\} \tag{4.16}$$

which is similar to the original uncertainty set defined in our previous formulations. The total deviation from the nominal resource consumption for the job-resource pairs is restricted to be less than or equal to the budget Γ. The robust problem can be

formulated as follows:

$$\min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} + R(x, \mathcal{U}(\Gamma)) \tag{4.17a}$$

$s.t.$

$$\sum_{i=1}^{m} x_{ij} = 1 \qquad \forall j \tag{4.17b}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j \tag{4.17c}$$

in which the recourse can be formulated as :

$$R(x, \mathcal{U}(\Gamma)) = \max \sum_{i=1}^{m} b_i \max\{0, \sum_{(i,j):x_{ij}=1} \tilde{r}_{ij} - R_i\} \tag{4.18a}$$

$s.t.$

$$\sum_{(i,j):x_{ij}=1} \frac{\tilde{r}_{ij} - \overline{r}_{ij}}{\hat{r}_{ij}} \leq \Gamma \tag{4.18b}$$

$$\overline{r}_{ij} \leq \tilde{r}_{ij} \leq \overline{r}_{ij} + \hat{r}_{ij}. \qquad \forall i, j \tag{4.18c}$$

As it can be seen in the first stage problem (4.17a)-(4.17c), we aim to minimize the assignment cost and recourse cost with the restriction that each job j should be assigned to a resource i. The recourse problem $R(x, \mathcal{U}(\Gamma))$ finds the worst case overage cost over all resources for a given assignment x and uncertainty budget Γ. The decisions are the values for the resource consumption coefficients which are bounded by their nominal and worst case values. In addition, the total normalized deviation from the nominal job-resource resource consumption cannot exceed the budget of uncertainty Γ.

This formulation is very similar to the previously presented formulations earlier this chapter. However, we aim to reformulate this problem into a single optimization problem that does not require a CP or C&CG type solution methodology and can be solved directly using a solver, as a single optimization problem.

In the objective of the recourse problem, either having or not having overage in a resource can happen. Thus we can reformulate the recourse problem as follows:

$$\max \sum_{i=1}^{m} b_i (\sum_{j=1}^{n} \tilde{r}_{ij} - R_i) z_i \tag{4.19a}$$

$s.t.$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} \frac{\tilde{r}_{ij} - \overline{r}_{ij} x_{ij} z_i}{\hat{r}_{ij}} \leq \Gamma \tag{4.19b}$$

$$\overline{r}_{ij} x_{ij} z_i \leq \tilde{r}_{ij} \leq (\overline{r}_{ij} + \hat{r}_{ij}) x_{ij} z_i \qquad \forall i, j \tag{4.19c}$$

$$z_i \in \{0, 1\}. \qquad \forall i \tag{4.19d}$$

Note that in this formulation variables $z_i \ \forall i$ are different than the previous definition. Note that if there is overage for resource i, then $z_i = 1$ to include the overage cost for that resource in the objective. In addition, from constraints (4.19c), if there is an overage, the resource consumption coefficient will be bounded to its nominal and worst case values, otherwise it will be fixed to zero and will not affect the objective function or budget of uncertainty constraint.

Considering the fact that when $z_i = 1$, $\tilde{r}_{ij} = 0$ holds, we can linearize the objective

and formulate the recourse as the following optimization:

$$\max \sum_{i=1}^{m} \sum_{j=1}^{n} b_i \tilde{r}_{ij} - \sum_{i=1}^{m} b_i R_i z_i \tag{4.20a}$$

$s.t.$

$(4.19b) - (4.19d)$.

Following the same steps presented by [Denton et al., 2010], we make the change of variable

$$\Delta_{ij} = \frac{\tilde{r}_{ij} - \overline{r}_{ij} x_{ij} z_i}{\hat{r}_{ij}}.$$

With the new variable definition, the recourse problem can be reformulated as follows:

$$\max \sum_{i=1}^{m} \sum_{j=1}^{n} b_i \hat{r}_{ij} \Delta_{ij} + \sum_{i=1}^{m} \sum_{j=1}^{n} b_i \overline{r}_{ij} x_{ij} z_i - \sum_{i=1}^{m} b_i R_i z_i \tag{4.21a}$$

$s.t.$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} \Delta_{ij} \leq \Gamma \tag{4.21b}$$

$$0 \leq \Delta_{ij} \leq x_{ij} z_i \qquad \forall i, j \tag{4.21c}$$

$$z_i \in \{0, 1\}. \qquad \forall i \tag{4.21d}$$

Theorem 4.3.5. *Proposition 6 in [Denton et al., 2010]: The polyhedron* $X = \{(\Delta, z) : (4.21b) - (4.21c); 0 \leq z_i \leq 1, \forall i\}$ *has integer extreme points.*

Proof. See the proof of Proposition 6 in [Denton et al., 2010]. □

In light of the theorem 4.3.5, the authors employ strong duality to the linear relaxation of (4.21a)-(4.21d) to reformulate the recourse as a minimization problem and integrate it into the first stage as a single minimization problem as follows:

$$\min \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \Gamma\alpha + \sum_{i=1}^{m}\gamma_i \tag{4.22a}$$

$s.t.$

$$\sum_{i=1}^{m} x_{ij} = 1 \tag{4.22b}$$

$$\alpha + \beta_{ij} \geq b_i\hat{r}_{ij} \qquad \forall i,j \tag{4.22c}$$

$$-\sum_{j=1}^{n} x_{ij}\beta_{ij} + \gamma_i \geq -b_i(R_i - \sum_{j=1}^{n}\overline{r}_{ij}x_{ij}) \qquad \forall i \tag{4.22d}$$

$$x_{ij} \in \{0,1\},\ \alpha, \beta_{ij}, \gamma_i \geq 0. \qquad \forall i,j \tag{4.22e}$$

Note that α is the dual variable corresponding to the uncertainty budget constraint (4.21b). β_{ij} defines the dual variables corresponding to the constraints (4.21c). Finally, γ_i defines the dual variables corresponding to the linear relaxation of binary constraints (4.21d). In addition there is a nonlinear term in the constraints (4.22d).

[Denton et al., 2010] propose a reformulation to replace $x_{ij}\beta_{ij}$ with a new variable

κ_{ij} to obtain the following formulation:

$$\min \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \Gamma\alpha + \sum_{i=1}^{m}\gamma_i \tag{4.23a}$$

$s.t.$

$$\sum_{i=1}^{m} x_{ij} = 1 \tag{4.23b}$$

$$\alpha + \kappa_{ij} \geq b_i \hat{r}_{ij} x_{ij} \qquad \forall i, j \tag{4.23c}$$

$$\sum_{j=1}^{n} \kappa_{ij} \leq b_i(R_i - \sum_{j=1}^{n} \overline{r}_{ij} x_{ij}) + \gamma_i \qquad \forall i \tag{4.23d}$$

$$x_{ij} \in \{0,1\},\ \alpha, \kappa_{ij}, \gamma_i \geq 0 \qquad \forall i, j \tag{4.23e}$$

which produces solutions with the same objective value as the formulation (4.22a)-(4.22e). To see the proof, refer to Proposition 7 in [Denton et al., 2010].

Here we challenge the proposed formulation (DMBH), specifically Proposition 6 in [Denton et al., 2010].

Proposition 4.3.6. *Theorem 4.3.5 is not correct. That is, not all the extreme points of the polyhedron* $X = \{(\Delta, z) : (4.21b) - (4.21c); 0 \leq z_i \leq 1, \forall i\}$ *are integer.*

Proof. We provide a simple example where this polyhedron has a non integer extreme point. Consider the simple case where there are only two jobs ($j = 2$) and one resource ($i = 1$). Therefore, we can drop the index for resource and keep the index for jobs. Assume the assignment where both jobs are assigned to the resource, thus $x_1 = x_2 = 1$. In addition, consider the case where the budget of uncertainty $\Gamma = 1$. The feasible region for the linear relaxation of the second-stage problem (4.21a)-(4.21d) is captured by the following equations after adding the slack variables:

$$
\begin{array}{rrrrrrrr}
\Delta_1 & +\Delta_2 & & +s_1 & & & & = 1 \\
\Delta_1 & & -z & & +s_2 & & & = 0 \\
& \Delta_2 & -z & & & +s_3 & & = 0 \\
& & z & & & & +s_4 & = 1
\end{array}
$$

If we choose the set of basic variables to be the set $\{\Delta_1, \Delta_2, z, s_4\}$ and solve for the system of linear equations, the basic feasible solutions $\{\Delta_1 = 0.5, \Delta_2 = 0.5, z = 0.5, s_4 = 0.5\}$ is obtained which is clearly a non integer extreme point for the feasible region, which can result in over-estimation of the recourse costs and creation of invalid realizations of uncertain parameters. Figure 4.1 represents the feasible region for our example, which clearly confirms the existence of a non integer extreme point.

□

To clarify more on the Proposition 6 in [Denton et al., 2010], we also explain why their proof is not correct. First, we present the proof of the Theorem 4.3.5, as explained in the paper:

"First, observe that $X' = \{(\Delta, z) : (4.21c), 0 \leq z_i \leq 1, \forall i\}$ *is an integral polyhedron, because the constraint matrix is totally unimodular ((4.21c) has exactly one coefficient of* 1 *and one coefficient of* -1 *in each row, and the bounds on* z_i *define an identity matrix). Next, observe that* $X'' = \{(\Delta, z) \in X' : \sum_{(i,j)} \Delta_{ij} = \Gamma\}$ *is a subset of* X *(defined in the Theorem 4.3.5) in which all extreme points are integer. Finally, observe that all extreme points of* X *are either extreme points of* X''*, or extreme points of* X' *in which (4.21b) is satisfied by strict inequality. "*

The main issue comes from the fact that while both X' and $X_\Gamma = \{(\Delta, z) : \sum_{(i,j)} \Delta_{ij} \leq \Gamma, 0 \leq \Delta_{ij} \leq 1\}$ have integer extreme points, the intersection of the two polytope can create other extreme points that do not belong to the extreme points of either sets. In other words, assuming that X'' , the intersection of the hyperplane for the budget of uncertainty with X', has all integer extreme points is incorrect.

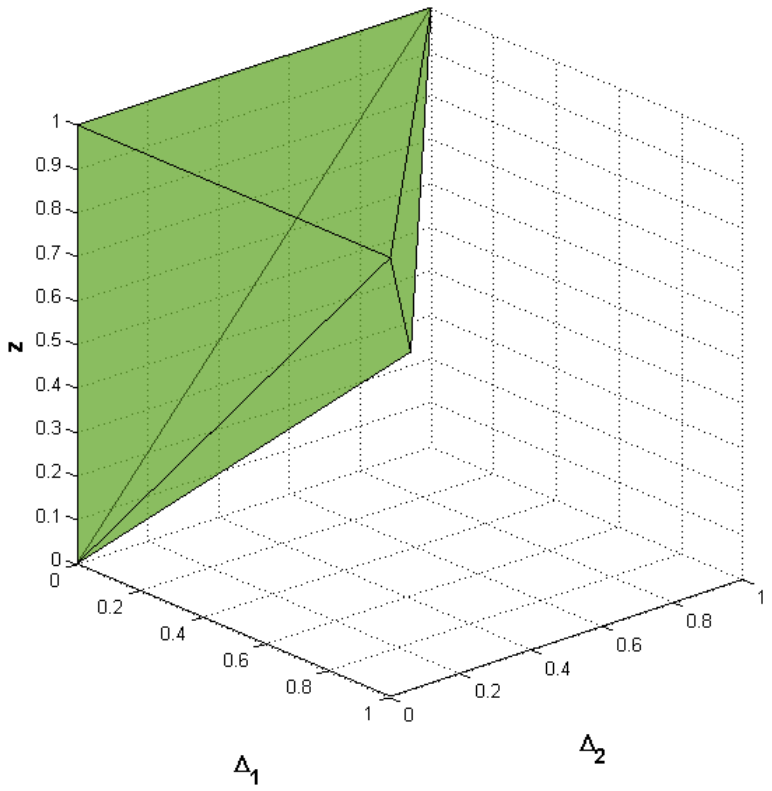


Figure 4.1: Feasible region for problem with two jobs and one resource and $\Gamma = 1$.

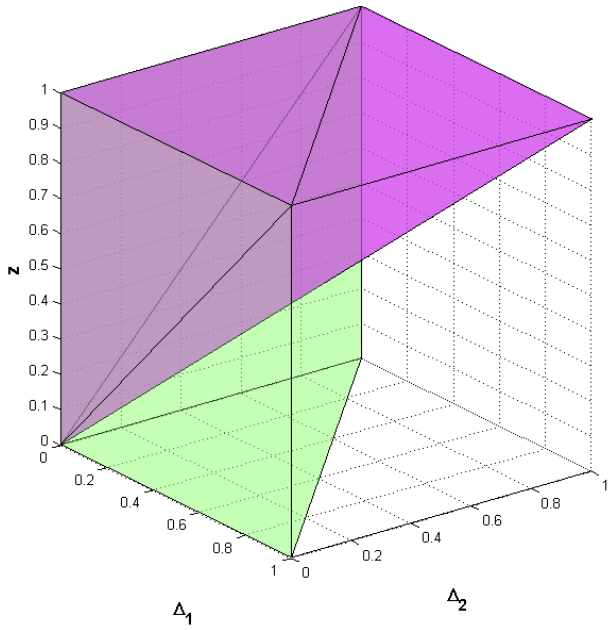


Figure 4.2: Intersection of the sets X' and X_Γ in our example.

Figure 4.2 shows the intersection of the sets X' and X_Γ for our example, and how each set has integer extreme points independently, while their intersection does not. Note that [Ardestani-Jaafari and Delage, 2016] has provided a counter-argument by providing an example, in their report, to point out the error in [Denton et al., 2010]. Our findings are independent of their work.

Implications - As shown in Proposition 4.3.6, the second stage formulation (4.21a)-(4.21d) may not have an integer optimal solution for certain assignments. Thus, its dual will overestimate the overage costs in the objective for those assignments. This can cause some assignments that may be optimal to become infeasible in the formulation (4.23a)-(4.23e). Therefore, there can be instances where the DMBH formulation will provide a solution that is not, in fact, optimal. It is important to note that solutions from the DMBH formulation are, in fact, feasible assignments to the two-stage

Table 4.1: Comparison of results between DMBH and C&CG methods for an instance $m = 5, n = 15$.

Γ	DMBH			C&CG			
	Objective	Recourse	Run Time (s)	Iterations	Objective	Recourse	Run Time (s)
0	240	0	0.016	2	240	0	0.11
1	**281.33**	**12.33**	0.016	5	**258**	**0**	0.423
2	**292.33**	**3.33**	0.031	7	**279**	**0**	0.451
3	294	5	0.031	10	294	5	0.69
4	294	5	0.054	8	294	5	0.633
5	294	5	0.031	6	294	5	0.364
6	294	5	0.016	7	294	5	0.451
7	294	5	0.016	5	294	5	0.306
8	294	5	0.031	4	294	5	0.234
9	294	5	0.016	4	294	5	0.23
10	294	5	0.016	4	294	5	0.232
11	294	5	0.016	4	294	5	0.234
12	294	5	0.031	4	294	5	0.23
13	294	5	0.016	4	294	5	0.228
14	294	5	0.016	3	294	5	0.168
15	294	5	0.016	2	294	5	0.109

robust GAP. Table 4.1 compares the results for a randomly generated instance with $m = 5$ and $n = 15$ that is solved by both DMBH and C&CG method. It can be seen in this instance, for the cases where $\Gamma \in \{1, 2\}$, the objective value and recourse costs from DMBH are greater than those produced by the C&CG method. These are the cases where the DMBH provides sub-optimal assignments and overestimates the recourse costs. On the other hand, it can be seen that DMBH offers superior running time as compared to the iterative C&CG method, and in many cases provides the true optimal solution.

Since the DMBH formulation tends to provide feasible solutions in a non iterative way (unlike C&CG and CP methods), it converges to a *good* feasible solution relatively quickly. We can take advantage of this formulation and the feasible solution provided by the DMBH formulation to establish high quality upper-bounds in our proposed iterative methods.

A Proposed Improved Formulation

As previously discussed, the linear relaxation of the formulation (4.21a)-(4.21d) does not necessarily produce integer solutions, thus can lead to sub-optimal solutions as well as an over-estimation of the recourse costs. On the other hand, the final formulation (4.23a)-(4.23e) offers a better computational performance since, unlike the iterative methods, it does not require solving the master problem for each iteration. It is of great interest to improve the formulation (4.21a)-(4.21d) without increasing its complexity so its relaxation has a smaller integrality gap. This can improve the resulting robust formulation to provide solutions that are closer to the optimal solution.

Based on the structure of the feasible space (4.19b)-(4.21d), we can add a set of constraints that are valid for the integer problem and can strengthen the LP relaxation of the feasible set.

Proposition 4.3.7. *Constraints of type*

$$\sum_{j=1}^{n} \Delta_{ij} \leq \Gamma z_i \qquad \forall i \qquad (4.24a)$$

are valid inequalities for a set defined by the constraints (4.19b)-(4.21d).

Proof. First, we show that inequalities (4.24a) do not remove any feasible integer solutions. Note that if $z_i = 0$ then $\Delta_{ij} = 0 \ \forall j$ due to (4.21c). Therefore, (4.24a) is not affecting that solution. If $z_i = 1$, then (4.24a) is still a valid constraint because it is not more restrictive than (4.21b).

Next, we show that inequalities (4.24a) do remove some fractional solutions, thus improving the linear relaxation of (4.19a)-(4.21d). As previously showen in our counter example, the solution $\Delta_1 = \Delta_2 = z = 0.5$ is a feasible solution to the linear relaxation of (4.19b)-(4.21d) when $\Gamma = 1$. Inequality $\Delta_1 + \Delta_2 \leq z$ does not

hold for this solution. Thus this inequality removes a fractional solution and improves the integrality gap. □

The recourse problem (4.21a)-(4.21d) can be formulated by adding the constraints (4.24a). Let $\nu_i, \forall i$ be the corresponding dual variables for these constraints. Taking the dual of the linear relaxation of this formulation and substituting it with the first stage, and the linearization provides the following formulation:

$$\min \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \Gamma\alpha + \sum_{i=1}^{m}\gamma_i \tag{4.25a}$$

$s.t.$

$$\sum_{i=1}^{m} x_{ij} = 1 \tag{4.25b}$$

$$\alpha + \nu_i + \kappa_{ij} \geq b_i\hat{r}_{ij}x_{ij} \qquad \forall i,j \tag{4.25c}$$

$$\Gamma\nu_i + \sum_{j=1}^{n}\kappa_{ij} \leq b_i(R_i - \sum_{j=1}^{n}\overline{r}_{ij}x_{ij}) + \gamma_i \qquad \forall i \tag{4.25d}$$

$$x_{ij} \in \{0,1\},\ \alpha, \kappa_{ij}, \gamma_i, \nu_i \geq 0. \qquad \forall i,j \tag{4.25e}$$

Note that the linearization of the variables $\beta_{ij}x_{ij} = \kappa_{ij}$ remains the same and holds for the improved formulation as discussed in [Denton et al., 2010]. This formulation offers stronger upperbounds on the true optimal solution of the problem compared to the original DMBH formulation.

It is important to note that solving the same instance addressed in Table 4.1 using our improved version of DMBH yields the optimal solution for all values of Γ. Table 4.2 shows the results for an instance $m = 5$ and $n = 15$ that is solved by the original DMBH formulation, our improved version, and strong C&CG version to compare the quality of the solutions generated. This instance is the same instance as the previous

Table 4.2: Comparing original DMBH formulation with improved DMBH

Γ	DMBH Original				DMBH Improved			
	Objective	Recourse	Time (s)	Gap (%)	Objective	Recourse	Time (s)	Gap (%)
0	441	85	0.034	0	441	85	0.021	0
1	635	295	0.085	19.361	532	176	0.028	0
2	701	361	0.154	2.187	686	309	0.096	0
3	750.667	344.667	0.152	0.223	750.667	344.667	0.183	0.223
4	788	382	0.239	0.254	788	382	0.36	0.254
5	822	416	0.413	0	822	416	0.537	0
6	840	449	0.534	0	840	449	0.496	0
7	854	463	0.486	0	854	463	0.48	0
8	868	477	0.669	0	868	477	0.517	0
9	872	494	0.333	0	872	494	0.342	0
10	872	494	0.359	0	872	494	0.357	0
11	872	494	0.218	0	872	494	0.312	0
12	872	494	0.156	0	872	494	0.156	0
13	872	494	0.156	0	872	494	0.144	0
14	872	494	0.133	0	872	494	0.125	0
15	872	494	0.097	0	872	494	0.071	0

one with the only difference of lowering the capacity for resources.

It can be seen in the Table 4.2 that the original DMBH formulation fails to find the optimal solution for $\Gamma = 1, 2, 3, 4$ with the largest gap of over 19% form the optimal solution (obtained by C&CG) for the case of $\Gamma = 1$. Improved DMBH does not find the optimal solution for $\Gamma = 3, 4$ with gaps less than 0.5% and performs similar to DMBH. However, for $\Gamma = 1, 2$, it yields the optimal solution while original DMBH failed.

We make no claims about the improved formulation to give exact optimal solution, but we believe it performs at least as good as the original DMBH formulation with the possibility of great improvement in solution quality.

4.3.3 Computational Results

Instance Generation

In order to test our proposed algorithms, random instances are generated. For a given number of resources m, and jobs n, the following data are generated:

- **Cost:** The assignment cost of a resource i to job j is generated from an integer uniform distribution $Unif(1, 100)$.

- **Nominal resource consumption:** The nominal value for resource consumption for each resource-job pair is chosen randomly from $Unif(50, 100)$ distribution.

- **Deviation in resource consumption:** The worst-case deviation from the nominal resource consumption for each resource-job pair is randomly selected from a $Unif(10, 40)$ distribution.

- **Overage cost:** The cost of ogoing over capacity for each resource is randomly selected from a $Unif(1, 20)$ distribution.

- **Resource capacity:** The capacity for each resource is calculated as a function of $\overline{d}$ and $\hat{d}$ and the number of jobs and machines such that for large values of Γ the chances of being able to avoid going over the capacity is very small. Here, is the general form to calculate the capacity for each resource:

$$R_i = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} (\overline{d}_{ij} + \hat{d}_{ij})}{f(m, n)} \qquad \forall i$$

in which there are multiple choices for the function f and two of the forms that were used are $f(m, n) = cmn$ and $f(m) = cm^2$, where c is a constant. The

capacity of all resources are equal to each other. This is to create cases where increasing the value of Γ at some point will cause going over capacity.

As discussed before, we presented solution methods based on a cutting-plane (CP) algorithm and the column-and-constraint generation (C&CG) algorithm. For each method, we proposed a variant that employs stronger cuts by defining the uncertainty as a resource-independent deviation in the resource requirement coefficient. All the tests are run on a Windows machine with an Intel Core i5, 3.20GHz CPU, and 4 GB of RAM.

To show how the CP method and weak and strong versions of the C&CG compare, we solved a randomly generated problem with $m = 5$ and $n = 10$ for different values of $\Gamma = 0, .., n$ using all three methods. Figure 4.4 shows how each method compares in terms of the number of iterations required to reach to the optimal solution for different values of Γ. While both methods converge to the same optimal solution for each value of Γ, the C&CG method that employs stronger constraints has a significant advantage in convergence over the weak version. Figure 4.3 shows the comparison between the run times for methods to reach optimality for each value of Γ. It can be seen that methods that employ the strong constraints have significant advantage over the variants with the weaker constraints. Figure 4.5 compares how the optimality gap for the four methods decreases as the number of iterations increases for the case where $\Gamma = 6$.

Our tests show similar trends for different problem sizes. It is important to note that there is a trade-off between using CP and C&CG method. While C&CG method tends to improve the convergence by adding multiple constraints at each iteration, the size of the master problem grows faster compared to the CP method. This can increase the time required to perform each iteration. To test this, we generated a random instance with $m = 5$ and $n = 15$ and solved it using the strong versions of CP and C&CG. For the case when $\Gamma = 5$, CP takes 16.37 seconds and 42 iterations, while

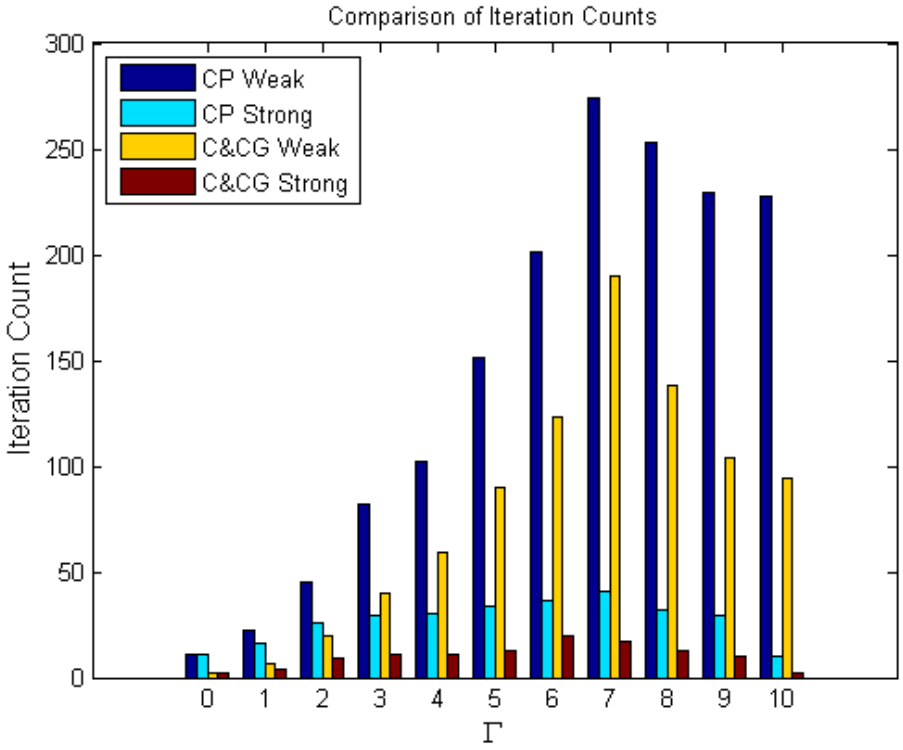


Figure 4.3: Iteration comparison between the weak and strong versions of CP and C&CG methods (Iteration count is the value of the counter k in the algorithm).

C&CG solved the same instance in6.78 seconds and 21 iterations. Figure 4.6 shows the reduction of optimality gap with respect to the running time of the algorithms. It shows C&CG has superior convergence than CP in this instance.

Since our results show that C&CG method has a better performance, we conduct the rest of the computational test using the strong version of the C&CG method.

It is important to note that after solving the robust GAP for a specific value of Γ, the optimization finds the best allocation of jobs to resources to minimize the assignment and overage costs for the worst-case deviation of at most Γ jobs from their nominal resource requirement. In reality, the resource requirements are uncertain and can have any values in their respective ranges. To analyze the performance of the robust solutions, a simulation study is performed to measure the average overage

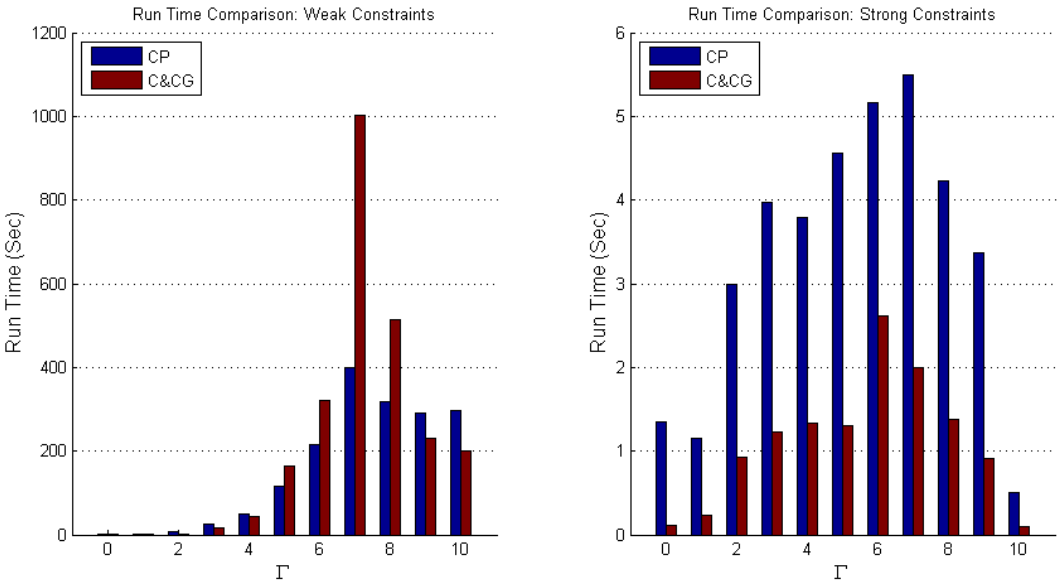


Figure 4.4: Run time comparison between the weak and strong versions of CP and C&CG methods (y-axes are not the same scale).

costs incurred for a given assignment. Additionally, the average probability of going over the capacity in at least one of the resources is calculated.

To perform the simulation, for a given assignment, the resource requirements for the job-resource pairs $(i, j), x_{ij} = 1$ are chosen from the $Unif(\overline{d}_{ij}, \overline{d}_{ij} + \hat{d}_i j)$ distribution. Note that the simulation does not consider the value of the uncertainty budget in generating the random numbers and all the jobs have deviations in their resource requirements from their nominal values.

We created random instances using the scheme discussed above. We estimated the resource capacity using the function $f(m) = 1.1m^2$. Ten random instances of different sizes $(m, n) \in \{(5, 10), (5, 20), (10, 20), (10, 30)\}$ are generated and solved using he strong version of the C&CG method. The values for the budget of uncertainty are are selected with 10% increments and each instance is solved for different values of Γ. To clarify, $\Gamma = 10\%$ for a problem with $m = 5$ and $n = 20$ means that at most 10%

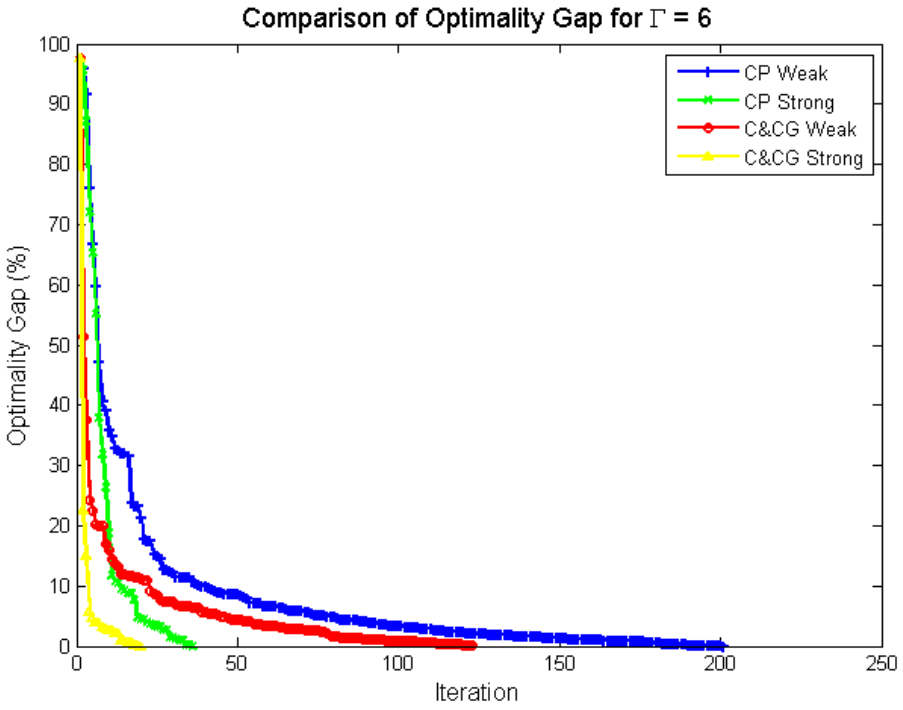


Figure 4.5: Optimality gap comparison between the weak and strong versions of CP and C&CG methods for $\Gamma = 6$.

of the jobs can deviate from their nominal values, which translates into at most two jobs for this specific problem size. To limit the running time for large instances, we limit the algorithm running time to 500 seconds.

After solving each instance for a specific value of Γ, our simulation code generates 5000 replications for the resource requirements for the best found assignment. Values for average overage cost and probability of going over the capacity are calculated. The results are then averaged among the 10 instance for each value of Γ. Table 4.3 includes the results obtained from our tests. Each cell of the table shows a pair of numeric values. The first is the expected value and the second number represents the standard deviation of the metrics that are measured.

From the Table 4.3, it can be seen that the time required to solve an instance depends on the size of the problem and the value of the budget of uncertainty Γ. In

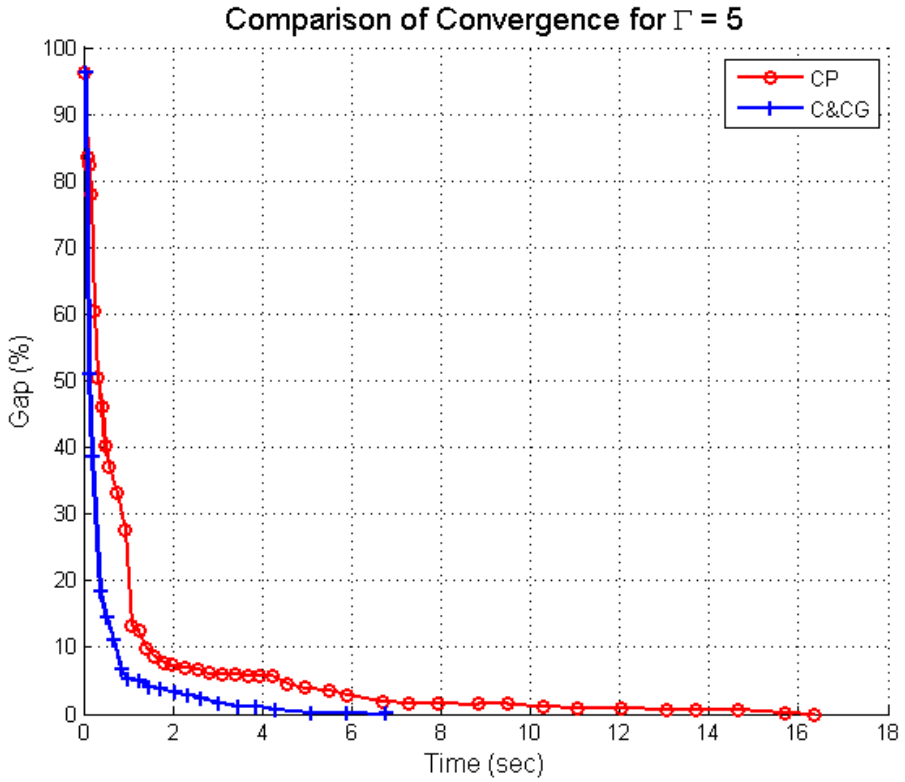


Figure 4.6: Convergence comparison between the strong versions of CP and C&CG methods for $\Gamma = 5$.

general, when $\Gamma = 0\%$ or $\Gamma = 100\%$, the problem becomes a deterministic one with one known scenario. For different values of $\Gamma(\%)$, the total number of scenarios is bounded by $\binom{n}{\Gamma n}$. This can be seen from the general pattern in the number of scenarios required to solve different instances. The objective function and the recourse which is the worst-case overage costs are also presented. The column named "over cost" shows the results of the simulation which calculates the average overage cost when jobs randomly deviate from their nominal values. It can be seen that as Γ increases, the objective function increases since the worst-case realizations become larger in impact. However, the average overage cost and probability of going over the capacity (Over Prob.) decreases as Γ increases. In other words, as we choose to be more

conservative, the assignment will change to accommodate the risks of jobs deviating from their nominal value.

It is important to note that while the objective function is non-decreasing in Γ, no such claim can be made about the assignment costs or recourse costs. Furthermore, we cannot conclude that average overage cost or overage probability are decreasing in Γ. This is discussed by an example.

To better understand how Γ impacts the assignments, a random instance with $m = 5$ and $n = 20$ was generated and solved for all the values of $\Gamma \in \{0, .., n\}$. Table 4.4 shows the number of iterations, run time, optimal objective value, and recourse costs for each value of Γ. It can be seen that for $\Gamma \geq 8$ the objective value stays the same. This means that the optimal assignment for worst-case deviations for these values of the uncertainty budget is the same. In this case, it can be seen that the worst-case recourse (overage) costs are not monotonic.

To understand how the assignments impact the performance in the case of random realizations of the resource requirements, we performed our simulation analysis to measure the average utilization rate for each resource for a given value of Γ. Additionally, we measured the average overage costs for each resource as well to understand and highlight how loads are assigned to each resource. To measure the impact of the uncertainty, two different distributions are used. First we use a uniform distribution same as discussed above. Next, we generate the resource requirements using a triangular distribution, $Tri(\overline{d}_{ij}, \overline{d}_{ij}, \overline{d}_{ij} + \hat{d}_i j)$, with its peak at the nominal value.

Figure 4.7 compares the average utilization rate among the five resources for different values of Γ. It can be seen that for $\Gamma = 0$, where the decision maker does not consider any risk for deviations, the assignments are not balanced and resources one, two and five, on average, are over-utilized. Including some level of risk $\Gamma = 1, 2, 3$ redistributes the load among resources such that the average utilization is less volatile. As the value for Γ goes to eight, there is no way to avoid overage costs. In other

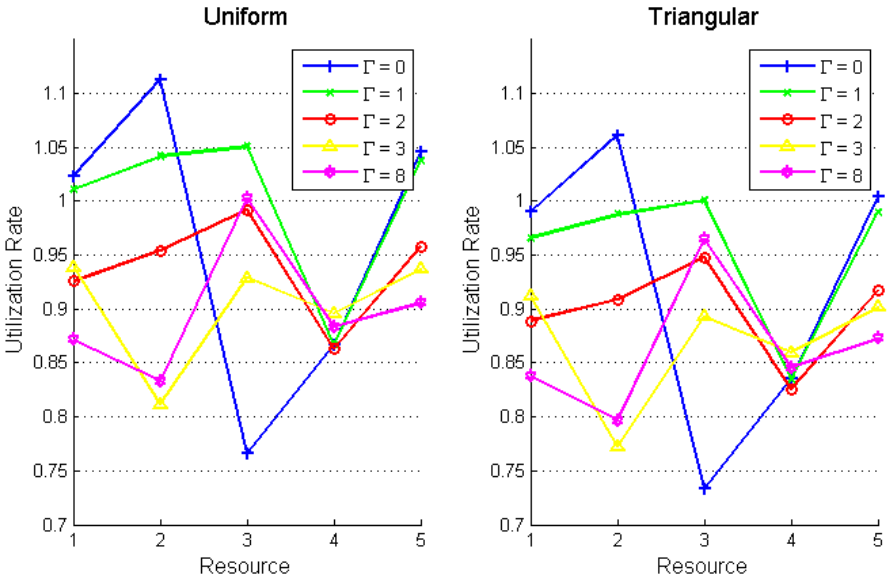


Figure 4.7: Utilization rate for each resource for different values of Γ for different distributions.

words, for $\Gamma \geq 8$, worst case realizations will always have overages. In this case the assignment is changed such that overages are assigned to the resource with the least overage cost (resource three) while maintaining a low assignment cost. In the case of the triangular distribution, the general patterns are similar to the uniform distribution. However, the average rates are smaller compared to the case of uniform distribution.

Figure 4.8, shows the distribution of average overage costs among different resources for different values of Γ. In the case where $\Gamma = 0$, the optimistic assignment aims to minimize the assignment cost without considering any deviation. The jobs are assigned to resources with low assignment cost. However, the results from the simulation shows that there will be a high overage cost for overloaded resources. As we increase the value of the uncertainty budget, since the optimization is concerned with the worst-case overage costs, the average overage cost for each resource drops

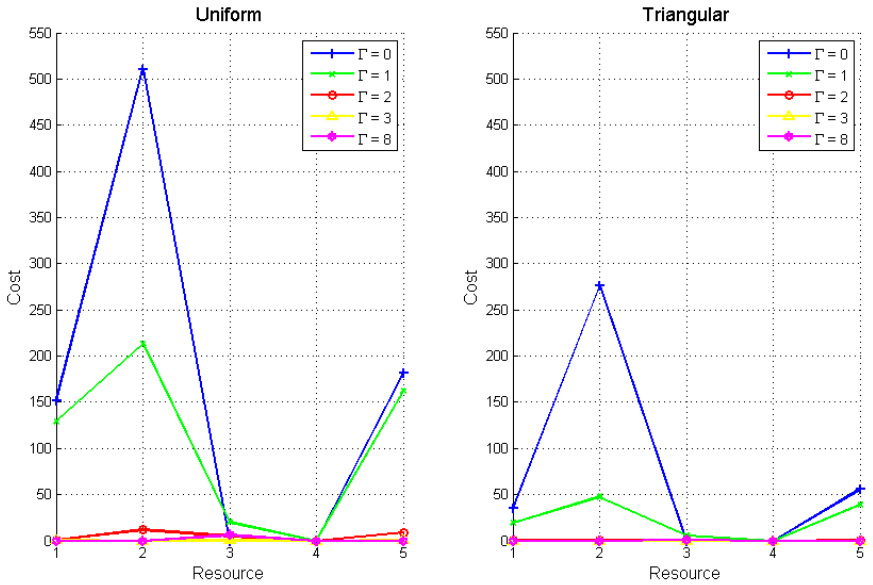


Figure 4.8: Average overage cost for each resource for different values of Γ for different distributions.

significantly. For large values of Γ in which overage costs are unavoidable, the assignment is performed such that resource with lower overage costs incur the cost. The triangular distribution creates the same patterns of behavior with smaller values.

Finally, we studied the impact of distributions on the probability of having an overage. It can be seen from Figure 4.9 that if no risk is taken into account, $\Gamma = 0$, with certainty there will be a lack of capacity. As we increase the value of Γ to six, the probability of having overages drops to zero. As we increase the uncertainty budget even more, which means that we are very pessimistic about the resource requirements for jobs and expect more that six jobs to have their largest resource requirements, the assignment is changed such that the probability of having overages are increased but the cost of having overages are very small. It can be seen that triangular distribution, with its peak at the nominal resource requirement value, estimates a lower probability of overages. This is intuitive since it is more likely for jobs to have resource requirements closer to their nominal values.

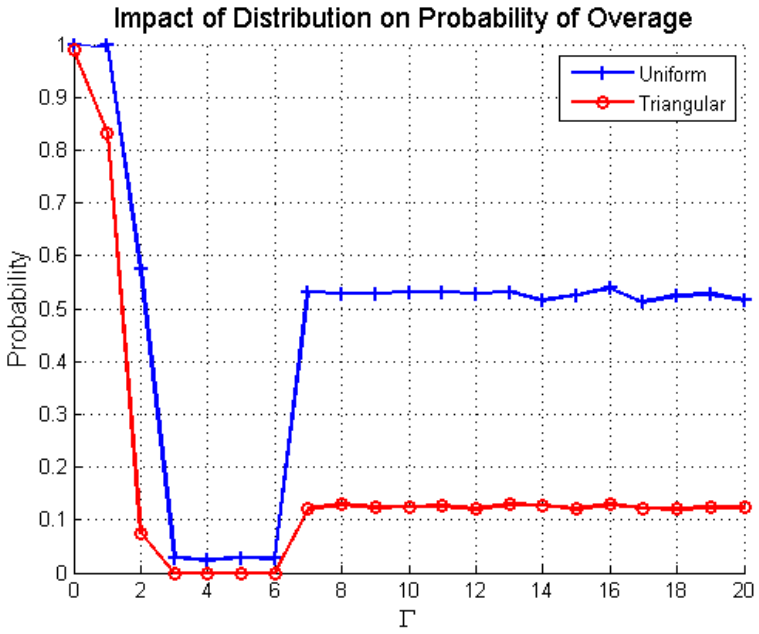


Figure 4.9: Impact of distribution on the probability of having overages for different values of Γ.

Figure 4.10 shows how the average total overage costs behave as the value of the budget of uncertainty is increased. It can be seen that as we increase the value of Γ to be more than zero, a significant drop in average overage costs happens.

4.4 Conclusion

In this chapter, we presented the formulation and exact solution approach for the robust generalized assignment problem. We also studied the structural properties of this problem and proposed a alternative view of uncertainty which results in better performance for both CP and C&CG method by strengthening the generated constraints. We also adapted our formulation based on the DMBH formulation and showed that their formulation does not necessarily provide the optimal solution. Our extensive computational tests show the performance of our solution methodology and

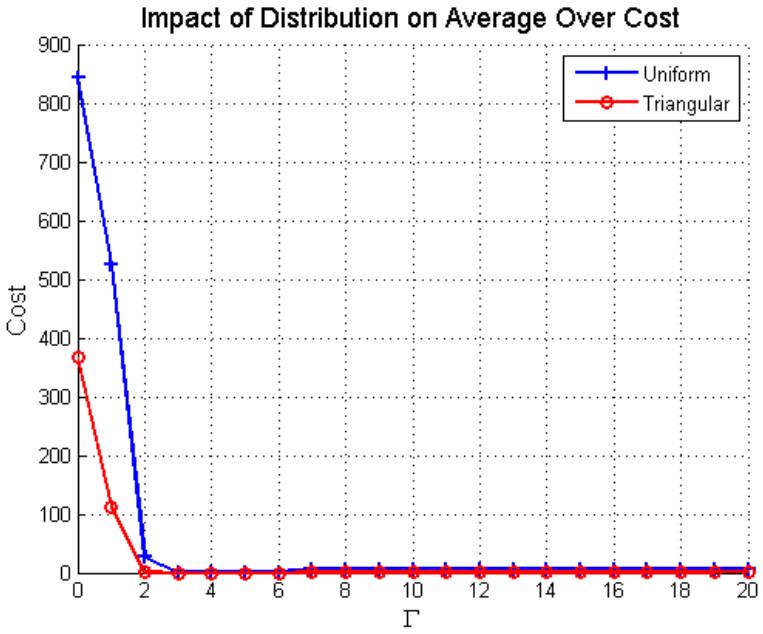


Figure 4.10: Impact of distribution on the average total overage costs for different values of Γ.

our simulation study gives us a view on how the robust assignments would perform.

Table 4.3: Results for 10 instances of different sizes using the strong C&CG (Average, Standard deviation)

Size	Γ (%)	Iteration	Time (s)	Objective	Recourse	Gap (%)	Over Cost	Over Prob.
$m = 5$ $n = 10$	0	(2, 0)	(0.12, 0.03)	(194.2, 50.11)	(0, 0)	(0, 0)	(346.84, 334.41)	(0.84, 0.31)
	10	(5.2, 2.59)	(0.62, 0.4)	(243.4, 74.11)	(1.6, 2.61)	(0, 0)	(12.16, 14.44)	(0.33, 0.17)
	20	(9, 5.96)	(2.24, 2.65)	(306, 98.7)	(37.4, 29.78)	(0, 0)	(2.37, 1.99)	(0.17, 0.12)
	30	(6.6, 1.67)	(1.04, 0.75)	(306, 98.7)	(37.4, 29.78)	(0, 0)	(2.41, 2.02)	(0.17, 0.12)
	40	(5.4, 2.07)	(0.78, 0.42)	(320.2, 93.16)	(24.2, 30.44)	(0, 0)	(1.09, 2)	(0.1, 0.13)
	50	(4.8, 2.05)	(0.7, 0.59)	(320.2, 93.16)	(24.2, 30.44)	(0, 0)	(1.08, 1.96)	(0.1, 0.13)
	60	(4, 1)	(0.5, 0.16)	(320.2, 93.16)	(24.2, 30.44)	(0, 0)	(1.05, 1.87)	(0.1, 0.13)
	70	(3.8, 0.84)	(0.43, 0.15)	(320.2, 93.16)	(24.2, 30.44)	(0, 0)	(1.05, 1.89)	(0.1, 0.13)
	80	(3.6, 0.89)	(0.35, 0.25)	(320.2, 93.16)	(24.2, 30.44)	(0, 0)	(1.11, 2.01)	(0.1, 0.14)
	90	(3.2, 1.1)	(0.32, 0.21)	(320.2, 93.16)	(24.2, 30.44)	(0, 0)	(1.1, 1.98)	(0.1, 0.13)
	100	(2, 0)	(0.11, 0)	(320.2, 93.16)	(24.2, 30.44)	(0, 0)	(1.12, 1.97)	(0.1, 0.13)
$m = 5$ $n = 20$	0	(2, 0)	(0.17, 0.07)	(384.6, 56.27)	(1.6, 3.58)	(0, 0)	(966.76, 523.68)	(0.98, 0.04)
	10	(14.6, 3.91)	(5.47, 2.83)	(445.4, 52.42)	(1.4, 3.13)	(0, 0)	(53.29, 53.97)	(0.41, 0.18)
	20	(72.8, 11.58)	(436.48, 133.11)	(589.4, 38.62)	(75, 36.08)	(4.17, 3.96)	(9.23, 16.17)	(0.3, 0.43)
	30	(48.4, 11.08)	(280.53, 210.23)	(589.6, 44.61)	(67.6, 57.17)	(1.3, 2.26)	(15.12, 29.17)	(0.31, 0.45)
	40	(27.6, 10.92)	(171.91, 188.24)	(595.4, 39.03)	(15.8, 26.44)	(0, 0)	(1.74, 3.89)	(0.11, 0.25)
	50	(18, 5.34)	(61.29, 75.24)	(595.4, 39.03)	(15.8, 26.44)	(0, 0)	(1.75, 3.91)	(0.11, 0.25)
	60	(10.6, 2.51)	(14.53, 14.31)	(595.4, 39.03)	(15.8, 26.44)	(0, 0)	(1.73, 3.86)	(0.11, 0.25)
	70	(9.4, 2.61)	(8.86, 7.58)	(595.4, 39.03)	(15.8, 26.44)	(0, 0)	(1.75, 3.92)	(0.11, 0.25)
	80	(8.4, 2.3)	(7.04, 8.62)	(595.4, 39.03)	(15.8, 26.44)	(0, 0)	(1.71, 3.83)	(0.11, 0.25)
	90	(8.4, 2.3)	(7.11, 8.36)	(595.4, 39.03)	(15.8, 26.44)	(0, 0)	(1.68, 3.75)	(0.11, 0.24)
	100	(2, 0)	(0.24, 0.09)	(595.4, 39.03)	(15.8, 26.44)	(0, 0)	(1.7, 3.79)	(0.11, 0.24)
$m = 10$ $n = 20$	0	(2, 0)	(0.21, 0.04)	(223.4, 33.99)	(0, 0)	(0, 0)	(1049.94, 182.61)	(1, 0)
	10	(49.2, 4.02)	(331.58, 162.74)	(445, 54.72)	(58.6, 38.69)	(0.68, 1.53)	(22.32, 17.64)	(0.69, 0.41)
	20	(29.8, 8.87)	(288.7, 240.38)	(473.8, 67.72)	(58.8, 49.13)	(0.62, 1.04)	(23.15, 23.82)	(0.62, 0.51)
	30	(17.2, 4.71)	(75.85, 89.3)	(474.8, 65.61)	(14.8, 33.09)	(0, 0)	(5.65, 12.64)	(0.2, 0.44)
	40	(11, 1.41)	(13.39, 10.93)	(474.8, 65.61)	(14.8, 33.09)	(0, 0)	(5.75, 12.87)	(0.2, 0.44)
	50	(8.4, 1.14)	(3.64, 1.81)	(474.8, 65.61)	(14.8, 33.09)	(0, 0)	(5.7, 12.74)	(0.2, 0.44)
	60	(7, 1)	(2.16, 1.04)	(474.8, 65.61)	(14.8, 33.09)	(0, 0)	(5.74, 12.84)	(0.2, 0.44)
	70	(6.8, 1.64)	(2.12, 0.86)	(474.8, 65.61)	(14.8, 33.09)	(0, 0)	(5.77, 12.9)	(0.2, 0.44)
	80	(6.8, 1.1)	(1.99, 0.95)	(474.8, 65.61)	(14.8, 33.09)	(0, 0)	(5.74, 12.84)	(0.2, 0.44)
	90	(6.2, 1.1)	(1.51, 0.69)	(474.8, 65.61)	(14.8, 33.09)	(0, 0)	(5.68, 12.69)	(0.2, 0.44)
	100	(2, 0)	(0.28, 0.08)	(474.8, 65.61)	(14.8, 33.09)	(0, 0)	(5.66, 12.66)	(0.2, 0.44)
$m = 10$ $n = 30$	0	(2, 0)	(0.2, 0.04)	(326.6, 35.25)	(0.6, 1.34)	(0, 0)	(1545.55, 952.62)	(1, 0)
	10	(55.4, 7.44)	(333.54, 228.29)	(508.2, 98.02)	(52.2, 34.68)	(5.7, 6.22)	(4.96, 4.11)	(0.33, 0.32)
	20	(34.8, 7.79)	(348.91, 214.14)	(561.6, 170.45)	(81.6, 110.91)	(7.95, 13.66)	(3.32, 4.36)	(0.2, 0.25)
	30	(22.6, 4.72)	(246.82, 238.38)	(571, 169.68)	(75.6, 101.42)	(7.24, 11.84)	(1.12, 2.09)	(0.07, 0.09)
	40	(16.2, 5.45)	(212.75, 262.56)	(583.4, 194.24)	(84.8, 114.25)	(6.86, 12.54)	(2.19, 3.04)	(0.14, 0.22)
	50	(15.8, 7.66)	(206.19, 268.3)	(544.4, 114.38)	(37.8, 33.47)	(1.32, 1.83)	(0.07, 0.07)	(0.01, 0.02)
	60	(12.4, 5.13)	(79.67, 98.53)	(537.2, 104.95)	(28.4, 27.99)	(0, 0)	(1.37, 2.96)	(0.11, 0.22)
	70	(10.4, 3.05)	(46.95, 84.73)	(537.2, 104.95)	(28.4, 27.99)	(0, 0)	(1.38, 2.98)	(0.11, 0.23)
	80	(10.6, 6.35)	(43.97, 71.56)	(537.2, 104.95)	(28.4, 27.99)	(0, 0)	(1.29, 2.78)	(0.11, 0.22)
	90	(10.8, 6.61)	(31.05, 43.16)	(537.2, 104.95)	(28.4, 27.99)	(0, 0)	(1.38, 2.99)	(0.11, 0.22)
	100	(2, 0)	(0.44, 0.25)	(537.2, 104.95)	(26, 29.67)	(0, 0)	(1.34, 2.9)	(0.11, 0.22)

Table 4.4: Results from the instance with $m = 5$ and $n = 20$

Γ	Iteration	Time (s)	Objective	Recourse
0	2	0.12	311	0
1	6	0.38	343	0
2	24	5.23	398	7
3	74	115.02	464	0
4	94	501.5	507	24
5	66	178.39	507	24
6	40	58.81	507	24
7	30	30.39	511	44
8	29	29.56	521	54
9	23	20.33	521	54
10	17	10.51	521	54
11	12	5.06	521	54
12	9	1.83	521	54
13	10	2.63	521	54
14	9	1.52	521	54
15	9	1.39	521	54
16	9	1.39	521	54
17	9	1.4	521	54
18	9	1.48	521	54
19	9	1.4	521	54
20	2	0.12	521	54

Chapter 5: Discussion and Contributions

Uncertainty is an unavoidable element in our decision-making process. Availability of data and advances in optimization software enable us to use stochastic programming (SP) and robust optimization (RO) theories to include uncertainty in our decision-making process. Through these tools we are able to include such considerations into our models and enable decision-makers to better understand the consequences of such uncertain events.

In this dissertation we focused on studying decision-making under uncertainty, in particular two-stage robust optimization, and finding solution approaches for such problems.

5.1 Integrated Surgery Scheduling

In Section 3, we studied a problem in which we consider the impact of uncertainty in the length-of-stay for patients in an ICU, thereby reducing the potential for surgery schedule disruptions and consequent harm to patients requiring intensive care after surgery. This problem has a unique structure in which the uncertainty in the length-of-stay (LOS) belongs to a discrete set. This structure makes the planning very complex due to making the estimation of downstream resource requirements in the future very difficult. Such structures have not been addressed in theoretical developments of robust optimization. The work done in Section 3 is based on [Neyshabouri and Berg, 2016].

Our definition of uncertainty carefully models the patient's movements from the operating rooms to the downstream units and allows us to keep track of the required

downstream resources. In addition, we provide a methodology that weights the costs of not having ICU beds available against the efficiency of using the capacity of the surgical suits completely. The methodology takes into consideration the stochastic nature of time required for surgical procedures as well as the variability of the length-of-stay in ICU facilities. Unlike stochastic optimization, our method does not require distributional information for each type of surgical procedure. Instead, a robust optimization that requires a nominal and a maximum parameter for surgical duration and length-of-stay for each patient is proposed. This information can be obtained using data-driven approaches, as well as from subject-matter experts.

The robust formulation inherently includes risk into the decision process by optimizing against the worst-case realization of the uncertain parameters. To avoid over-conservative solutions, a budget of uncertainty is defined to control and model the number of uncertain parameters that take on their worst-case values. Thus, decision-makers are given the opportunity to control the decisions based on their attitudes towards risk.

Our robust formulation for surgery scheduling consists of a master problem and two sub-problems. The master problem serves as the schedule generator and produces surgery schedules. Each sub-problem serves as the evaluation of the impact of the generated schedule under uncertainty. Sub-problems provide information in the way of columns and constraints back to the master problem which is then re-solved to create a new schedule. The process continues until convergence is achieved.

We show that the special structure of the uncertainty in LOS and its dependence on the first-stage decisions is new to the literature, and causes proposed solution methodologies in the literature not to be directly applicable in solving this problem. We adapt a column-and-constraint generation approach to address this issue.

Computational evaluation of the method showed that as the instances get larger, proving optimality may not be possible in a reasonable amount of time. However,

the methodology does provide provably *near optimal* solutions within a reasonable amount of time to problems of medium size. In addition, taking the uncertainty in downstream units into account can reduce the congestion in the operating rooms. Simulation studies done as part of this research show the quality of our robust optimization technology in providing the decision-maker with a deeper understanding of the consequences of the inherent uncertainty. Operational and risk metrics are calculated which along with costs can provide the decision-maker with alternative surgery schedules and their likely performance.

Our methodology opens the door for other applications where scheduling ahead of time with resource considerations are important.

5.2 Generalized Assignment Problem

In Section 4 we studied the very important problem of the generalized assignment problem (GAP) when resources have a stochastic component, i.e., the capacity requirements are not known with certainty.

In the deterministic version of GAP, the aim is to find the best allocation of jobs to machines with limited capacity such that the assignment cost is minimized and no machine capacity restrictions are violated. In our case, we study GAP under uncertainty in the job-machine resource requirements. Since the amount of capacity that is required by a job on a machine is not exactly known, there is an inherent risk in not having sufficient capacity.

We formulate this problem as a two-stage robust optimization problem in which each job-machine resource requirement belongs to a range. The formulation allows us to better understand the trade-offs between assignment costs and overage costs due to uncertainty. We investigate the literature and propose two different solution algorithms: one is based on Bender's decomposition and the other is a column-and-constraint (C&CG) generation method. In both methods, the problem is consisted of

a master problem and one sub-problem. The master generates assignments, while the sub-problem evaluates the impact of uncertainty on that assignment and provides the master with information in the form of cuts (in Bender's) or columns and constraints (in C&CG) to revise the assignment, then the master is re-solved. The process continues until convergence to the optimal solution is achieved.

A careful study of the structural properties of this problem enables us to improve each of the solution methods by proposing stronger cuts to the master. Intuitively, these stronger cuts are grouping multiple realizations of uncertainty in one single scenario which allows for faster improvement of the lower bounds. Our computational results show these cuts offer substantial improvement in the performance of the proposed solution algorithms. In general, we found that the column-and-constraint generation method does better than the cutting-plane methods.

In our study of the literature, we found only one formulation that claims to be able to formulate a two-stage robust bin-packing problem as a compact optimization formulation that does not require iterative methods such as C&CG. We are able to provide a counter-example to show that the proposed formulation may not provide optimal solutions, but may over-estimates the total costs. We propose valid inequalities to strengthen their formulation and provide numerical experiments to show the effectiveness of those inequalities.

Our computational tests show that the proposed exact solution procedure can solve medium-sized instances in a reasonable amount of time. Our simulation study sheds light on the implications of robustness and uncertainty on the assignment. Through simulation we show that when uncertainty is considered, the optimal assignment tends to reduce the risk of requiring extra capacity, where possible, by dispersing the load over multiple machines, thereby working to load-balance the machines. As the decision-maker's optimism decreases, or the budget of uncertainty increases, such that overages are unavoidable, the assignment changes to assign risky

jobs to the machine that has the lower overage cost.

5.3 Future Directions

In this research we focused on exact solution methods. Due to the complexity of the problems we studied, these exact methods may not be able to always find the optimal solution, although they find good solutions in reasonable time. Our studies showed that convergence to the optimal solution can be slow and is directly related to the quality of the lower bounds generated by our algorithms. One possible avenue for future research is to accelerate these solution methods by developing tighter cuts and feeding feasible solutions to the master when possible.

We hoped that using the objective function for different values of the budget of uncertainty would improve the convergence of the methods. However, our preliminary tests showed that using the objective values for a certain uncertainty budget as a lower or upper bound for another does not improve the solution time. Thus, better bounding techniques are required.

Another avenue for research is to use the structural insights from the formulations to develop high quality heuristics to obtain high-quality solutions faster.

It is also possible to extend the application of surgery scheduling by accurately costing various outcomes, inclusion if pre-op, and combining appointment scheduling and consultation with surgery scheduling.

Finally, one can use this research as a starting step for other applications where uncertainty occurs as discrete events. Most of the literature in RO does not address this issue, while the number of instances where the events that are uncertain can take on only a finite and small number of distinct values, are many.

Bibliography

Bibliography

[Addis et al., 2014] Addis, B., Carello, G., and Tànfani, E. (2014). A robust optimization approach for the operating room planning problem with uncertain surgery duration. In *Proceedings of the International Conference on Health Care Systems Engineering*, pages 175–189. Springer.

[Albareda-Sambola and Fernández, 2000] Albareda-Sambola, M. and Fernández, E. (2000). The stochastic generalised assignment problem with bernoulli demands. *Top*, 8(2):165–190.

[Albareda-Sambola et al., 2006] Albareda-Sambola, M., Van Der Vlerk, M. H., and Fernández, E. (2006). Exact solutions to a class of stochastic generalized assignment problems. *European journal of operational research*, 173(2):465–487.

[Ardestani-Jaafari and Delage, 2016] Ardestani-Jaafari, A. and Delage, E. (2016). Linearized robust counterparts of two-stage robust optimization problem with applications in operations management. *Research Report*.

[Argo et al., 2009] Argo, J. L., Vick, C. C., Graham, L. A., Itani, K. M., Bishop, M. J., and Hawn, M. T. (2009). Elective surgical case cancellation in the veterans health administration system: identifying areas for improvement. *The American Journal of Surgery*, 198(5):600–606.

[Atamtürk and Zhang, 2007] Atamtürk, A. and Zhang, M. (2007). Two-stage robust network flow and design under demand uncertainty. *Operations Research*, 55(4):662–673.

[Bam et al., 2015] Bam, M., Denton, B. T., Van Oyen, M. P., and Cowen, M. (2015). Surgery scheduling with recovery resources.

[Ben-Tal et al., 2009] Ben-Tal, A., El Ghaoui, L., and Nemirovski, A. (2009). *Robust optimization*. Princeton University Press.

[Ben-Tal et al., 2004] Ben-Tal, A., Goryashko, A., Guslitzer, E., and Nemirovski, A. (2004). Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99(2):351–376.

[Ben-Tal and Nemirovski, 1998] Ben-Tal, A. and Nemirovski, A. (1998). Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805.

[Ben-Tal and Nemirovski, 1999] Ben-Tal, A. and Nemirovski, A. (1999). Robust solutions of uncertain linear programs. *Operations research letters*, 25(1):1–13.

[Ben-Tal and Nemirovski, 2000] Ben-Tal, A. and Nemirovski, A. (2000). Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical programming*, 88(3):411–424.

[Benders, 1962] Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik*, 4(1):238–252.

[Berg and Denton, 2014] Berg, B. and Denton, B. (2014). Fast approximations for online scheduling of outpatient procedure centers.

[Bertsimas et al., 2011] Bertsimas, D., Brown, D. B., and Caramanis, C. (2011). Theory and applications of robust optimization. *SIAM review*, 53(3):464–501.

[Bertsimas and Caramanis, 2010] Bertsimas, D. and Caramanis, C. (2010). Finite adaptability in multistage linear optimization. *Automatic Control, IEEE Transactions on*, 55(12):2751–2766.

[Bertsimas and Patterson, 1998] Bertsimas, D. and Patterson, S. S. (1998). The air traffic flow management problem with enroute capacities. *Operations research*, 46(3):406–422.

[Bertsimas and Sim, 2003] Bertsimas, D. and Sim, M. (2003). Robust discrete optimization and network flows. *Mathematical programming*, 98(1):49–71.

[Bertsimas and Sim, 2004] Bertsimas, D. and Sim, M. (2004). The price of robustness. *Operations research*, 52(1):35–53.

[Birge, 1997] Birge, J. R. (1997). State-of-the-art-survey-stochastic programming: Computation and applications. *INFORMS journal on computing*, 9(2):111–133.

[Birge and Louveaux, 2011] Birge, J. R. and Louveaux, F. (2011). *Introduction to stochastic programming*. Springer.

[Birge and Louveaux, 1988] Birge, J. R. and Louveaux, F. V. (1988). A multicut algorithm for two-stage stochastic linear programs. *European Journal of Operational Research*, 34(3):384–392.

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

[Demeulemeester et al., 2013] Demeulemeester, E., Beliën, J., Cardoen, B., and Samudra, M. (2013). Operating room planning and scheduling. In *Handbook of Healthcare Operations Management*, pages 121–152. Springer.

[Deng et al., 2014] Deng, Y., Shen, S., and Denton, B. (2014). Chance-constrained surgery planning under uncertain or ambiguous surgery duration. *Available at SSRN 2432375*.

[Denton et al., 2010] Denton, B. T., Miller, A. J., Balasubramanian, H. J., and Huschka, T. R. (2010). Optimal allocation of surgery blocks to operating rooms under uncertainty. *Operations research*, 58(4-part-1):802–816.

[El Ghaoui et al., 1998] El Ghaoui, L., Oustry, F., and Lebret, H. (1998). Robust solutions to uncertain semidefinite programs. *SIAM Journal on Optimization*, 9(1):33–52.

[Erdogan et al., 2011] Erdogan, S. A., Denton, B. T., Cochran, J., Cox, L., Keskinocak, P., Kharoufeh, J., and Smith, J. (2011). Surgery planning and scheduling. *Wiley Encyclopedia of operations research and management science. http://ca. wiley. com/WileyCDA/Section/id-380199. html*.

[Fei et al., 2008] Fei, H., Chu, C., Meskens, N., and Artiba, A. (2008). Solving surgical cases assignment problem by a branch-and-price approach. *International Journal of Production Economics*, 112(1):96–108.

[Ferrand et al., 2014] Ferrand, Y. B., Magazine, M. J., and Rao, U. S. (2014). Managing operating room efficiency and responsiveness for emergency and elective surgeries — a literature survey. *IIE Transactions on Healthcare Systems Engineering*, 4(1):49–64.

[Fu et al., 2014] Fu, Y., Sun, J., Lai, K., and Leung, J. W. (2014). A robust optimization solution to bottleneck generalized assignment problem under uncertainty. *Annals of Operations Research*, pages 1–11.

[Fügener et al., 2014] Fügener, A., Hans, E. W., Kolisch, R., Kortbeek, N., and Vanberkel, P. T. (2014). Master surgery scheduling with consideration of multiple downstream units. *European Journal of Operational Research*.

[Gabrel et al., 2014a] Gabrel, V., Lacroix, M., Murat, C., and Remli, N. (2014a). Robust location transportation problems under uncertain demands. *Discrete Applied Mathematics*, 164:100–111.

[Gabrel et al., 2014b] Gabrel, V., Murat, C., and Thiele, A. (2014b). Recent advances in robust optimization: An overview. *European Journal of Operational Research*, 235(3):471–483.

[Gallo and Ülkücü, 1977] Gallo, G. and Ülkücü, A. (1977). Bilinear programming: an exact algorithm. *Mathematical Programming*, 12(1):173–194.

[Green, 2012] Green, L. V. (2012). Om forum-the vital role of operations analysis in improving healthcare delivery. *Manufacturing & Service Operations Management*, 14(4):488–494.

[Guerriero and Guido, 2011] Guerriero, F. and Guido, R. (2011). Operational research in the management of the operating theatre: a survey. *Health care management science*, 14(1):89–114.

[Gul et al., 2012] Gul, S., Denton, B. T., and Fowler, J. W. (2012). A multi-stage stochastic integer programming model for surgery planning. *Technical Report*.

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

[Hanasusanto et al., 2015] Hanasusanto, G. A., Kuhn, D., and Wiesemann, W. (2015). K-adaptability in two-stage robust binary programming. *Operations Research*, 63(4):877–891.

[Hoffman and Padberg, 2001] Hoffman, K. and Padberg, M. (2001). Set covering, packing and partitioning problemsset covering, packing and partitioning problems. In *Encyclopedia of Optimization*, pages 2348–2352. Springer.

[Hsu et al., 2003] Hsu, V. N., de Matta, R., and Lee, C.-Y. (2003). Scheduling patients in an ambulatory surgical center. *Naval Research Logistics (NRL)*, 50(3):218–238.

[Jackson, 2002] Jackson, R. L. (2002). The business of surgery. managing the or as a profit center requires more than just it. it requires a profit-making mindset, too. *Health management technology*, 23(7):20.

[Jonnalagadda et al., 2005] Jonnalagadda, R., Walrond, E., Hariharan, S., Walrond, M., and Prasad, C. (2005). Evaluation of the reasons for cancellations and delays of surgical procedures in a developing country. *International journal of clinical practice*, 59(6):716–720.

[Kelley, 1960] Kelley, Jr, J. E. (1960). The cutting-plane method for solving convex programs. *Journal of the Society for Industrial & Applied Mathematics*, 8(4):703–712.

[Küçükyavuz, 2011] Küçükyavuz, S. (2011). Mixed-integer optimization approaches to deterministic and stochastic inventory management. *INFORMS TutORials in Operations Research (ed. J. Geunes)*, 8:90–105.

[Lamiri et al., 2009] Lamiri, M., Grimaud, F., and Xie, X. (2009). Optimization methods for a stochastic surgery planning problem. *International Journal of Production Economics*, 120(2):400–410.

[Macario et al., 1995] Macario, A., Vitez, T. S., Dunn, B., and McDonald, T. (1995). Where are the costs in perioperative care?: Analysis of hospital costs and charges for inpatient surgical care. *Anesthesiology*, 83(6):1138–1144.

[Marcon and Dexter, 2006] Marcon, E. and Dexter, F. (2006). Impact of surgical sequencing on post anesthesia care unit staffing. *Health Care Management Science*, 9(1):87–98.

[Martin et al., 2011] Martin, A., Lassman, D., Whittle, L., Catlin, A., et al. (2011). Recession contributes to slowest annual rate of increase in health spending in five decades. *Health Affairs*, 30(1):11–22.

[Min and Yih, 2010] Min, D. and Yih, Y. (2010). Scheduling elective surgery under uncertainty and downstream capacity constraints. *European Journal of Operational Research*, 206(3):642–652.

[Neyshabouri and Berg, 2016] Neyshabouri, S. and Berg, B. (2016). Two-stage robust optimization approach to elective surgery and downstream capacity planning. *European Journal of Operational Research*, doi: 10.1016/j.ejor.2016.11.043.

[Öncan, 2007] Öncan, T. (2007). A survey of the generalized assignment problem and its applications. *INFOR: Information Systems and Operational Research*, 45(3):123–141.

[Pham and Klinkert, 2008] Pham, D.-N. and Klinkert, A. (2008). Surgical case scheduling as a generalized job shop scheduling problem. *European Journal of Operational Research*, 185(3):1011–1025.

[Rawls and Turnquist, 2010] Rawls, C. G. and Turnquist, M. A. (2010). Pre-positioning of emergency supplies for disaster response. *Transportation research part B: Methodological*, 44(4):521–534.

[Ruszczyński, 1997] Ruszczyński, A. (1997). Decomposition methods in stochastic programming. *Mathematical programming*, 79(1-3):333–353.

[Santoso et al., 2005] Santoso, T., Ahmed, S., Goetschalckx, M., and Shapiro, A. (2005). A stochastic programming approach for supply chain network design under uncertainty. *European Journal of Operational Research*, 167(1):96–115.

[Savelsbergh, 1997] Savelsbergh, M. (1997). A branch-and-price algorithm for the generalized assignment problem. *Operations research*, 45(6):831–841.

[Shapiro et al., 2014] Shapiro, A., Dentcheva, D., et al. (2014). *Lectures on stochastic programming: modeling and theory*, volume 16. SIAM.

[Shylo et al., 2012] Shylo, O. V., Prokopyev, O. A., and Schaefer, A. J. (2012). Stochastic operating room scheduling for high-volume specialties under block booking. *INFORMS Journal on Computing*, 25(4):682–692.

[Sobolev et al., 2005] Sobolev, B. G., Brown, P. M., Zelt, D., and FitzGerald, M. (2005). Priority waiting lists: Is there a clinically ordered queue? *Journal of evaluation in clinical practice*, 11(4):408–410.

[Soyster, 1973] Soyster, A. L. (1973). Technical note—convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations research*, 21(5):1154–1157.

[Strum et al., 2000] Strum, D. P., May, J. H., and Vargas, L. G. (2000). Modeling the uncertainty of surgical procedure times: comparison of log-normal and normal models. *Anesthesiology*, 92(4):1160–1167.

[Thiele et al., 2009] Thiele, A., Terry, T., and Epelman, M. (2009). Robust linear optimization with recourse. *Rapport technique*, pages 4–37.

[Truong et al., 2013] Truong, V.-A., Wang, X., and Liu, N. (2013). Integrated scheduling and capacity planning with considerations for patients' length-of-stays.

[Utzolino et al., 2010] Utzolino, S., Kaffarnik, M., Keck, T., Berlet, M., and Hopt, U. T. (2010). Unplanned discharges from a surgical intensive care unit: Readmissions and mortality. *Journal of critical care*, 25(3):375–381.

[van Oostrum et al., 2008] van Oostrum, J. M., Van Houdenhoven, M., Hurink, J. L., Hans, E. W., Wullink, G., and Kazemier, G. (2008). A master surgical scheduling approach for cyclic scheduling in operating room departments. *OR spectrum*, 30(2):355–374.

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

[Wallace and Ziemba, 2005] Wallace, S. W. and Ziemba, W. T. (2005). *Applications of stochastic programming*, volume 5. Siam.

[Wolsey and Nemhauser, 2014] Wolsey, L. A. and Nemhauser, G. L. (2014). *Integer and combinatorial optimization*. John Wiley & Sons.

[Zeng and Zhao, 2013] Zeng, B. and Zhao, L. (2013). Solving two-stage robust optimization problems using a column-and-constraint generation method. *Operations Research Letters*, 41(5):457–461.

Curriculum Vitae

Saba Neyshabouri received his Bachelor of Science in Industrial Engineering in Iran from Sharif University of Technology in 2010. He joined the Systems Engineering and Operations Research department at George Mason University the same year to pursue his Ph.D. He obtained his Masters of Science in Operations Research from George Mason University in 2013. He graduated with Ph.D. in Systems Engineering and Operations Research from George Mason University in 2016.