Appointment Scheduling with Discrete Random Durations and Applications

by

Mehmet Atilla Begen

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

The Faculty of Graduate Studies

(Business Administration)

THE UNIVERSITY OF BRITISH COLUMBIA
(Vancouver)

April 2010

© Mehmet Atilla Begen 2010
Abstract

We study scheduling of jobs on a highly utilized resource when the processing durations are stochastic and there are significant underage (resource idle-time) and overage (job waiting and/or resource overtime) costs. Our work is motivated by surgery scheduling and physician appointments. We consider several extensions and applications.

In the first manuscript, we determine an optimal appointment schedule (planned start times) for a given sequence of jobs (surgeries) on a single resource (operating room, surgeon). Random processing durations are integers and given by a discrete probability distribution. The objective is to minimize the expected total underage and overage costs. We show that an optimum solution is integer and can be found in polynomial time.

In the second manuscript, we consider the appointment scheduling problem under the assumption that the duration probability distributions are not known and only a set of independent samples is available, e.g., historical data. We develop a sampling-based approach and determine bounds on the number of independent samples required to obtain a provably near-optimal solution with high probability.

In manuscript three, we focus on determining the number of surgeries for an operating room in an incentive-based environment. We explore the interaction between the hospital and the surgeon in a game theoretic setting, present empirical findings and suggest incentive schemes that the hospital may offer to the surgeon to reduce its idle time and overtime costs.

In manuscript four, we consider an application to inventory management in a supply chain context. We introduce advance multi-period quantity commitment with stochastic characteristics (demand or yield) and describe several real-world applications. We show these problems can be solved as special cases of the appointment scheduling problem.

In manuscript five, an appendix, we develop an alternate solution approach for the appointment scheduling problem. We find a lower bound value, obtain a subgradient of the objective function, and develop a special-purpose integer rounding algorithm combining discrete convexity and non-smooth convex optimization methods.
# Table of Contents

Abstract .................................................................................. ii

Table of Contents ................................................................. iii

List of Tables ........................................................................... vi

List of Figures ........................................................................... vii

Acknowledgements .................................................................. viii

Dedication ................................................................................ x

Co-Authorship Statement ......................................................... xi

1 Introduction ........................................................................... 1

  1.1 Motivation and Appointment Scheduling ................................. 1

  1.2 Overview of the Thesis ....................................................... 6

    1.2.1 Chapter 2: Appointment Scheduling ................................. 6

    1.2.2 Chapter 3: A Sampling-Based Approach ............................ 7

    1.2.3 Chapter 4: Incentive-Based Surgery Scheduling ................. 9

    1.2.4 Chapter 5: Advance Quantity Commitment ....................... 10

    1.2.5 Appendix A: Minimizing a Discrete-Convex Function .......... 11

  1.3 Outline of the Thesis ......................................................... 12

  1.4 Bibliography ....................................................................... 13

2 Appointment Scheduling ....................................................... 16

  2.1 Introduction and Motivation ............................................... 16

  2.2 Related Literature ............................................................ 20

  2.3 Assumptions and Notation ................................................ 22
2.4 Basic Properties ................................................. 24
2.5 Optimality of an Integer Appointment Vector .................... 26
2.6 L-convexity ....................................................... 32
2.7 Algorithms ....................................................... 39
2.8 Objective Function with a Due Date ............................... 40
2.9 No-shows and Emergency Jobs .................................. 44
2.10 Current Work, Future Work and Conclusion .................... 46
2.11 Bibliography .................................................... 49

3 Sampling Approach ................................................... 52
3.1 Introduction and Motivation ...................................... 52
3.2 Appointment Scheduling Problem ................................ 58
3.3 Convexity ......................................................... 60
3.4 Subdifferential Characterization .................................. 63
3.5 Sampling Approach .............................................. 71
3.6 Conclusion ......................................................... 75
3.7 Proofs ............................................................. 76
3.8 Bibliography ....................................................... 86

4 Incentive-Based Surgery Scheduling .................................. 90
4.1 Introduction ....................................................... 90
4.2 Problem Description and Motivation ............................... 94
4.3 The Model ........................................................ 102
4.4 Misalignment of Incentives ...................................... 106
  4.4.1 Deciding the Number of Surgeries ......................... 106
4.5 Contracts ........................................................ 109
  4.5.1 Hospital has Complete Information and Coercive Power ... 110
  4.5.2 Take-It-or-Leave-It Offer .................................. 111
  4.5.3 Three-Part Contract ........................................ 112
  4.5.4 Implementing the Two Contracts ......................... 115
  4.5.5 Welfare Considerations .................................... 116
4.6 Dependent Surgeries with Identical Realizations ................ 117
  4.6.1 Preliminaries and Misalignment of Incentives ............. 118
4.6.2 Risk-Averse Surgeon ........................................ 120
4.6.3 Intuitive Discussion ........................................ 121
4.6.4 Discrete Time Distribution with Two Values ............... 122
4.6.5 Sufficient Conditions on Cost Coefficients .................. 125
4.7 Conclusion and Future Directions ............................. 125
4.8 Proofs ..................................................... 126
4.9 Bibliography ................................................ 133

5 Advance Quantity Commitments ................................. 136
5.1 Introduction ................................................. 136
5.2 Description of Appointment Scheduling Problem .......... 138
5.3 An Inventory Model with Commitments ...................... 140
5.4 A Production Model with Commitments ..................... 145
5.5 Conclusion .................................................. 149
5.6 Bibliography ................................................ 151

6 Concluding Remarks ............................................ 152
6.1 Bibliography ................................................ 155

Appendices

A Minimizing a Discrete-Convex Function ......................... 156
A.1 Introduction and Motivation ................................ 156
A.2 Lower Bound (on the Value) and Computation of $F$ ....... 158
A.3 Obtaining a Subgradient in Polynomial Time ................. 163
  A.3.1 Probability Computations ................................. 163
  A.3.2 Complexity of Subgradient Computations .............. 171
  A.3.3 Obtaining a Subgradient Fast (in Polynomial Time) ... 172
A.4 Algorithms .................................................. 180
A.5 Conclusion and Future Work ................................ 183
A.6 Bibliography ................................................ 184
List of Tables

Table 4.1  Estimates of Daily Overtime Probability per OR . . . . . . . . . . . . 98
Table 5.1  Comparison of the Appointment Scheduling and Inventory Models . . 142
Table 5.2  Comparison of the Appointment Scheduling and Production Models . 147
List of Figures

Figure 1.1  Surgery Durations by Surgical Specialty ............................. 2
Figure 1.2  Duration Distribution of a Simple Hernia Operation ............. 2
Figure 1.3  Surgery Scheduling Process ............................................. 4
Figure 1.4  An Instance with Three Surgeries ............................... 5

Figure 2.1  Surgery Durations ..................................................... 17
Figure 2.2  A Three-Job Instance, and A Realization of the Processing Durations 18
Figure 2.3  An Example Schedule with Emergency Jobs ...................... 45

Figure 4.1  Healthcare Players and Tasks ....................................... 91
Figure 4.2  Surgery Scheduling Process ........................................ 94
Figure 4.3  The Third Level of Surgery Scheduling Process ................. 95
Figure 4.4  Duration Distribution of a Simple Hernia Operation ........... 97
Figure 4.5  Actual and Scheduled Surgery Durations .......................... 98
Figure 4.6  Daily Average Overtime Minutes and Probability of Overtime per OR 99
Figure 4.7  Existence of $m$ ...................................................... 123
Figure 4.8  Illustration of $n_R \leq n_S$ with Two $t$ Values .............. 124
Figure 4.9  Illustration of Condition Eq(4.16) ................................. 125

Figure 5.1  A Realization of Inventory Levels for 6 Periods ................ 143
Figure A.1  Event $\{I_5 = \{1,2,3,5\}\}$ Visualization ..................... 164
Figure A.2  Event $\{I_8 = \{3,5,6\}\}$ Visualization ..................... 165
Acknowledgements

This thesis would not have been what it is without the encouragement and generous support from many individuals. I am grateful to every person who has offered help to me and contributed in some way to the process leading to my dissertation.

It is difficult to overstate my gratitude to my supervisor, Maurice Queyranne. With his patience, inspiration, and great efforts to explain things clearly and simply, he has been a great mentor throughout my Ph.D. studies. I would like to thank him for our long research meetings, all of the advice and insightful direction he has passed along. I wish one day I can be a researcher like him and supervise my own Ph.D. students in the same way.

The second individual I would like to express my appreciation to is Marty Puterman. He persuaded me to come to Canada for the COE program in the first place and encouraged me to pursue a Ph.D. degree. I would like to thank him for all of the advice, encouragement and support he has given during my studies and my employment at UBC as well as introducing me to the world of applied operations research.

I have also been fortunate enough to have a great thesis committee. Besides Maurice Queyranne and Marty Puterman, I had the privilege to have Ralph Winter and Mahesh Nagarajan as my thesis committee members. I thank them for being supportive of my research, for their encouragement and constructive critiques that have greatly contributed to my thesis. Furthermore, I thank Ralph Winter for introducing me to the topic of Contract Theory and getting started the seed ideas of Chapter 4 in a term paper. And I thank Mahesh Nagarajan for providing me advice on my research and other academic issues when it was most needed.

Let me also thank a few individuals who contributed to this research. I start with Retsef Levi who suggested using a sampling-based approach for surgery scheduling in his timely visit to UBC. I also thank him for being a co-author for Chapter 3. Second, my appreciation goes to Chris Ryan who has been a very good friend, a great research collaborator and a co-author of Chapter 4. For Chapter 4, I also thank Mati Dubrovinsky, Zhongzhi Song,
and Gavin Yang and for their interesting and fruitful discussions and feedback, and give my thanks to the management and system analysts at a local hospital for their support and help in obtaining data. Last but not least, I thank Philip D. Loewen and Steven Shechter for their comments on the thesis.

Since I first came to UBC, the campus and its tenants offered me a pleasant, friendly and motivating work environment. I would like to thank OPLOG faculty (in particular Steven Shechter, Derek Atkins, Harish Krishnan, Danny Granot, Anming Zhang and Tom McCormick), Elaine Cho (who has guided me through the university bureaucracy with her endless goodwill), Michelle Medalla, Joel Feldman, Geoffrey Blair, the COE staff and others (in particular Fredrik Odegaard, Jonathan Patrick, Mariel Lavieri, Antoine Saure V., Pablo Santibanez, Vincent Chow, Abelardo Mayoral, Steven Kabanuk Greg Werker, Anita Parkinson). Furthermore, thanks to UBC and NSERC for their financial support during my studies.

Last but not least, I want to thank my mom, dad, and brother. I am forever indebted for their understanding, endless patience and encouragement when it was most required. It is to them that I dedicate this work.
To my family,

Sündüz, Cevdet and Ali Cengiz.
Co-Authorship Statement

Chapter 2, Chapter 5 and Appendix A are manuscripts co-authored with the candidate’s supervisor, Maurice Queyranne. The identification and design of the research program, for these papers were carried out jointly. Research, analysis and manuscript preparation were performed by the candidate with close supervision from Maurice Queyranne.

Chapter 3 is co-authored with Maurice Queyranne and Retsef Levi. The identification and design of the research program, for this paper were carried out jointly. Research, analysis and manuscript preparation were performed by the candidate with close supervision from Maurice Queyranne and with comments on revisions provided by Retsef Levi.

Chapter 4 is co-authored with Maurice Queyranne and Chris Ryan. The design of the research program, research, analysis and manuscript preparation for this paper were carried out jointly with Chris Ryan with close supervision from Maurice Queyranne. The identification of the initial research question and preliminary data analysis were performed by the candidate.
1 Introduction

We begin with motivation and introduction of the appointment scheduling problem in Section 1.1. In Section 1.1, we also discuss how and where appointment scheduling fits in the context of surgery scheduling and give a summary of related previous work. Then in Section 1.2, we give an overview of the thesis. Finally, Section 1.3 concludes the chapter with an outline of the thesis.

1.1 Motivation and Appointment Scheduling

Healthcare is one of the biggest industries in North America. Canada was expected to spend $148 billion on healthcare in 2006 [13], which accounts for more than 10% of its GDP. In the United States the situation is similar, in 2006 it accounted for 15.3% of GDP [8]. Healthcare challenges, on one end costs and on the other end demand, are growing not only in Canada but in almost every country in the world [6]. To address these challenges, one may think of either increasing available resources (capacity), limiting demand or finding ways to improve efficiency [24]. In most cases, increasing capacity or limiting demand may not be possible, and even if they are possible the challenges may require a deeper analysis and efficiency improvements. One way to improve healthcare operations is with effective scheduling of resources and patients who need them. Scheduling issues become more important and challenging when there is uncertainty present in the system. Uncertainty may be involved with patients (e.g., priority levels and arrivals [25]), resources (e.g., availability of a vaccine) or any other aspect of the healthcare operations (e.g., surgery durations [10]). In our applied healthcare projects, we also observed uncertainty of patient arrivals and surgery durations [32]. For instance, Figures 1.1 and 1.2 show how variable surgery durations can be. Figure 1.1 shows an example of surgery durations (operating room (OR) time in minutes) by surgical specialty and Figure 1.2 depicts duration distribution of a simple hernia operation. (Data for these figures comes from local hospitals.)
Figure 1.1: Surgery Durations by Surgical Specialty

Uncertainty makes the scheduling and capacity allocation decisions more complex and challenging. In such an environment, one needs to find a balance in the tradeoff allocating too much (more idle-time but less patient waiting time) or too little (more patient waiting time but less overtime) capacity.

Figure 1.2: Duration Distribution of a Simple Hernia Operation

Motivated by the surgeries, oncologist consultations and radiation therapy treatments for cancer patients, we take an in-depth look at the appointment scheduling of jobs (e.g.,
surgeries, exams) of a highly utilized processor (e.g., OR, physician) when the job durations are stochastic and there are significant overage (job waiting and/or processor overtime) and underage (processor idle-time) costs. For a given sequence of jobs on a single processor, we determine an optimal appointment schedule (planned start times) minimizing the expected total underage and overage costs.

Before we get into the details of the appointment scheduling problem, we first take a look at the surgery scheduling process to see how and where appointment scheduling fits in this context. In practice, scheduling surgeries in a medical facility is a complex and important process, and the choice of schedule directly impacts the overall performance of the system [32]. The surgery scheduling process (for elective cases) is usually considered as a three-level process [2, 3, 23]. We can classify these three levels as the strategic, tactical and operational stages of the surgery scheduling process respectively. Figure 1.3 gives an overview of the process in terms of decisions, decision maker and decision level. The first level defines and assigns the OR time among the surgical specialties, usually called mix planning. A surgical OR block schedule\(^1\) is developed at the second level. Finally, in the third level, individual cases are scheduled on a daily basis, also known as patient mix. It is at this level that variability in surgery durations plays a key role and where one determines the number of surgeries to perform in a block, the sequence the surgeries performed and the planned start times (appointment times) of the surgeries. Ideally, one should consider all these three levels of decisions simultaneously and not in isolation. However, the practical applications and mathematical challenges force practitioners and academics to work on these problems individually. In this thesis, we concentrate in level three and for the appointment scheduling problem we assume that the number of jobs (surgeries) and a sequence are already determined and given.

For example, in the case of surgeries, for a given set of surgeries and their sequence, an appointment schedule, i.e., planned start times, needs to be prepared. This is an important and challenging task since surgery appointment schedule has a direct impact on amount of overtime and idle-time of ORs [10]. OR overtime can be costly since it involves staff overtime as well as additional overhead costs, on the other hand, idle-time costs can also

---

\(^{1}\)An OR block schedule is simply a table that assigns each specialty surgery time in ORs on each day. The times are called blocks. The OR block schedule is sometimes called the master surgical schedule (see Figure 2 of [32] for a sample OR block schedule).
be high due to the opportunity cost of unused capacity, especially important in a Canadian context due to important political and social issues related to the length of surgical waiting lists [34].

An appointment schedule assigns an allocated duration by specifying the appointment time of each surgery at which the required resource(s) (e.g., OR, surgeon, healthcare personnel and equipment), and the patient will be available. However, due to the uncertain surgery durations, some surgeries may finish earlier whereas some others may finish later. In the latter case, the next surgery has to wait for the preceding surgery to complete and will start later than its original appointment time. As the appointment times have to be determined in advance, there are only limited recourse options when the actual duration of a procedure differs from its planned value. When a surgery finishes earlier than the next surgery’s appointment time, there is under-utilization of the healthcare resources. On the other hand, if a procedure finishes later than the next procedure’s appointment date, there may be some overtime of the healthcare resources and is waiting of the next procedure. Therefore, there is an important trade-off between under-utilization, overtime and patient waiting times. We are interested in finding a schedule that minimizes the expected total cost of resources under-utilization, overtime of resources and patient waiting. Generating such a schedule is more challenging but more valuable and useful when processing durations have more variability. The need for a good schedule is crucial, and savings from such a schedule may be significant. Figure 1.4 shows an instance with 3 surgeries $G, B, R$ to be processed in this order. An appointment schedule $(A_G, A_B, A_R)$ is given. Once the processing starts,
due to the random processing durations, some surgeries may be early whereas some others may be late as shown in Figure 1.4. \((C_G, C_B, C_R\) denotes completion times of surgeries.\)

**Figure 1.4: An Instance with Three Surgeries**

In the last five decades, there has been a tremendous interest in appointment scheduling not only in healthcare and service industries [7, 5, 35] but also in other areas such as production and transportation [12, 31]. While our goal is to provide an overview of the prior work on appointment scheduling, here we can only give a glimpse of it. However, in the subsequent chapters, we survey the related work about the problems under discussion in more detail. Weiss [36] recognized that the appointment scheduling problem has a closed form solution when there is only a single job, and it coincides with the well known newsvendor [23] solution from inventory theory. However, the problem departs from newsvendor characteristics and solution methods in the case of multiple jobs [29]. In multi-period newsvendor problem, naturally, decisions are taken at each period sequentially. Whereas in appointment scheduling, one needs to have a schedule before any processing can start, i.e., one determines all the decision variables (appointment times) simultaneously at the beginning of the planning horizon, i.e., at time zero. In terms of solution methods, we see studies based on stochastic programming [28, 10], queuing theory [35, 5, 16], simulation and other methods, see [7] and references therein. Cayirli and Veral [7] classify the literature in terms of methodologies and modeling aspects considered, and provide a discussion of performance measures. The authors conclude that the existing literature provides very situation-specific solutions and does not offer generally applicable and portable methodologies for appointment systems design in outpatient scheduling. We finally would like to point out some differences between appointment scheduling and single machine scheduling [27].
Unlike machine scheduling, in appointment scheduling a sequence is given and the release dates are the decision variables. Furthermore, the objective function of the appointment scheduling problem is different than the objective functions of classical machine scheduling problems. Processing durations are usually deterministic in machine scheduling problems but random processing durations are also studied in the literature [27].

Appointment scheduling problem can be modeled as a multistage stochastic programming problem [28, 10, 29], but there are significant computational difficulties due to the need for multidimensional numerical integration, e.g., even computing expected cost for a given schedule is difficult. Hence, heuristic methods have to be developed for realistic size problems. To the best of our knowledge, all the analytical studies that we are aware of about appointment scheduling, even the ones with discrete epochs [16, 5, 35] for job arrivals, use continuous job duration distributions.

1.2 Overview of the Thesis

1.2.1 Chapter 2: Appointment Scheduling with Discrete Random Durations

In Chapter 2, we study a discrete time version of the appointment scheduling problem, i.e., the processing durations are integer and given by a discrete probability distribution. This assumption fits many applications, for example, surgeries and physician appointments are scheduled on minute(s) basis (usually a block of certain minutes). (For instance, one 20 minute physician appointment could be two blocks of 10 minutes.) We establish discrete convexity [21] properties of the objective function (under a mild condition on cost coefficients) and show that there exists an optimal integer appointment schedule minimizing the objective. This result is important as it allows us to optimize only over integer appointment schedules without loss of optimality. All these results on the objective function and optimal appointment schedule enable us to develop a polynomial time algorithm, based on discrete convexity [22], that, for a given processing sequence, finds an appointment schedule minimizing the total expected cost. When processing durations are stochastically independent we evaluate the expected cost for a given processing order and an integer appointment schedule in polynomial time. Independent processing durations lead to faster algorithms. Our modeling framework can include a given due date for the end of the processing of all
jobs (e.g., end of day for an operating room) after which overtime is incurred, instead of letting the model choose an end date. We also extend the analysis to include no-shows and some emergency jobs. Our setting is quite general, and it could be applied to various real-life scenarios (in healthcare and other areas) including surgeries, MRI exams, physician and specialist consultations, radiation therapy, project scheduling, container vessel and terminal operations, gate and runway scheduling of aircrafts in an airport. We believe our approach is sufficiently generic and portable to solve the appointment scheduling problem efficiently.

1.2.2 Chapter 3: A Sampling-Based Approach to Appointment Scheduling

In Chapter 2, we assume complete information of job duration distributions, i.e., there is an underlying discrete probability distribution for job durations (true distribution), and this distribution is available and known fully. This may be the case for some applications. However, for others, the true duration distributions may not be known but its (past) realizations or some samples may be available. One good example for such an application comes from healthcare; hospitals and surgeons usually have some data (historical) available on the length of surgeries but no one knows what the true distribution for a certain type of surgery is.

In Chapter 3, we consider the problem of appointment scheduling with discrete random durations under the assumption that the true duration probability distributions are not known and only a set of independent samples is available. These samples may correspond to historical data, for example daily observations of surgery durations. We show that the objective function of the appointment scheduling problem is convex (as a function of continuous appointment vectors) under a simple and sufficient condition on cost coefficients. Under this condition we characterize the subdifferential\(^2\) of the objective function with a closed-form formula. We use this formula to determine bounds on the number of independent samples required to obtain a provably near-optimal solution with high probability, i.e., the cost of the sampling-based optimal schedule is with high probability no more than \((1 + \epsilon)\) times the cost of the optimal schedule that is computed based on the true distribution. Our bound for number of required samples is polynomial in number of jobs, accuracy level, confidence level and cost coefficients.

\(^2\)The set of all subgradients at a point of a convex function [30, 14].
There has been much interest for studying stochastic models, especially the newsvendor problem and its multiperiod extension, with partial probabilistic characterization. When the true distribution is not fully known then the question is how to find a “good solution”. Depending on how much is known about the true distribution(s) different approaches are possible, e.g., parametric and non-parametric. One may know the family of the true distribution but be uncertain about its parameters. This is called parametric approach, e.g., see [11, 19] and the references therein. If there are no assumptions on the true distribution, i.e., no prior assumptions on its family or its parameters, then the approach becomes non-parametric, e.g., see [17, 33, 26, 4].

Our approach is non-parametric and we employ sample average approximation (SAA) [33] to solve the appointment scheduling problem with samples. In other words, we use available samples to form an empirical distribution and find an optimal solution with respect to this empirical solution, i.e., sampling solution. Then we use the subdifferential characterization of the objective function (Section 3.4) and the well-known Hoeffdings inequality [15] to determine the number of samples required to guarantee that there will exist a (sufficiently) small (in terms of the specified accuracy level) subgradient at the sampling solution with high probability (i.e., at least the specified confidence level). As a final step we show that the objective value (w.r.t. the true distribution) of the sampling solution is no more than (1 + the accuracy level) of the true optimal value with probability at least the confidence level.

For our sampling-based approach, job durations may not necessarily be independent but we require samples to be independent. In other words, each sample is a vector of durations where each coordinate corresponds to job duration and these vectors are independent. Independence assumption of probability distributions (e.g., job durations) is common but we do not require it in our sampling-based analysis. To the best of our knowledge Chapter 3 is the first to address the appointment scheduling problem when the probability distributions of durations are unknown. We develop a sampling-approach for the appointment scheduling problem which is a stochastic non-linear integer program. Furthermore, we believe Chapter 3 presents the first rigorous analysis of the convexity of the objective function of appointment scheduling problem with the simple sufficient condition. Last but not least, we characterize the set of all subgradients, i.e., the subdifferential at a given appointment
date vector, with a closed-form formula\(^3\).

### 1.2.3 Chapter 4: Incentive-Based Surgery Scheduling: Determining Optimal Number of Surgeries

In Chapter 4, we look at a different but related problem of determining the number of surgeries for an OR block with a focus on the incentives of the parties involved (hospital and surgeon). We investigate the interaction between the hospital and the surgeon in a game theoretic setting, present empirical findings on surgery durations and suggest payment schemes that the hospital may offer to the surgeon to reduce its (idle and especially overtime) costs. In particular, we investigate the commonly observed situation reported in the literature [23] and observed empirically (Section 4.2) that surgeons over-schedule their allotted OR time, i.e., they schedule too many surgeries for their OR time. Olivares et al. [23] reports that the amount of schedule overruns are mainly caused by incentive conflicts and over-confidence. We take a systematic look at this and provide a model by which these incentive conflicts can be identified and effectively analyzed. Based on historical data analysis (Section 4.2) we see that in 81% of the surgeries, actual durations were longer than booked/scheduled durations. This high percentage suggests that duration of individual surgeries are often underestimated. One may ask how this phenomenon actually effects the daily overall performance of an OR block; i.e., amount of overtime for an OR as well as the likelihood of an OR to go overtime. To answer this question, we look the data at an operating room level. For each OR, we compute daily average of scheduled and overtime OR minutes and find the percentage of overtime, i.e., the ratio of overtime OR minutes and scheduled OR minutes. We find that the overtime amount is well over 20% for each OR (Figure 4.6). We also find the percentage of days that each OR has an overtime to estimate the probability of daily overtime for each OR (Table 4.1). The smallest of these numbers is 75%. These empirical findings, significant amount and high likelihood of overtime, suggest that the cost of overtime can be substantial. If an OR can be managed in such a way that overtime is decreased then this may translate to immediate and significant cost savings. Additionally, savings from reduction in overtime costs may be used to increase hospital

---

\(^3\)This is unusual since only a single subgradient may be obtained in most applications. We make use of this subdifferential characterization in finding optimum appointment schedules by using non-smooth optimization methods in Appendix A.
resources such as regular OR time, recovery and intensive care beds. We argue that these observations can be explained by the incentive of surgeons to take advantage of fee-for-service\textsuperscript{4} payment structure for surgeries performed combined with the fact surgeons do not bear overtime costs at the hospital level. This creates a cost which is borne by the hospital who operates the OR and pays surgery support staff. Thus we argue that the hospital has an incentive to limit the number of surgeries performed by surgeons to reduce overtime expenditures. We explore this misalignment of incentives – for the surgeon to over-schedule and the hospital to control overtime costs – in a game theoretic setting. We characterize analytically the number of surgeries that minimizes hospital costs, find conditions when this number is less than surgeon’s preference, and propose contracts that induce the surgeon to schedule a number of surgeries more aligned with the goals of the hospital. Depending on how much power the hospital has over surgeons and how much information is available to the hospital, we suggest several contracts that hospital might consider.

1.2.4 Chapter 5: Advance Multi-Period Quantity Commitment and Appointment Scheduling

As discussed briefly above there is a connection between the celebrated newsvendor problem and the appointment scheduling problem. If we have only a single job (surgery), i.e., \( n = 1 \), then the appointment scheduling problem becomes the newsvendor problem. This was first recognized by Weiss [36]. In Chapter 5, we investigate this relationship in the case of multiple jobs. We introduce advance multi-period quantity (order or supply) commitment problems with random characteristics (demand or yield). There are underage and overage costs if there is a mismatch between committed and realized quantities. All quantity decisions (how much to order or supply in each of the next \( n \) periods) are needed now, before any realization of demand or yield. The objective is to maximize the total expected profit after \( n \) periods. We establish a link between these problems and the appointment scheduling problem (as given in Chapter 2). We show that these problems can be studied and solved as special cases of the appointment scheduling problem.

In a supply chain, uncertainty effects (e.g., due to stochastic demand or random yield)
are something that players would like to minimize or pass to others. Consider a buyer and a supplier where the buyer can order any amount from the supplier whenever it is convenient. This may be the case where there are many suppliers and they are competing for buyers. However, the supplier would prefer a contract in which the buyer (who has better information about the demand uncertainty) commits in advance how much to purchase over a certain period of time. In return, the supplier may offer a discount to the buyer to make this choice attractive. These type of agreements are reported in practice [1, 20, 18]. With such an agreement, the challenge for the buyer becomes to determine how much to commit to purchase in advance (e.g., in total for the entire horizon or per period) and how much to order in each period. This problem and its variants (such as finite or infinite horizons, with or without fixed costs, total or individual period commitments) have been well motivated and studied in literature [1, 20, 9]. These studies mostly (and naturally) use dynamic programming to determine an optimal policy and in some cases they develop heuristics. Nevertheless, all the previous studies on this topic that we are aware of consider situations where a buyer commits to how much to purchase in advance and decides how much to order in each period consecutively, i.e., ordering decision for the next period is given after this period’s demand realization. In our setting, the buyer needs to decide how much to order for all periods at once and now, before any realization of random demands. There can be some situations where the buyer needs to enter such a contract to secure any orders from a strong supplier.

We discuss two models and a few examples. The first one is a multi-period inventory model for a buyer with a perishable product and backordering. The second one is a multi-period production model for a producer with random yield with high inventory and product shortage costs. The distinct feature of these models from previous ones reported in literature is that all quantity commitment (order and supply amounts) decisions are to be made at once and before the decision horizon starts. To the best of our knowledge, the problems considered in Chapter 5 have not yet been studied.

### 1.2.5 Minimizing a Discrete-Convex Function for Appointment Scheduling

The objective function of the appointment scheduling problem, under a simple sufficient condition, is discretely convex as a function of the integer appointment vector (Chapter 2),
but it is convex and non-smooth when appointment vectors are continuous (Chapter 3). In Appendix A, we investigate whether we can take advantage of both discrete convexity and non-smooth convex optimization methods to solve the appointment scheduling problem. Our purpose is to find a way to combine both sets of methods to minimize the objective function of the appointment scheduling problem more efficiently and practically.

In this Appendix, we compute a subgradient of the objective in polynomial time for any given (real-valued) appointment schedule with independent processing durations by using the subdifferential characterization obtained in Chapter 3. Finding a subgradient in polynomial time is not trivial because the subdifferential formulas include exponentially many terms, and some of the probability computations are complicated. We also extend computation of the expected total cost (in polynomial time) for any (real-valued) appointment vector. These results allow us to use non-smooth convex optimization techniques to find an optimal schedule. To combine the discrete and non-smooth algorithms, a hybrid approach, we develop a special-purpose integer rounding method which takes any fractional solution and rounds it to an integer one with the same or improved objective value. We believe this hybrid approach may perform well in practice.

1.3 Outline of the Thesis

The rest of this thesis, as seen from Section 1.2, is organized as a series of chapters. At the beginning of every chapter, we motivate the problem in discussion and examine the related work. We provide our analysis and results. We conclude each chapter with a summary of the main findings. In addition to the chapters discussed in Section 1.2, we have Chapter 6 which summarizes the thesis contributions and provides a brief discussion of future research directions.
1.4 Bibliography


2 Appointment Scheduling with Discrete Random Durations

We consider the problem of determining an optimal appointment schedule for a given sequence of jobs (e.g., medical procedures) on a single processor (e.g., operating room, examination facility, physician), to minimize the expected total underage and overage costs when each job has a random processing duration given by a joint discrete probability distribution. Simple conditions on the cost rates imply that the objective function is submodular and L-convex. Then there exists an optimal appointment schedule which is integer and can be found in polynomial time. Our model can handle a given due date for the total processing (e.g., end of day for an operating room) after which overtime is incurred, and no-shows and some emergencies.

2.1 Introduction and Motivation

Our research concerns appointment scheduling of jobs on a highly utilized processor when the processing durations are stochastic, and jobs are not available before their appointment dates. We came across this problem in surgery scheduling and in appointment scheduling of oncologist consultations and radiation therapy treatments for cancer patients. There are many other challenging and important real-life applications for this setting including healthcare diagnostic operations (such as CAT scan, MRI) and physician appointments, as well as project scheduling, container vessel and terminal operations, gate and runway scheduling of aircrafts in an airport. For example, in surgery scheduling, patients or surgeries

---

1 A version of this chapter has been submitted for publication. Begen M. A. and Queyranne M. Appointment Scheduling with Discrete Random Durations.
2 A conference version of this chapter appeared in [1].
3 To conform with scheduling terminology, we use the term “date” to denote a point in time. In most applications of appointment scheduling, the appointment “dates” are actually appointment times within the day for which the jobs are being scheduled.
are the jobs, the operating room (OR) and associated resources are the processor, and the surgeon or the hospital is the scheduler. Figure 2.1 shows an example of surgery durations (OR time in minutes) per surgical specialty. As seen from the box plots of Figure 2.1 surgery durations show variability. This data was obtained during an applied research project [28].

Some appointment scheduling applications may have a specific due date for the end of processing, e.g., end of day for an OR, after which additional cost per time unit, e.g., overtime, is incurred. The need for a good schedule is crucial, and savings from such a schedule can be significant. In most cases, an appointment schedule needs to be prepared before any processing starts. It assigns each procedure an allocated duration by specifying the appointment date at which the required personnel and equipment, and the job or patient will be available. However, due to the uncertain processing durations, some jobs may finish earlier, whereas some others may finish later, than the appointment date of the next job. As the appointment dates have to be determined in advance, there are only limited recourse options when the actual duration of a job differs from its planned value. When a procedure finishes earlier than the next procedure’s appointment date, the processor and other resources remain idle until the appointment date of the next job. This results in
resource under-utilization. On the other hand, if a job finishes later than the next job’s appointment date, the next job has to wait for the preceding procedure to complete and will start later than its original appointment date. This results in waiting for the next job and may cause overtime for the processor and resources at the end of the schedule. Therefore, there is an important trade-off between under-utilization, overtime and job waiting times. We are interested in generating an appointment vector\(^4\) that minimizes the expected total cost of resource under-utilization, overtime and job waiting times. Finding such a schedule is more challenging but more valuable and useful when processing durations have more variability. Figure 2.2 shows an instance with 3 jobs \(G, B, R\) to be processed in this order. An appointment schedule \((A_G, A_B, A_R)\) is given. Once the processing starts, due to the random processing durations, some jobs may be early whereas some others may be late as shown in Figure 2.2.

\(\text{an appointment schedule: } A_G, A_B, A_R\)

\(\text{a realization of durations and the completion times}\)

\(\text{W: wait time}\)

\(\text{O: overtime}\)

\(\text{I: idle time}\)

\(\text{Figure 2.2: A Three-Job Instance, and A Realization of the Processing Durations}\)

This problem can be modeled as a multistage stochastic program, but there are significant computational difficulties due to the need for multidimensional numerical integration (see Section 2.2). To our best knowledge, all the analytical studies we are aware of, even the ones with discrete epochs for job arrivals, use continuous processing duration distributions. For a given sequence of jobs, only small instances can be solved to optimality, larger instances require heuristics.

We study a discrete time version of the appointment scheduling problem and establish discrete convexity properties of the objective function. Discrete convex analysis has been advocated by Murota [18] and for recent developments in the topic see [20]. We prove that

\(^4\)We use appointment schedule and appointment vector interchangeably.
the objective function is L-convex under mild assumptions on cost coefficients. L-convex functions, introduced by Murota in [17], play a central role in discrete convexity and our research. Furthermore, we show that there exists an optimal integer appointment schedule minimizing expected total costs. This result is important as it allows us to optimize only over integer appointment schedules without loss of optimality. All these results on the objective function and optimal appointment schedule enable us to develop a polynomial time algorithm, based on discrete convexity [19], that, for a given processing sequence, finds an appointment schedule minimizing the total expected cost. This algorithm invokes a sequence of submodular set function minimizations, for which various algorithms are available, see e.g., [9], [13], [8], [16] and [21].

When processing durations are stochastically independent we evaluate the expected cost for a given processing order and an integer appointment schedule, efficiently both in theory (in polynomial time) and in practice (computations are quite fast as shown in our preliminary computational experiments). Independent processing durations lead to faster algorithms.

Our modeling framework can include a given due date for the end of processing (e.g., end of day for an operating room) after which overtime is incurred, instead of letting the model choose an end date. We also extend our analysis to include no-shows and emergency jobs. The expected benefits of this research effort include reduced job waiting times, reduced overtime and improved capacity utilization.

Our chapter is organized as follows. We start with a literature summary in Section 2.2. Section 2.3 states our assumptions, introduces our notation and formally defines the problem and objective function. Section 2.4 gives some basic properties of the objective function and optimal solutions. In section 2.5 we show the existence of an optimal appointment vector which is integer. Section 2.6 establishes the submodularity and L-convexity of the objective function under a mild condition on cost coefficients. We show that the total expected cost can be minimized efficiently and give the complexity of this minimization in Section 2.7. In this section, we also compute the objective function for any integer appointment vector and determine its complexity when the processing durations are stochastically independent. This independence assumption leads to faster algorithms. We extend our analysis for an objective function with a due date for the end of processing in Section 2.8. Section 2.9 shows how to handle no-shows and some emergency jobs within our framework. Section 2.10
discusses the current work and future work, and it concludes the chapter.

2.2 Related Literature

There are many studies in the last 50 years about appointment scheduling, especially in healthcare. Here, we present the ones that we believe are the most relevant to our research. The use of appointment systems is not limited to service industries but also extends to other areas, such as project management, production and transportation.

Sabria and Daganzo [27] consider scheduling of arrivals of container vessels at a seaport. Weiss [33] recognized that the appointment scheduling problem has a closed form solution when there are only two jobs, and it coincides with the well known newsvendor solution from inventory theory. Robinson et al. [26] extend this result to three jobs by obtaining optimality conditions. Zipkin [34] presents an analysis on the structure of a single-item multi-period inventory system, closely related to the newsvendor problem, by using discrete convexity. Elhafsi [7] studies a production system of multiple stages with stochastic lead-times. The objective is to determine planned leadtimes such that the expected total cost (inventory, tardiness and earliness) is minimized. Bendavid and Golany [2] consider project scheduling with stochastic activity durations. They address the problem of determining for each activity a gate, i.e., a time before which the activity cannot begin, so as to minimize total expected holding and shortage costs, for which they use a heuristic based on the Cross Entropy methodology. Cayirli and Veral [5] review the literature on appointment systems of outpatient scheduling. The authors classify the literature in terms of methodologies and modeling aspects considered, and provide a discussion of performance measures. The authors conclude that the existing literature provides very situation-specific solutions and does not offer generally applicable and portable methodologies for appointment systems design in outpatient scheduling. Another literature review by Cardoen et al. [4] on operating room scheduling evaluates the papers on either the problem setting, such as performance measures, or technical properties such as solution methods.

In a queuing based study, Wang [31] develops a model to find appointment dates of jobs in a single server system to minimize expected customer delay and server completion time with identical jobs and costs, and exponential processing duration distributions. In his numerical studies, the optimal allocated time for each job shows a “dome” structure,
i.e., it increases first and then decreases. In another study, Wang [32] investigates the sequencing problem with the same setting but with distinct exponential distributions. He conjectures that sequencing with increasing variance is optimal. Bosch et al. [3] present a model with i.i.d. Erlang processing durations and identical cost coefficients. In their model, customers can arrive only at discrete potential arrival epochs, which are equally spaced, and the decision variable is the number of customers to be scheduled at each potential arrival epoch. In a related paper, Kaandorp and Koole [14] study outpatient appointment scheduling with exponential processing durations and no-shows. They take advantage of the exponential distribution in their computations and define a neighborhood structure and an exact search method. However for large instances, they develop a heuristic due to high computation times of their search method.

Another important stream of appointment scheduling research is based on stochastic programming. Denton and Gupta [6] develop a two-stage stochastic linear program to determine optimal appointment dates for a given surgery sequence and due date for the end of processing horizon. The authors use general, i.i.d. and continuous processing durations, and identical server idling cost coefficients for all jobs. They infer from stochastic programming results that their model is a convex minimization problem, and they develop an algorithm with sequential bounding for solving small sized instances. They develop heuristics to solve larger instances. In a related study, Robinson and Chen [25] develop a stochastic linear program for finding appointment dates for a fixed sequence of surgeries and propose a Monte-Carlo based solution method. Due to the high computational requirements of Monte-Carlo integration, they develop heuristics in which they use the “dome” structure of the optimal policy as reported in Wang [31].

Appointment scheduling can be thought of as an operational level of capacity planning problem since it concerns with scheduling of jobs/patients available on the day of processing/service [28], [22] and [29]. Researchers also study the problem of scheduling patients in advance of the service date. In this stream of research, e.g., [22], [29], [10], [11] and the references therein, arrivals are random but processing durations are deterministic and the main decision is how to allocate available capacity to incoming demand. Different objectives are considered such as revenue maximization [11] or cost minimization to achieve target waiting times [22]. Luzon et al. [15] use a fluid approximation to minimize average waiting time.
We finally would like to point out the similarities between appointment scheduling and single machine scheduling, see e.g., [24] for machine scheduling. Unlike machine scheduling, in appointment scheduling a sequence is given and the release dates are the decision variables. Furthermore, the objective function of the appointment scheduling problem is quite different than the objective functions of classical machine scheduling problems. Processing durations are usually deterministic in machine scheduling problems but random processing durations are also studied in literature, see e.g., [23] and [24].

In this chapter we develop a sufficiently generic and portable framework to solve the appointment scheduling problem efficiently.

### 2.3 Assumptions and Notation

There are $n + 1$ jobs that need to be sequentially processed on a single processor. The processing sequence is given. An appointment schedule needs to be prepared before any processing can start. Jobs will not be available before their appointment dates. When a job finishes earlier than the next job’s appointment date, the system experiences some cost due to under-utilization. We refer to this cost as the *underage* cost. On the other hand, if a job finishes later than the next job’s appointment date, the system experiences *overage* cost due to the overtime of the current job and the waiting of the next job.

The processing durations are given by their joint discrete distribution. In Section 2.7, we will show that assuming independent discrete processing durations lead to faster algorithms. We assume that this joint distribution is known. Complete information of distributions is reasonable in most settings, but we relax this assumption in Chapter 3. Our next assumption is a natural one: all cost coefficients and processing durations are non-negative and bounded. A key assumption in this work is that processing durations are integer valued\(^5\). Although we obtain some of our results without this assumption, it is important for our main results.

We assume job 1 starts on-time, i.e., the start time for the first job is zero, and there are $n$ real jobs. The $(n + 1)^{th}$ job is a dummy job with a processing duration of 0. The appointment time for the $(n + 1)^{th}$ job is the total time available for the $n$ real jobs. We use the dummy job to compute the overage or underage cost of the $n^{th}$ job.

Let $\{1, 2, 3, ..., n, n+1\}$ denote the set of jobs. We denote the random processing duration

---

\(^5\)We can restrict ourselves to integer appointment schedules without loss of optimality by Theorem 2.5.10.
of job $i$ by $p_i$ and the random vector\footnote{We write all vectors as row vectors.} of processing durations by $p = (p_1, p_2, \ldots, p_n, 0)$. Let $\underline{p}_i$ and $\overline{p}_i$ denote the minimum and maximum possible value of processing duration $p_i$, respectively. The maximum of these $\overline{p}_i$’s is $\overline{p}_{\text{max}} = \max(\overline{p}_1, \ldots, \overline{p}_n)$. The 	extit{underage cost rate} $u_i$ of job $i$ is the unit cost (per time unit) incurred when job $i$ is completed at a date $C_i$ before the appointment date $A_{i+1}$ of the next job $i+1$. The 	extit{overage cost rate} $o_i$ of job $i$ is the unit cost incurred when job $i$ is completed at a date $C_i$ after the appointment date $A_i + 1$.

Thus the total cost due to job $i$ completing at date $C_i$ is $u_i(A_i + 1 - C_i) + o_i(C_i - A_{i+1})$ where $(x)^+ = \max(0, x)$ is the positive part of real number $x$. We define $u = (u_1, u_2, \ldots, u_n)$ and $o = (o_1, o_2, \ldots, o_n)$. We denote unit vectors in $\mathbb{R}^{n+1}$ as $\mathbf{1}_i$ where the $i^{th}$ component is 1 and all other components are 0.

The underage cost may be interpreted as the idling cost and/or opportunity cost of the resources, whereas the overage cost may be thought as the waiting cost of the next job and/or the overtime of the current job. The overage cost of the last job may include the overtime cost for the whole facility at the end of the schedule after a specified due date.

Next we introduce our decision variables. Let $a_i$ be the allocated duration and $A_i$ the appointment date for job $i$. Then we have $A_1 = 0$ and $A_{i+1} = A_i + a_i$ for $i = 1, \ldots, n$. Thus we may equivalently use the 	extit{allocated duration vector} $a = (a_1, a_2, \ldots, a_{n-1}, a_n)$ or the 	extit{appointment vector} $A = (A_1, A_2, \ldots, A_n, A_{n+1})$ (with $A_1 = 0$) as our decision variables; we choose to work with the appointment vector $A$. We introduce additional variables which help define and compute the objective function. Let $S_i$ be the start date and $C_i$ the completion date of job $i$. Since job 1 starts on-time we have $S_1 = 0$ and $C_1 = p_1$. The other start times and completion times are determined as follows: $S_i = \max\{A_i, C_{i-1}\}$ and $C_i = S_i + p_i$ for $2 \leq i \leq n + 1$. Note that the dates $S_i$ and $C_i$ are random variables which depend on the appointment vector $A$.

Let $F(A|p)$ be the total cost of appointment vector $A$ given processing duration vector $p$:

$$F(A|p) = \sum_{i=1}^{n} (o_i(C_i - A_{i+1})^+ + u_i(A_{i+1} - C_i)^+). \quad (2.1)$$

The objective to be minimized is the expected total cost $F(A) = E_p[F(A|p)]$ where the expectation is taken with respect to random processing duration vector $p$. We simplify notations by defining the lateness $L_i = C_i - A_{i+1}$ of job $i$, its tardiness $T_i = (L_i)^+$, and its
earliness \( E_i = (-L_i)^+ \). The objective \( F(A) \) can now be written as

\[
F(A) = E_p \left[ \sum_{i=1}^{n} (o_i T_i + u_i E_i) \right] = \sum_{i=1}^{n} (o_i E_p T_i + u_i E_p E_i).
\]

### 2.4 Basic Properties

We start by making an observation about the completion times and expressing the objective function in a different form that is useful for deriving some of our later results. Since \( C_i = S_i + p_i = \max\{A_i, C_{i-1}\} + p_i \), the completion time of job \( i \) may be seen as the length of the longest (or critical) path from some job \( j \) \((j \leq i)\) to job \( i+1 \) in a corresponding “project network” (Pinedo [24]), namely:

**Lemma 2.4.1. (Critical Path)** For all jobs \( i = 1, \ldots, n \),

\[
C_i = \max_{j \leq i} \{ A_j + \sum_{k=j}^{i} p_k \}
\]

\[
F(A|p) = \sum_{i=1}^{n} \left( o_i \left( \max_{j \leq i} \{ A_j + \sum_{k=j}^{i} p_k \} - A_{i+1} \right)^+ + u_i \left( A_{i+1} - \max_{j \leq i} \{ A_j + \sum_{k=j}^{i} p_k \} \right)^+ \right).
\]

**Proof.** The claim holds trivially for \( i = 1 \). By induction let the claim be true for \( i = m \), i.e., \( C_m = \max_{j \leq m} \{ A_j + \sum_{k=j}^{m} p_k \} \). Then

\[
C_{m+1} = S_{m+1} + p_{m+1} = \max\{A_{m+1}, C_m\} + p_{m+1} \quad \text{by definition}
\]

\[
= \max \left\{ A_{m+1}, \max_{j \leq m} \left\{ A_j + \sum_{k=j}^{m} p_k \right\} \right\} + p_{m+1} \quad \text{by inductive assumption}
\]

\[
= \max \left\{ A_{m+1} + p_{m+1}, \max_{j \leq m} \left\{ A_j + \sum_{k=j}^{m+1} p_k \right\} \right\} = \max_{j \leq m+1} \left\{ A_j + \sum_{k=j}^{m+1} p_k \right\}.
\]

The expression for \( F(A|p) \) follows. \( \square \)

The next result is not only important on its own but also crucial for the existence of an optimal solution.

**Lemma 2.4.2. (Continuity)** Functions \( F(.|p) \) and \( F(.) \) are continuous.

**Proof.** By expression Eq(3.1), \( F(.|p) \) is a weighted sum of piecewise linear continuous functions of \( A \), hence is itself piecewise linear continuous. Since we have a finite sample space, the expectation \( F(.) = E_p F(.|p) \) is also continuous. \( \square \)
We next establish the existence of an optimal solution. The proof follows from the
fact that our objective function is continuous (by Lemma 2.4.2), and we can restrict
the appointment vector to a compact set without loss of optimality. Let \( \mathbf{A} = (A_1, \ldots, A_{n+1}) \)
and \( \overline{\mathbf{A}} = (\overline{A}_1, \ldots, \overline{A}_{n+1}) \) where \( A_1 = \overline{A}_1 = 0 \), \( A_i = \sum_{j<i} p_j \) and \( \overline{A}_i = \sum_{j<i} \overline{p}_j \) for \( i = 2, \ldots, n+1 \). We define the compact set \( \mathcal{K} \) as the cartesian product of the intervals \([A_i, \overline{A}_i]\),
\( \mathcal{K} = \prod_{i=1}^{n+1} [A_i, \overline{A}_i] = [\mathbf{A}, \overline{\mathbf{A}}] \subseteq \mathbb{R}^{n+1} \).

**Lemma 2.4.3. (Existence of an Optimal Vector)** There exists an appointment vector \( \mathbf{A}^* \in \mathcal{K} \) such that \( F(\mathbf{A}^*) \leq F(\mathbf{A}) \) for any appointment vector \( \mathbf{A} \).

**Proof.** We show that we can restrict, without loss of optimality, the appointment vector \( \mathbf{A} \) to the compact set \( \mathcal{K} = [\mathbf{A}, \overline{\mathbf{A}}] \) and recall that job 1 starts at time zero, i.e., \( A_1 = 0 = A_1 = \overline{A}_1 \).
Consider any appointment vector \( \mathbf{A} \not\in \mathcal{K} \) with \( A_1 = 0 \).

If \( \mathbf{A} \not\geq \mathbf{A} \) then define the appointment vector \( \mathbf{A}' = \mathbf{A} \vee \mathbf{A} \) with component \( A'_i = \max\{A_i, A_j\} \). For any realization \( \mathbf{p} \) of the processing durations, the completion times \( C'_i \) in the resulting schedule satisfy \( C'_i = C_i \geq A_{i+1} \). (Indeed, \( C'_1 = p_1 = C_1 \geq A_2 \) and, by induction \( C'_i = \max\{A'_i, C_{i-1}\} + p_i = \max\{A_i, A_{i-1}\} + p_i = \max\{A_i, C_{i-1}\} + p_i = C_i \geq A_{i+1} \). Then the resulting tardiness and earliness become: if \( A_{i+1} \geq A_{i+1} \) then \( T'_i = (C'_i - A'_{i+1}^+) = (C_i - A_{i+1})^+ = T_i \) and \( E'_i = (A'_{i+1} - C'_i)^+ = (A_{i+1} - C_i)^+ = E_i \); and, if \( A_{i+1} < A_{i+1} \) then \( T'_i = (C'_i - A'_{i+1}^+) = (C_i - A_{i+1})^+ \leq (C_i - A_{i+1})^+ = T_i \) and \( 0 \leq E_i = (A_{i+1} - C_i)^+ \leq (A_{i+1} - C_i)^+ = E'_i = 0 \) (so \( E'_i = E_i = 0 \)). Since all \( u_i, o_i \geq 0 \), it follows from Eq(3.1) that \( F(\mathbf{A}'|\mathbf{p}) \leq F(\mathbf{A}|\mathbf{p}) \) and thus \( F(\mathbf{A}') \leq F(\mathbf{A}) \). We have shown that for every \( \mathbf{A} \) there exists \( \mathbf{A}' \geq \mathbf{A} \) with \( F(\mathbf{A}') \leq F(\mathbf{A}) \).

Now, for any vector \( \mathbf{A} \in \mathbb{R}^{n+1} \) satisfying \( \mathbf{A} \geq \mathbf{A} \), \( A_1 = 0 \) and \( \mathbf{A} \not\in \mathcal{K} \), let \( i(\mathbf{A}) \) denote the smallest index such that \( A_i > \overline{A}_i \). Let \( \mathbf{A} \in \mathbb{R}^{n+1} \) be a vector with largest \( i(\mathbf{A}) \) value satisfying \( \mathbf{A} \geq \mathbf{A} \), \( A_1 = 0 \) and \( \mathbf{A} \not\in \mathcal{K} \). We claim that there exists \( \mathbf{A}' \) satisfying \( \mathbf{A}' \geq \mathbf{A} \), \( A'_1 = 0 \), \( F(\mathbf{A}') \leq F(\mathbf{A}) \), and either \( \mathbf{A}' \in \mathcal{K} \) or \( i(\mathbf{A}') > i(\mathbf{A}) \). Then after at most \( n \) such changes we obtain \( \mathbf{A}'' \in \mathcal{K} \) satisfying \( F(\mathbf{A}'') \leq F(\mathbf{A}) \), which is what we wanted to show.

We now prove the claim.

Let \( i = i(\mathbf{A}) \), \( \varepsilon = A_i - \overline{A}_i > 0 \), and define \( \mathbf{A}' \) with \( A'_j = A_j \) for all \( j \leq i - 1 \) and \( A'_j = A_j - \varepsilon \) for all \( j \geq i \), so \( A'_i = \overline{A}_i \). For every realization \( \mathbf{p} \) of the processing durations, the completion time \( C'_j \) in the resulting schedule satisfy \( C'_j = C_j \) for all \( j \leq i - 1 \). Note that for all \( j \leq i - 1 \), \( A_j \leq \overline{A}_j \) implies \( C_j \leq \overline{A}_{j+1} \). Therefore \( C_i = A_i + p_i \) and
\[ C'_i = A'_i + p_i = A'_i + C_i - A_i = C_i - \varepsilon. \] It follows that \( C'_j = C_j - \varepsilon \) for all \( j \geq i \). As a result, \( E'_j = E_j \) and \( T'_j = T_j \) for all \( j \neq i - 1 \), and \( E'_{i-1} = E_{i-1} - \varepsilon \), \( T'_{i-1} = T_{i-1} = 0 \). Since \( \varepsilon > 0 \) and \( u_{i-1} \geq 0 \), \( F(A'p) \leq F(Ap) \) and thus \( F(A') \leq F(A) \). Since \( A'_j = A_j \leq A_j \) for all \( j \leq i - 1 \) and \( A'_i = A_i \), then either \( A' \in K \) or \( i(A') \geq i + 1 = i(A) + 1 \), establishing the claim. This shows that for any \( A \notin K \) there exists a vector \( A'' \in K \) with \( F(A'') \leq F(A) \).

As a result, since \( F \) is continuous, its minimum on compact set \( K \) is attained and is therefore the global minimum.

The next lemma gives bounds on the difference between any two consecutive components of an optimal appointment vector, and from this we obtain a useful and intuitive result in Lemma 2.4.5.

**Lemma 2.4.4.** There exists an optimal appointment schedule \( A^* \in K \) satisfying \( p_i \leq A^*_{i+1} - A^*_i \leq \sum_{j<i} p_j - \sum_{j<i} p_j \) for all \( i = 1, \ldots, n \).

**Proof.** By Lemma 2.4.3, we immediately obtain \( p_1 \leq A^*_2 - A^*_1 \leq \bar{p}_1 \) and \( A^*_{i+1} - A^*_i \leq \sum_{j<i} p_j - \sum_{j<i} p_j \) for all \( i = 2, \ldots, n \). Next, we show that \( p_i \leq A^*_{i+1} - A^*_i \) holds for all \( i = 2, \ldots, n \). By contradiction, suppose \( p_i + A^*_k > A^*_i + A^*_{k+1} \) for some \( k = 2, \ldots, n \) then job \( k \) is late at least \( (p_k + A^*_k - A^*_{k+1}) \) time units, so increasing \( A^*_{k+1} \) to \( p_k + A^*_k \) will improve the objective function by \( o_k (p_k + A^*_k - A^*_{k+1}) \geq 0 \). Therefore we must have \( p_i \leq A^*_{i+1} - A^*_i \) for all \( i = 2, \ldots, n \). \( \Box \)

**Lemma 2.4.5. (Non-Decreasing Appointment Dates)** There exists an optimal appointment vector \( A^* \in K \) with non-decreasing components, i.e., \( A^*_i \leq A^*_i+1 \) for all \( i = 1, \ldots, n \).

**Proof.** By Lemma 2.4.4, \( A^*_{i+1} - A^*_i \geq p_i \geq 0 \) (1 \( i \leq n \)). \( \Box \)

### 2.5 Optimality of an Integer Appointment Vector

The existence of an optimal appointment vector which is integer is crucial. It implies that we can restrict attention to integer appointment vectors without loss of optimality. We establish this result in the Appointment Vector Integrality Theorem 2.5.10. Its proof is surprisingly non-trivial.

Let \( A^* \) be any non-integer appointment vector and \( A^*_j \) the first non-integer component of \( A^* \). Knowing all the jobs which have the same fractional part as \( A^*_j \) is crucial, so we
define \( J \) to be the set of all jobs \( j \geq f \) such that \( A'_j - A''_j \) is integer. Let \( \mathbb{Z} \) denote the set of integers, and \( \lfloor x \rfloor = \sup\{n \in \mathbb{Z} : n \leq x\} \) and \( \lceil x \rceil = \inf\{n \in \mathbb{Z} : n \geq x\} \) for \( x \in \mathbb{R} \). Let \( \varphi(x) \) be the distance to the nearest integer for \( x \in \mathbb{R} \), i.e., \( \varphi(x) = \min(x - \lfloor x \rfloor, \lceil x \rceil - x) \). Let \( \Delta \) be a strictly positive scalar satisfying \( 0 < \Delta < \frac{1}{2} \min(\Delta_1, \Delta_2) \) where \( \Delta_1 = \frac{1}{4} \min\{\varphi(\lfloor A'_j - A''_k \rfloor) : j \in J, k \notin J\} > 0 \) and \( \Delta_2 = \frac{1}{4} \min\{\varphi(\lfloor A'_j - A''_k \rfloor) : j \notin J, k \in J, A_j - A_k \notin \mathbb{Z}\} > 0 \). We use \( \Delta \) to construct two new appointment schedules \( A' \) and \( A'' \) from \( A^* \): let \( A'_j = A''_j - \Delta \) if \( j \in J \), and \( A'_j = A''_j \) otherwise; similarly, let \( A''_j = A'_j + \Delta \) if \( j \in J \), and \( A''_j = A'_j \) otherwise.

For any realization of the processing duration vector \( p \), denote the completion times of job \( j \) as \( C'_j, C''_j \) in schedules \( A^*, A' \) and \( A'', \) respectively.

One of the main ideas in proving the Appointment Vector Integrality Theorem 2.5.10 is that \( \Delta \) is small enough so that there is “no event change” when we move from schedule \( A^* \) to schedules \( A' \) and \( A'' \). When there is no event change, we show in Lemma 2.5.9 that our objective function changes linearly between schedules \( A' \) and \( A'' \). To make the no event change concept precise, we define the following. Job \( i \) (\( 1 < i \leq n + 1 \)) is late if \( C^*_{i-1} > A'_i \) (strictly positive tardiness), early if \( C^*_{i-1} < A'_i \) (strictly positive earliness), just-on-time if \( C^*_{i-1} = A'_i \), and on-time if \( C^*_{i-1} \leq A'_i \). Then no event change means that if any job is late, early or just-on-time, respectively, in schedule \( A^* \) then it is also late, early or just-on-time, respectively, in both schedules \( A' \) and \( A'' \).

We consider all possible realizations \( r \) of the random processing duration vector \( p \), so \( r_i \) is the corresponding realization of the processing duration \( p_i \). We start by establishing relationships between the completion times in the schedules \( A' \) and \( A^* \), and \( A'' \) and \( A^* \).

**Lemma 2.5.1.** For every realization of the processing durations and every \( j = 1, \ldots, n+1 \), \( C'_j + \Delta \geq C''_j \geq C'_j \geq C''_j \geq C^*_j - \Delta \).

**Proof.** Let \( 1 \leq j \leq n + 1 \) and let \( r \) be a realization of \( p \). Then \( A'_j - \Delta \leq A'_j \leq A''_j \leq A'_j + \Delta \) by definition of \( A' \) and \( A'' \). By the Critical Path Lemma 2.4.1, \( C'_j = \max_{k \leq j}\{A'_k + \sum_{i=k}^j r_i\} \), \( C''_j = \max_{k \leq j}\{A''_k + \sum_{i=k}^j r_i\} \) and \( C''_j = \max_{k \leq j}\{A''_k + \sum_{i=k}^j r_i\} \).

Hence, \( A'_j \leq A''_j \) implies that \( C'_j \leq C''_j \). On the other hand, \( A^*_j - \Delta \leq A'_j \) implies that \( C^*_j - \Delta = \max_{k \leq j}\{A^*_k - \Delta + \sum_{i=k}^j r_i\} \leq \max_{k \leq j}\{A'_k + \sum_{i=k}^j r_i\} = C'_j \) so \( C^*_j - \Delta \leq C'_j \). Similarly \( A_2^* + \Delta \geq A^*_j \) implies that \( C^*_j + \Delta = \max_{h \leq j}\{A^*_k + \Delta + \sum_{i=k}^j r_i\} \geq \max_{k \leq j}\{A'_k + \sum_{i=k}^j r_i\} = C'_j \) so \( C^*_j + \Delta \geq C''_j \). The result follows. \( \square \)

The next two results are about late and early jobs. Lemma 2.5.2 below implies that if
job $k$ is late (resp., early) then its tardiness (resp., earliness) is strictly greater then $2\Delta$.

Lemma 2.5.3 implies that if job $k$ is late (resp., early) in schedule $A^*$ then it is also late (resp., early) in $A'$ and $A''$.

**Lemma 2.5.2.** For every realization of the processing durations and every $k = 2, \ldots, n+1$, if $|C_{k-1}^* - A_k^*| > 0$ then $|C_{k-1}^* - A_k^*| > 2\Delta$.

Proof. Let $r$ be a realization of $p$. Let $t$ be the last on-time job before $k$ ($1 \leq t < k$) so $C_{k-1}^* = A_t^* + \sum_{i=t}^{k-1} r_i$. Note that $t$ exists and is well defined since job 1 is always on-time, i.e., $A_1^* = 0$. We consider two cases: $(A_k^* - A_t^*) \in \mathbb{Z}$ or $(A_k^* - A_t^*) \notin \mathbb{Z}$. If $(A_k^* - A_t^*) \in \mathbb{Z}$ then $0 < |C_{k-1}^* - A_k^*| = |A_t^* + \sum_{i=t}^{k-1} r_i - A_k^*|$ but since the $r_i$'s and $A_k^* - A_t^*$ are integer, $|A_t^* + \sum_{i=t}^{k-1} r_i - A_k^*|$ is a positive integer, hence $|C_{k-1}^* - A_k^*| = |A_t^* + \sum_{i=t}^{k-1} r_i - A_k^*| \geq 1 > 2\Delta$.

By Lemma 2.5.2, $|C_{k-1}^* - A_k^*| > 2\Delta$, and this implies $|A_k^* - A_t^*| - (A_k^* - A_t^*) > 2\Delta$ and $(A_k^* - A_t^*) - (A_k^* - A_t^*) > 2\Delta$. Since $\sum_{i=t}^{k-1} r_i$ is integer, we must have either $\sum_{i=t}^{k-1} r_i \leq |A_k^* - A_t^*|$ or $\sum_{i=t}^{k-1} r_i \geq |A_k^* - A_t^*|$. Therefore $|C_{k-1}^* - A_k^*| = |A_t^* + \sum_{i=t}^{k-1} r_i - A_k^*| > 2\Delta$. □

**Lemma 2.5.3.** For every realization of the processing duration and every $k = 2, \ldots, n+1$, if $C_{k-1}^* > A_k^*$ then $C_{k-1}^* > A_k^*$ and $C_{k-1}'' > A_k''$, and if $C_{k-1}^* < A_k^*$ then $C_{k-1}'' < A_k''$.\n
Proof. By Lemma 2.5.2, $C_{k-1}^* > A_k^*$ implies $C_{k-1}^* - A_k^* > 2\Delta$. Note that $A_k^* - \Delta \leq A_k' \leq A_k^* \leq C_{k-1}'' \leq C_{k-1}^* - \Delta$ by definition, and $C_{k-1}^* + \Delta \geq C_{k-1}'' \geq C_{k-1}^* \geq C_{k-1}^* - \Delta$ by Lemma 2.5.1. Then $C_{k-1}^* - A_k^* > 2\Delta$ implies $C_{k-1}'' - A_k' \geq C_{k-1}'' - A_k^* > \Delta$ and $C_{k-1}'' - A_k'' \geq C_{k-1}'' - A_k^* - \Delta > \Delta$. Similarly, by Lemma 2.5.2, $C_{k-1}'' < A_k^*$ implies $A_k' - C_{k-1}'' > 2\Delta$. Note that $A_k' - \Delta \leq A_k' \leq A_k' \leq A_k' + \Delta$ by definition, and $C_{k-1}^* + \Delta \geq C_{k-1}'' \geq C_{k-1}'' \geq C_{k-1}'' - \Delta$ by Lemma 2.5.1. Then $A_k' - C_{k-1}'' > 2\Delta$ implies $A_k' - C_{k-1}'' \geq A_k' - C_{k-1}'' - \Delta > \Delta$ and $A_k' - C_{k-1}'' \geq A_k' - C_{k-1}'' - \Delta > \Delta$. The result follows. □

Just-on-time jobs require more care, and we need further definitions and results before we can establish similar results as Lemmata 2.5.2 and 2.5.3. Let a **block** $B[t, k]$ be a sequence of consecutive jobs, $[t, t+1, \ldots, k]$ ($1 \leq t < k \leq n+1$) such that either $t = 1$ or job $t$ is early, i.e., $C_{t-1}^* < S_t^* = A_t^*$; no other job in the block is early, i.e., $S_{j+1}^* = C_j^* \geq A_{j+1}^*$ for $j = t+1, \ldots, k$; and job $k$ is just-on-time, i.e., $C_{k-1}^* = A_k^*$. Let $K = \{ i : t < i \leq k \; \text{and} \; C_{i-1}^* = A_i^* \}$ denote the set of just-on-time jobs in the block $B[t, k]$. So we have $S_t^* = A_t^*$ and $S_j^* = A_j^* = C_{j-1}^*$ for
all $j \in K$. Our next result, Lemma 2.5.4, implies that the first (job $t$) and all just-on-time jobs in a block (i.e., elements of $K$) are either all in $J$ or all outside $J$.

**Lemma 2.5.4.** If $B[t, k]$ is a block then either $\{t\} \cup K \subseteq J$ or $\{t\} \cup K \subseteq B[t, k] \setminus J$.

*Proof.* Let $j \in K$. We have $C_k^{*} = A_k^* + \sum_{i=1}^{k-1} r_i$ since $t$ is on-time, and there is no idle time between $t$ and $j$. We obtain $0 = C_k^{*} - A_k^* = A_k^* + \sum_{i=1}^{k-1} r_i - A_k^*$. Since $\sum_{i=1}^{k-1} r_i$ is integer, $A_k^* - A_k^*$ must be integer. This implies that if $j \in J$ then $t \in J$, and if $j \not\in J$ then $t \not\in J$. \hfill $\square$

Lemma 2.5.5 will be used to prove Lemmata 2.5.6 and 2.5.7.

**Lemma 2.5.5.** Let $k \in \{2, \ldots, n+1\}$ be such that $A_k^* \not\in \mathbb{Z}$. Then for every realization of the processing durations such that $C_{k-1}^{*} = A_k^*$ there is an early job $j < k$.

*Proof.* Let $r$ be a realization of $p$. Seeking a contradiction, assume there is no early job before job $k$. Then $C_{k-1}^{*} = A_k^* + \sum_{i=1}^{k-1} r_i = A_k^*$. This implies $A_k^* \in \mathbb{Z}$ (since $A_1^* = 0$ and $\sum_{i=1}^{k-1} r_i$ are integer), a contradiction. \hfill $\square$

In Lemmata 2.5.6 and 2.5.7 below we prove that no event change occurs for any just-on-time job. Therefore Lemma 2.5.8 states that no event change occurs for any job.

**Lemma 2.5.6.** Let $k \in \{2, \ldots, n+1\}$. For every realization of the processing durations such that $C_{k-1}^{*} = A_k^*$, if there exists an early job $j < k$ then $C_{k-1}^{*} = A_k^*$ and $C_{k-1}'' = A_k''$.

*Proof.* Let $r$ be a realization of $p$. Let $t$ be the last early job before $k$, so $B[t, k]$ is a block. As explained above, let $K = \{i : t < i \leq k$ and $C_i^{*} = A_i^*\}$ be the set of just-on-time jobs between $t$ and $k$. By Lemma 2.5.4, either (i) $\{t\} \cup K \subseteq J$ or (ii) $\{t\} \cup K \subseteq B[t, k] \setminus J$.

**Case (i) $\{t\} \cup K \subseteq J$**

First, by induction we show that $C_j' = C_j^* - \Delta$ for all $j \in B[t, k]$. Indeed, $C_{i-1}' \leq C_{i-1}^* < C_i^* - 2\Delta < A_i' \ (by \ \text{Lemmata} \ 2.5.1 \ \text{and} \ 2.5.2)$ so $S_i' = A_i' = A_i^* - \Delta$ and $C_i' = A_i^* - \Delta + r_i = C_i^* - \Delta$. Consider $t < j \in B[t, k]$. By inductive assumption, $C_{j-1}' = C_{j-1}^* - \Delta$. If $j \in K$ then $j \in J$ and $A_j' = A_j^* - \Delta$, so $S_j' = \max\{C_{j-1}' , A_j'\} = \max\{C_{j-1}^* - A_j^*\} - \Delta = C_{j-1}^* - \Delta$. Otherwise, $j \not\in K$, i.e., $j$ is late, then by Lemma 2.5.2 $C_{j-1}' > A_j'^* + 2\Delta$. So $A_j' \leq A_j'^* < C_{j-1}^* - 2\Delta = C_{j-1}' - \Delta$ and hence $S_j' = C_{j-1}' = C_{j-1}^* - \Delta$. In both cases, $C_j' = S_j' + r_j = C_{j-1}^* - \Delta + r_j = \max\{C_{j-1}^* , A_j^*\} + r_j - \Delta = C_j^* - \Delta$, completing our
inductive proof. This implies that $C'_{k-1} = C^*_{k-1} - \Delta = A^*_k - \Delta = A'_k$ since $k < K \subseteq J$ so $C'_{k-1} = A'_k$ as claimed.

Similarly, by induction we show that $C''_j = C^*_j + \Delta$ for all $j \in B[t, k]$. Indeed, $C''_{t-1} \leq C^*_{t-1} + \Delta < A^*_t < A^*_t + \Delta = A''_t$ (by Lemmata 2.5.1 and 2.5.2) so $S''_t = A''_t = A^*_t + \Delta$ and $C''_t = A^*_t + \Delta + r_t = C^*_t + \Delta$. Consider $t < j \in B[t, k]$. By inductive assumption, $C''_{j-1} = C^*_{j-1} + \Delta$. If $j \in K$ then $j \in J$ and $A''_j = A^*_j + \Delta$, so $S''_j = \max\{C''_{j-1}, A''_j\} = \max\{C^*_{j-1}, A^*_j\} + \Delta = C^*_{j-1} + \Delta$. Otherwise, $j \not\in K$, i.e., $j$ is late, then by Lemma 2.5.2 $C^*_{j-1} > A^*_j + 2\Delta$. So $A''_j \leq A^*_j + \Delta < C^*_{j-1} - \Delta = C''_{j-1} - 2\Delta$ and hence $S''_j = C''_{j-1} = C^*_{j-1} + \Delta$, completing our inductive proof. In both cases, $C''_j = S''_j + r_j = C^*_{j-1} + \Delta + r_j = \max\{C^*_{j-1}, A^*_j\} + r_j + \Delta = C^*_j + \Delta$. This implies that $C''_{k-1} = C^*_{k-1} + \Delta = A^*_k + \Delta = A'_k$ since $k \in K \subseteq J$ so $C''_{k-1} = A'_k$ as claimed.

Case (ii) \{t\} \cup K \subseteq B[t, k] \setminus J

First, by induction we show that $C'_j = C^*_j$ for all $j \in B[t, k]$. Indeed, $C'_{t-1} \leq C^*_{t-1} < A^*_t - 2\Delta < A^*_t = A'_t$ (by Lemmata 2.5.1 and 2.5.2) so $S'_t = A'_t = A^*_t + r_t = C^*_t$. Consider $t < j \in B[t, k]$. By inductive assumption, $C'_j = C^*_j$. If $j \in K$ then $j \not\in J$ and $A'_j = A^*_j$, so $S'_j = \max\{C'_{j-1}, A'_j\} = \max\{C^*_{j-1}, A^*_j\} = C^*_{j-1}$. Otherwise, $j \not\in K$, i.e., $j$ is late, then by Lemma 2.5.2 $C^*_j > A^*_j + 2\Delta$. So $A'_j \leq A^*_j < C^*_j - 2\Delta = C''_{j-1} - 2\Delta$ and hence $S'_j = C''_{j-1} = C^*_{j-1}$. In both cases, $C'_j = S'_j + r_j = C^*_j + r_j = \max\{C^*_{j-1}, A^*_j\} + r_j = C''_j$, completing our inductive proof. This implies that $C'_{k-1} = C^*_{k-1} = A'_k = A'_k$ since $k \in K$ and $k \not\in J$, so $C'_{k-1} = A'_k$ as claimed.

Similarly, by induction we show that $C''_j = C^*_j$ for all $j \in B[t, k]$. Indeed, $C''_{t-1} \leq C^*_{t-1} + \Delta < A^*_t = A''_t$ (by Lemmata 2.5.1 and 2.5.2) so $S''_t = A''_t = A^*_t + r_t = C^*_t$. Consider $t < j \in B[t, k]$. By inductive assumption, $C''_j = C^*_j$. If $j \in K$ then $j \not\in J$ and $A''_j = A^*_j$, so $S''_j = \max\{C''_{j-1}, A''_j\} = \max\{C^*_{j-1}, A^*_j\} = C^*_j$. Otherwise, $j \not\in K$, i.e., $j$ is late, then by Lemma 2.5.2 $C^*_j > A^*_j + 2\Delta$. So $A''_j \leq A^*_j + \Delta < C^*_j - \Delta = C''_{j-1} - \Delta$ and hence $S''_j = C''_{j-1} = C^*_{j-1}$, completing our inductive proof. In both cases, $C''_j = S''_j + r_j = C''_{j-1} + r_j = \max\{C''_{j-1}, A''_j\} + r_j = C''_j$. This implies that $C''_{k-1} = C^*_{k-1} = A'_k = A'_k$ since $k \in K$ and $k \not\in K$, so $C''_{k-1} = A'_k$ as claimed.

\begin{lemma}
Let $k \in \{2, \ldots, n + 1\}$. For every realization of the processing durations such that $C^*_{k-1} = A^*_k$ we have $C'_{k-1} = A'_k$ and $C''_{k-1} = A''_k$.
\end{lemma}
Proof. If there is an early job before \( k \) then the result follows from Lemma 2.5.6. Otherwise, \( B[1, k] \) is a block. Therefore \( C_{k-1}^* = A_1^* + \sum_{i=1}^{k-1} r_i = A_k^* \). Furthermore, \( A_k^* \in \mathbb{Z} \) by Lemma 2.5.5 so \( k \not\in J \). Therefore \( \{1\} \cup K \subseteq B[1, k] \setminus J \) by Lemma 2.5.4, and hence the result follows from Lemma 2.5.6.

\[
\]

Our next result establishes that no event change occurs for any job and directly follows from Lemmata 2.5.3 and 2.5.7. We define the sign of a real number \( x \) as \( \text{sign}(x) = 1 \) if \( x > 0 \); \( 0 \) if \( x = 0 \); and \( -1 \) if \( x < 0 \).

**Lemma 2.5.8.** For every job \( j = 2, \ldots, n + 1 \) and every realization of the processing durations, \( \text{sign}(C_{j-1}^* - A_j^*) = \text{sign}(C_{j-1}'' - A_j'') = \text{sign}(C_{j-1}^* - A_j^*) \).

Lemma 2.5.9 below gives a consequence on the objective function of this no event change result.

**Lemma 2.5.9.** \( F \) changes linearly with \( \Delta \) between \( A' \) and \( A'' \).

**Proof.** There is no event change when moving from \( A' \) to \( A'' \) by Lemma 2.5.8. Therefore for every realization \( r \) of the processing duration vector \( p, F(.|p = r) \) changes linearly with \( \Delta \) between \( A' \) and \( A'' \). Hence, \( F(.) = E_p [F(.|p)] \), \( F \) also changes linearly with \( \Delta \) between \( A' \) and \( A'' \). \( \square \)

**Theorem 2.5.10.** (Appointment Vector Integrality) If the processing durations are integer random variables then there exists an optimal appointment vector which is integer.

**Proof.** By Lemma 2.4.3 we know that there exists an optimal appointment schedule in the set \( \mathcal{K} = \{ A \in \mathbb{R}^{n+1} : A \leq A \leq \overline{A} \} \). Let \( \mathcal{A} \) denote the set of all such optimal appointment vectors in \( \mathcal{K} \), so \( \mathcal{A} \) is nonempty, bounded and closed, since by Lemma 2.4.2 \( F \) is continuous.

For \( A \in \mathcal{A} \) let

\[
I(A) = \begin{cases} 
\min \{ A_j : j \in \{2, \ldots, n + 1\} \text{ and } A_j \not\in \mathbb{Z} \} & \text{if } A \not\in \mathbb{Z}^{n+1} \\
{n\bar{p}_{\text{max}} + 1} & \text{if } A \in \mathbb{Z}^{n+1}.
\end{cases}
\]

We claim \( I(.) \) is upper semi continuous (usc) on the compact set \( \mathcal{A} \). If \( A \in \mathcal{A} \cap \mathbb{Z}^{n+1} \) then \( I(A) = h + 1 \geq I(B) \) for all \( B \in \mathcal{A} \), implying that \( I(.) \) is usc at \( A \). Otherwise \( A \in \mathcal{A} \setminus \mathbb{Z}^{n+1} \), and let \( I(A) = A_f \). For any \( \epsilon > 0 \) let \( \delta = \min\{\epsilon, I(A) - [A_f], [A_f] - I(A)\} > 0 \). For all \( B \in \mathcal{A}, ||B - A|| < \delta \) implies \( B_f > A_f - \delta \geq A_f - (I(A) - [A_f]) = [A_f] \) and \( B_f < A_f + \delta \leq [A_f] + \delta \).
\[ A_f + [A_f] - I(A) = [A_f]. \] Therefore \( B_f \) is fractional so \( I(B) \leq B_f \leq A_f + \epsilon \), thus \( I(A) + \epsilon \). Therefore \( I(.) \) is usc at \( A \in A \setminus \mathbb{Z}^{n+1} \). This completes the proof that \( I(.) \) is usc on \( A \).

The fact that \( I(.) \) is usc and \( A \) is compact implies that there exists an element \( A^* \) of \( A \) maximizing \( I(.) \). Seeking a contradiction, assume \( A^* \notin \mathbb{Z}^{n+1} \). Let \( f = \min\{i : A_i^* = I(A_i^*)\} \), so for all \( j < f \), \( A_j^* < I(A^*) \) and thus \( A_j^* \in \mathbb{Z} \). Let \( A' \) and \( A'' \) be the schedules derived from \( A^* \) as defined at the beginning of this section. By optimality \( F(A^*) \leq F(A') \) and \( F(A^*) \leq F(A'') \). But by Lemma 2.5.9, \( F(A^*) \) changes linearly with \( \Delta \) between \( A' \) and \( A'' \). Hence we must have \( F(A^*) = F(A') = F(A'') \). Note that \( A'' \geq A^* \geq A \) and, for every \( j \in J \), \( A_j'' = A_j^* + \Delta < [A_j^*] \leq J \) so \( A'' \leq A \). This shows that \( A'' \in \mathcal{K} \) and therefore \( A'' \in A \). But \( I(A^*) = A_f^* < A_f'' + \Delta = A_f'' = I(A'') \), i.e., \( I(A^*) < I(A'') \), a contradiction with the definition of \( A^* \).

\[ \square \]

**Remark 2.5.11.** Linear overage and underage costs are essential for the integrality of an optimal appointment vector. Consider the following example with quadratic costs. Let \( n = 1 \) and \( F(A) = E_p \left[ o_1 ((C_1 - A_2)')^2 + u_1 ((A_2 - C_1)')^2 \right] \) with \( o_1 = u_1 = 1 \); and \( \text{Prob}\{p_1 = 1\} = \text{Prob}\{p_1 = 2\} = \frac{1}{2} \). Then \( F(A) = E_p \left[ 2(C_1 - A_2)^2 \right] \), \( C_1 = p_1 \), and the optimum is \( A_2^* = E_p(p_1) = \frac{3}{2} \) which is not integer.

### 2.6 L-convexity

We start by investigating an important property of our objective function, submodularity (see e.g., [9], [30] and [18]).

**Definition 2.6.1.** A function \( f : \mathbb{Z}^q \to \mathbb{R} \) is **submodular** iff \( f(z) + f(y) \geq f(z \lor y) + f(z \land y) \) for all \( z, y \in \mathbb{Z}^q \) where \( z \lor y = (\max(z_i, y_i) : 0 \leq i \leq q) \in \mathbb{Z}^q \), \( z \land y = (\min(z_i, y_i) : 0 \leq i \leq q) \in \mathbb{Z}^q \) ([18]).

We now define a property of an appointment vector and a realization of the processing durations that will play an important role in this section.

**Definition 2.6.2.** A quadruple \((i, j, k, l)\) is a **submodularity obstacle** for appointment schedule \( A \) and a realization \( r \) of the processing durations if

- \( 1 \leq i < j < k < l \leq n + 1 \);
- the cost coefficients satisfy \( o_{j-1} + u_{j-1} + \sum_{j \leq t < k-1} o_t < u_{k-1} \);
and, in schedule $A|p = r$

- both jobs $i$ and $j$ are on-time;
- job $l$ is the last job which starts on-time before job $n + 1$;
- there is no idle time between jobs $i$ and $j$;
- there is positive idle time between jobs $j$ and $l$; and
- job $k$ is the first early job after $j$.

**Proposition 2.6.3.** For any realization $r$ of the processing durations, the function $F(., p = r)$ is submodular if and only if there is no submodularity obstacle for any integer appointment vector $A$.

*Proof.* Let $r$ be any realization of the processing durations $p$. By the proof of Theorem 6.19 from Murota [18], $F(., p)$ is submodular if and only if

$$F(A + 1_i + 1_j|p = r) - F(A + 1_i|p = r) \leq F(A + 1_j|p = r) - F(A|p = r)$$

(2.2)

for each $A \in \mathbb{Z}^{n+1}$ and $1 \leq i < j \leq n + 1$. Let $A \in \mathbb{Z}^{n+1}$ and $i < j$. Let $l$ be the last on-time job before job $n + 1$. Job $l$ is well defined since job 1 always on-time with $S_1 = A_1$.

We consider the following cases for job $l$.

(A) $1 \leq l < j \leq n + 1$, i.e., job $j$ is late.

(B) $l = j$, i.e., job $j$ is on-time, and all the jobs after job $j$ are late.

(C) $j < l \leq n + 1$.

To ease notation we use $(., r)$ to denote schedule $(., p = r)$. We now verify the submodular inequality (2.2) in each case.

**Case (A)** ($l < j \leq n + 1$). Job $j$ is late for both schedules $(A|r)$ and $(A + 1_i|r)$, and job $j$ remains not early when $A_j$ is replaced with $A_j + 1$, therefore $F(A + 1_i + 1_j|r) - F(A + 1_i|r) = -o_{j-1}$ and $F(A + 1_j|r) - F(A|r) = -o_{j-1}$. As a result, (2.2) holds with equality.

**Case (B)** ($l = j \leq n + 1$). Job $j$ is the last on-time job for schedule $(A|r)$, and $(A + 1_j|r)$ pushes every job after job $j - 1$ to the right by one unit. Therefore $F(A + 1_j|r) - F(A|r) =$
\( u_{j-1} + o_j + o_{j+1} + \ldots + o_n \geq 0. \) If there is an idle slot between \( i \) and \( j \) then job \( j \) will still be on-time in schedules \((A + 1_i + 1_j|\mathbf{r})\) and \((A + 1_i|\mathbf{r})\). Since every job after job \( j - 1 \) in \((A + 1_i + 1_j|\mathbf{r})\) will also be pushed to the right by one unit, \( F(A + 1_i + 1_j|\mathbf{r}) - F(A + 1_i|\mathbf{r}) = u_{j-1} + o_j + o_{j+1} + \ldots + o_n \) and (2.2) holds with equality. Otherwise, there is no idle slot between \( i \) and \( j \). Then job \( j \) will be late in schedule \((A + 1_i|\mathbf{r})\) but on-time in schedule \((A + 1_i + 1_j|\mathbf{r})\) and all jobs \( k > j \) have the same start times in both schedules \((A + 1_i + 1_j|\mathbf{r})\) and \((A + 1_i|\mathbf{r})\). Therefore, \( F(A + 1_i + 1_j|\mathbf{r}) - F(A + 1_i|\mathbf{r}) = -o_{j-1} \leq 0 \) and inequality (2.2) holds.

**Case (C) (j < l ≤ n + 1).** If job \( j \) is late in schedule \((A|\mathbf{r})\) then it is also late in schedule \((A + 1_i|\mathbf{r})\), and it remains not early when \( A_j \) is replaced with \( A_j + 1 \). Therefore, \( F(A + 1_j|\mathbf{r}) - F(A|\mathbf{r}) = -o_{j-1} \) and \( F(A + 1_i + 1_j|\mathbf{r}) - F(A + 1_i|\mathbf{r}) = -o_{j-1} \). As a result, (2.2) holds with equality. Therefore assume that job \( j \) is on-time in schedule \((A|\mathbf{r})\). If there is positive idle time between \( i \) and \( j \) in schedule \((A|\mathbf{r})\), then \( j \) remains on-time in schedule \((A + 1_i|\mathbf{r})\) hence also remains on time \((A + 1_i + 1_j|\mathbf{r})\) and \((A + 1_j|\mathbf{r})\) and (2.2) holds with equality. Therefore we also assume that there is no idle time between \( i \) and \( j \).

We consider two subcases, CR1 and CR2, for the right hand side \( F(A + 1_j|\mathbf{r}) - F(A|\mathbf{r}) \) and three subcases, CL1, CL2 and CL3, for the left hand side \( F(A + 1_i + 1_j|\mathbf{r}) - F(A + 1_i|\mathbf{r}) \):

- **CR1** there is no idle time between \( j \) and \( l \);
- **CR2** there is positive idle time\(^7\) between \( j \) and \( l \);
- **CL1** job \( i \) is on-time;
- **CL2** job \( i \) is late, and there is no idle time between \( j \) and \( l \);
- **CL3** job \( i \) is late and there is positive idle time between \( j \) and \( l \).

In CR1, the time interval \([A_j, A_j + 1]\) is idle in schedule \((A + 1_j|\mathbf{r})\) and every job \( j, j + 1, \ldots, n \) incurs one more unit of overtime in schedule \((A + 1_j|\mathbf{r})\) than in schedule \((A|\mathbf{r})\) since all jobs between \( j \) and \( l \) are not early and all jobs after \( l \) are late. Hence, in CR1, \( F(A + 1_j|\mathbf{r}) - F(A|\mathbf{r}) = u_{j-1} + o_j + o_{j+1} + \ldots + o_n \).

In CR2, there is an early job \( k \) between \( j \) and \( l \). Choose \( k \) to be the first early job after \( j \) so \( j < k < l \). Similarly to CR1, the time interval \([A_j, A_j + 1]\) is idle in schedule \((A + 1_j|\mathbf{r})\) and every job \( j, j + 1, \ldots, k - 1 \) incurs one more unit of overtime in schedule \(A + 1_j|\mathbf{r})\) and every job \( j, j + 1, \ldots, k - 1 \) incurs one more unit of overtime in schedule

\(^7\)This means that there is at least one idle slot available between the jobs under consideration. In this case, there exists at least one job which starts on-time in the interval.
(A + 1_j|r) than in schedule (A|r) since all jobs between j and k are not early. Furthermore, job k - 1 incurs one less unit of idle time in schedule (A + 1_j|r) than in schedule (A|r) since k remains not late in schedule (A + 1_j|r). Hence, in CR2, \( F(A + 1_j|r) - F(A|r) = u_{j-1} + o_j + o_{j+1} + ... + o_{k-3} + o_{k-2} - u_{k-1} \).

In CL1, job i remains on-time in both schedules (A + 1_i|r) and (A + 1_i + 1_j|r), and because there is no idle time between i and j, job j is late in schedule (A + 1_i|r) but on-time in schedule (A + 1_i + 1_j|r). Therefore, schedule (A + 1_i + 1_j|r) will have one unit less overtime (just before job j) than schedule (A + 1_i|r). Hence, in CL1, \( F(A + 1_i + 1_j|p) - F(A + 1_i|p) = -o_{j-1} \).

In CL2, job j is just-on-time in schedule (A + 1_i|r) but one time unit early in schedule (A + 1_i + 1_j|r). In schedule (A|r), all jobs between j and l are not early (since there is no idle time between j and l), and all jobs after l are late (since l is the last on-time job). Furthermore, all jobs after j are late in schedule (A + 1_i|r) and therefore also late in schedule (A + 1_i + 1_j|r) because there is no idle time between i and j in schedule (A|r). As a result, schedule (A + 1_i + 1_j|r) has an idle slot just before \( A_j + 1 \) and one more unit of overtime for each job \( j, \ldots, n + 1 \) than schedule (A + 1_i|r). Hence, in CL2, \( F(A + 1_i + 1_j|p) - F(A + 1_i|p) = u_{j-1} + o_j + o_{j+1} + ... + o_n \).

Similarly to CL2, in CL3, job j is just-on-time in schedule (A + 1_i|r) but one time unit early in schedule (A + 1_i + 1_j|r). Furthermore, there is a first early job k between j + 1 and l since there is positive idle time between j and l in schedule (A|r). The time interval \( [A_j, A_{j+1}] \) is idle in schedule (A + 1_i + 1_j|r) and every job \( j, j + 1, \ldots, k - 1 \) incurs one more unit of overtime in schedule (A + 1_i + 1_j|r) than in schedule (A + 1_i|r). Furthermore, job k - 1 incurs one less unit of idle time in schedule (A + 1_i + 1_j|r) than in schedule (A + 1_i|r). Hence, in CL3, \( F(A + 1_i + 1_j|r) - F(A + 1_i|r) = u_{j-1} + o_j + o_{j+1} + ... + o_{k-3} + o_{k-2} - u_{k-1} \).

Note that we have the same job k as in CR2.

As a result, \( F(A + 1_i + 1_j|p) - F(A + 1_i|p) - (F(A + 1_j|p) - F(A|p)) \)

\[
= \begin{cases} 
-o_{j-1} - (u_{j-1} + o_j + o_{j+1} + ... + o_n) & \text{if CR1 and CL1} \\
0 & \text{if CR1 and CL2} \\
-o_{j-1} - (u_{j-1} + o_j + o_{j+1} + ... + o_{k-3} + o_{k-2} - u_{k-1}) & \text{if CR2 and CL1} \\
0 & \text{if CR2 and CL3} 
\end{cases}
\]

If there is no submodularity obstacle then inequality \( -o_{j-1} \leq u_{j-1} + o_j + o_{j+1} + ... + o_{k-3} + o_{k-2} - u_{k-1} \)
\[ o_{k-3} + o_{k-2} - u_{k-1} \] in CR2 and CL1 is satisfied and \( F(.|p) \) is submodular.

Conversely, if \( F(.|p) \) is submodular then \(-o_{j-1} \leq u_{j-1} + o_j + o_{j+1} + \ldots + o_{k-3} + o_{k-2} - u_{k-1}\) for all jobs \( i < j < k < l \) such that \( j \) is on-time, there is no idle time between \( i \) and \( j \), there is positive idle time between \( j \) and \( l \) and job \( i \) is on-time; i.e., there is no submodularity obstacle for the appointment vector \( A \) and processing duration realization \( r \), hence there cannot be a submodularity obstacle. \( \square \)

**Corollary 2.6.4.** If there is no submodularity obstacle for any integer appointment vector \( A \) and processing duration realization \( r \) then \( F \) is submodular.

*Proof.* The result holds since submodularity is preserved under expectation, \( F(.) = E_p [F(.|p)] \), and by Proposition 2.6.3 \( F(.|p) \) is submodular if there is no submodularity obstacle for any integer appointment vector \( A \) and processing duration realization \( r \). \( \square \)

A submodularity obstacle is a very specific configuration, and it does not exist with reasonable cost structures such as nonincreasing \( u_i \)'s \( (u_{i+1} \leq u_i \) for all \( i \)) or nonincreasing \( (o_i + u_i) \)'s \( (o_{i+1} + u_{i+1} \leq o_i + u_i \) for all \( i \)). To capture these cost structures we define the following:

**Definition 2.6.5.** The cost coefficients \( (u, o) \) are \( \alpha \)-monotone if there exists reals \( \alpha_i (1 \leq i \leq n) \) such that \( 0 \leq \alpha_i \leq o_i \) and \( u_i + \alpha_i \) are non-increasing in \( i \), i.e., \( u_i + \alpha_i \geq u_{i+1} + \alpha_{i+1} \) for all \( i = 1, \ldots, n - 1 \).

The following Lemma establishes a relation between existence of a submodularity obstacle and \( \alpha \)-monotonicity.

**Proposition 2.6.6.** If the cost coefficients \( (u, o) \) are \( \alpha \)-monotone then there is no submodularity obstacle for any integer appointment vector \( A \) and processing duration realization \( r \).

*Proof.* Assume \( (u, o) \) are \( \alpha \)-monotone. We will show that for every \( j \in \{2, \ldots, n\} \) there exists \( t \geq j + 1 \) such that \( o_{j-1} + u_{j-1} + \sum_{r=j}^{t-1} o_r \geq u_{t-1} \). For contradiction suppose,
\( o_{j-1} + u_{j-1} + \sum_{r=j}^{t-1} o_r < u_{t-1} \) for all \( t \geq j + 1 \). Then

\[
\alpha_{t-1} + o_{j-1} + u_{j-1} + \sum_{r=j}^{t-1} o_r < u_{t-1} + \alpha_{t-1} \quad \text{(add \( \alpha_{t-1} \) to both sides)}
\]

\[
\alpha_{t-1} + o_{j-1} + u_{j-1} + \sum_{r=j}^{t-1} o_r < u_{t-1} + \alpha_{t-1} \quad \text{(since \( o_{j-1} \leq o_{j-1} \))}
\]

\[
\alpha_{j-1} + u_{j-1} < u_{t-1} + \alpha_{t-1} \quad \text{(since \( \sum_{r=j}^{t-1} o_r + \alpha_{t-1} \geq 0 \))},
\]

but this is a contradiction to \( \alpha \)-monotonicity. Therefore the result follows. \( \square \)

**Theorem 2.6.7. (Submodularity)** If the cost vectors \((u, o)\) are \( \alpha \)-monotone then \( F \) is submodular.

**Proof.** If the cost vectors \((u, o)\) are \( \alpha \)-monotone then by Proposition 2.6.6 there is no submodularity obstacle for any integer appointment vector \( A \) and processing duration realization \( r \). Hence the result follows from Corollary 2.6.4. \( \square \)

Completion times, start times and tardiness and their expectations are also submodular:

**Corollary 2.6.8.** The tardiness \( T_k \), start time \( S_k \), completion time \( C_k \), and their expected values \( E_p[T_k], E_p[S_k] \) and \( E_p[C_k] \) are submodular functions of \( A \) for every \( k = 1, \ldots, n \).

**Proof.** Recall that \( F(\cdot |p) = \sum_{i=1}^{n} (o_i T_i + u_i E_i) \). Let \( 1 \leq k \leq n, u_i = 0 \) for all \( i \), and \( o_i = 1 \) if \( i = k \) and 0 otherwise. Then \( T_k = F(\cdot |p) \). Therefore \( T_k \) is submodular whenever \( F(\cdot |p) \) is. By Proposition 2.6.3, \( F(\cdot |p) \) is submodular if there is no submodularity obstacle. But the chosen \( u_i \)'s and \( o_i \)'s are \( \alpha \)-monotone so no submodularity obstacle exists by Proposition 2.6.6. As a result \( F(\cdot |p) \) and hence \( T_k \) is submodular. Next we show \( S_k \) is submodular. \( S_1 = 0 \) and \( S_k = A_k + \max\{0, C_{k-1} - A_k\} = A_k + T_{k-1} \) \((1 < k \leq n)\) by definition. Since \( A_k \) is a scalar and \( T_k \) is submodular, \( S_k \) is also submodular. Similarly, \( C_k = S_k + p_k \) \((1 \leq k \leq n)\) by definition. Since \( p_k \) is a scalar and \( S_k \) is submodular, \( C_k \) is submodular. Finally, the expected values \( E_p[T_k], E_p[S_k] \) and \( E_p[C_k] \) are submodular since submodularity is preserved under expectation and \( T_k, S_k \) and \( C_k \) are submodular. This completes the proof. \( \square \)

**Remark 2.6.9.** The earliness \( E_k \) is not a submodular function of \( A \) in general. To see this let \( A = (0, 3, 5, 6, 9) \), deterministic processing durations \( p_1 = 3, p_2 = 2, p_3 = 2, p_4 = 1 \).

\( E_4(A) = (A_5 - C_4)^+ = (9 - 8)^+ = 1 \), similarly \( E_4(A + 1_1 + 1_2) = 0, E_4(A + 1_1) = 0 \) and \( E_4(A + 1_2) = 0 \). Therefore \( 1+0 = E_4(A) + E_4(A + 1_1) + E_4(A + 1_2) > E_4(A + 1_1) + E_4(A + 1_2) = 0 + 0 \). Hence \( E_4 \) is not submodular.
The objective function is not only submodular but also L-convex, an important discrete convexity property. Before we show L-convexity results, we give the definition of L-convexity.

**Definition 2.6.10.** $f : \mathbb{Z}^q \rightarrow \mathbb{R} \cup \{\infty\}$ is L-convex iff $f(z) + f(y) \geq f(z \lor y) + f(z \land y) \ \forall z, \forall y \in \mathbb{Z}^q$ and $\exists r \in \mathbb{R} : f(z + 1) = f(z) + r \ \forall z \in \mathbb{Z}^q$ ([18]).

**Proposition 2.6.11.** For any realization $r$ of the processing durations, the function $F(\cdot \mid p = r)$ is L-convex if and only if there is no submodularity obstacle for any integer appointment vector $A$ and realization $r$.

*Proof.* Let $r$ be a realization of the processing durations. If there is no submodularity obstacle for any integer appointment vector $A$ and realization $r$ then $F(\cdot \mid p = r)$ is submodular by Proposition 2.6.3, the first property in the definition of L-convexity.

Recall that $F(A \mid p = r) = \sum_{i=1}^n (o_i T_i + u_i E_i)$, $T_i = (C_i - A_{i+1})^+$ and $E_i = (A_{i+1} - C_i)^+$. Consider $F(A + 1 \mid p = r) = \sum_{i=1}^n (o_i T_i^1 + u_i E_i^1)$, where $x_i^1$ = quantity of interest of job $i$ with appointment vector $A + 1$ for $x \in \{S, C, T, E\}$. Then $S_i^1 = S_i + 1$ and $C_i^1 = C_i + 1$ hence $T_i^1 = T_i$ and $E_i^1 = E_i$. Therefore $F(A + 1 \mid p = r) - F(A \mid p = r) = 0$. This gives us the second property of L-convexity definition.

Conversely, if $F(\cdot \mid p = r)$ is L-convex then $F(\cdot \mid p)$ must be submodular and by Proposition 2.6.3 there is no submodularity obstacle for any integer appointment vector $A$. □

**Corollary 2.6.12.** If there is no submodularity obstacle for any integer appointment vector $A$ and realization $r$ then $F(\cdot)$ is L-convex.

*Proof.* The claim holds since L-convexity is preserved under expectation, $F(\cdot) = E_p[F(\cdot \mid p)]$, and by Proposition 2.6.11 $F(\cdot \mid p)$ is L-convex if there is no submodularity obstacle for any integer appointment vector $A$ and realization $r$. □

**Theorem 2.6.13.** (L-convexity) If the cost vectors $(u, o)$ are $\alpha$-monotone then $F(A)$ is L-convex.

*Proof.* If the cost coefficients $(u, o)$ are $\alpha$-monotone then by Proposition 2.6.6 there is no submodularity obstacle for any integer appointment vector $A$ and processing duration realization $r$. Therefore the result follows from Corollary 2.6.12. □
2.7 Algorithms

Using algorithmic results, [19] and [18], for minimizing L-convex functions, we can minimize the expected cost \( F \) in polynomial time, using a polynomial number of expected cost computations and submodular set minimizations.

Assume the input to our problem consists of the number \( n \) of jobs, the cost vectors \( u \) and \( o \), the horizon \( h \) over which \( F \) is to be minimized. Assume also that the processing times are integer and that we have an oracle which computes the expected cost \( F(A) \) for any given integer appointment vector \( A \).

**Theorem 2.7.1. (Polynomial Time Algorithm 1)** If the cost vectors \( (u, o) \) are \( \alpha \)-monotone and the processing durations are integer then there exists an algorithm which minimize \( F \) using polynomial time and a polynomial number of expected cost evaluations.

**Proof.** The Appointment Vector Integrality Theorem 2.5.10 implies that to minimize \( F \) we only need to consider integer appointment vectors. If the cost vectors \( (u, o) \) are \( \alpha \)-monotone then \( F \) is an L-convex function by the L-convexity Theorem 2.6.13. Then \( F \) can be minimized in \( O(\sigma(n) EO n^2 \log([h/2n])) \) time by Iwata’s steepest descent scaling algorithm (Section 10.3.2 of Murota [18]) where \( \sigma(n) \) is the number of function evaluations required to minimize a submodular set function over an \( n \)-element ground set and \( EO \) is the time needed for an expected cost evaluation. \( \square \)

When the processing durations are independent, the expected cost of an integer appointment vector can be evaluated efficiently. We use recursive equations for the probability distributions of the start time, completion time, tardiness and earliness of each job and compute \( F \) at an integer point \( A \) in \( O(n^2p_{\text{max}}^2) \) time.

**Theorem 2.7.2.** If the processing durations are stochastically independent and \( A \) is an integer appointment vector then \( F(A) \) may be computed in \( O(n^2p_{\text{max}}^2) \) time.

**Proof.** The first job starts at time zero so \( S_1 = A_1 = 0 \), and \( C_1 = p_1 \), i.e., the distribution of \( C_1 \) is that of \( p_1 \). Next, we look at the start times \( S_i (2 \leq i \leq n) \). We have \( S_i = \max(A_i, C_{i-1}) \) so for all \( k = 0, 1, \ldots, np_{\text{max}} \),

\[
\text{Prob}\{S_i = k\} = \begin{cases} 
0 & \text{if } k < A_i \\
\text{Prob}\{C_{i-1} \leq k\} & \text{if } k = A_i \\
\text{Prob}\{C_{i-1} = k\} & \text{if } k > A_i.
\end{cases}
\] (2.3)
Note that $S_i$ and $p_i$ are independent because $S_i$ is completely determined by $p_1, p_2, \ldots, p_{i-1}$ and $A_1, A_2, \ldots, A_i$. Since $C_i = S_i + p_i$, by conditioning on $p_i$ and using independence of $p_i$ and $S_i$, we obtain for all $k = 0, 1, \ldots, n\bar{p}_{\text{max}}$,

$$
\text{Prob}\{C_i = k\} = \text{Prob}\{S_i = k - p_i\} = \sum_{j=0}^{\bar{p}_i} \text{Prob}\{S_i = k - j\} \text{Prob}\{p_i = j\}, \quad (2.4)
$$

and $\text{Prob}\{C_{i-1} \leq k\} = \text{Prob}\{C_{i-1} = k\} + \text{Prob}\{C_{i-1} \leq k - 1\}$. For each $i - 1$, $\text{Prob}\{C_{i-1} \leq k\}$ may be computed in $O((i-1)\bar{p}_{\text{max}})$ time. Hence $\text{Prob}\{C_i = k\}$ can be computed once we have the distribution of $S_i$. For each job $i$ and value $k$, computing $\text{Prob}\{S_i = k\}$ by Eq(2.3) requires a constant number of operations, and computing $\text{Prob}\{C_i = k\}$ by Eq(2.4) requires $O(\bar{p}_i + 1)$ operations. Therefore the total number of operations needed for computing the whole start time and completion time distributions for job $i$ is $O(n^{\bar{p}_i^2})$. The distribution of $T_i$ and $E_i$, their expected values $E_p T_i$ and $E_p E_i$ can then be determined in $O(n\bar{p}_{\text{max}})$ time. Therefore, the objective value $F(A)$ is obtained in $O(n^{\bar{p}_{\text{max}}^2})$ time.

The running time of the algorithm given in Theorem 2.7.1 depends on how the distributions of the processing durations are given. Under the common assumption of independent durations, the input to the algorithm includes the distribution of each processing duration $p_i$, which specifies $\bar{p}_i + 1$ probabilities $\text{Prob}\{p_i = x\}$ for $x = 0, 1, \ldots, \bar{p}_i$. In this case, $F$ can be minimized in $O(n^{\bar{p}_{\text{max}}^2} \log \bar{p}_{\text{max}})$ time.

**Theorem 2.7.3. (Polynomial Time Algorithm 2)** If the processing durations are independent, integer-valued random variables and the cost vectors $(u, o)$ are $\alpha$-monotone then we can minimize $F$ in $O(n^{\bar{p}_{\text{max}}^2} \log \bar{p}_{\text{max}})$ time.

**Proof.** The horizon $h$ can be taken as $n\bar{p}_{\text{max}} \geq \sum_{i=1}^{n} \bar{p}_i$, so $h$ is polynomially bounded in the input size. Theorem 2.7.2 shows that $EO = O(h^2)$ when processing durations are independent. Theorem 4 of Orlin [21] shows that $\sigma(n) = O(n^5)$. The result follows from Theorem 2.7.1. \qed

### 2.8 Objective Function with a Due Date

Suppose that we are given a due date $D$ for the end of processing, instead of letting the model choose a planned makespan $A_{n+1}$. We assume $D$ is integer and $0 \leq D \leq \sum_{i=1}^{n} \bar{p}_i$. 40
Define \( \mathbf{A} = (A_1, A_2, ..., A_n) \), then our new objective becomes

\[
F^D(\mathbf{A}) = \mathrm{E_p} \left[ \sum_{j=1}^{n-1} \left( o_j (C_j - A_{j+1})^+ + u_j (A_{j+1} - C_j)^+ \right) + o_n (C_n - D)^+ + u_n (D - C_n)^+ \right].
\]

We immediately observe that \( F(\mathbf{A}, D) = F^D(\mathbf{A}) \). Like \( F \), \( F^D \) has many properties such as discrete convexity (\( F^D \) is \( L^\natural \)-convex, see Definition 2.8.4), optimal vector integrality and existence of a polynomial time minimization algorithm.

We verify the properties of \( F^D \). Let \( \tilde{K} = \{ \tilde{\mathbf{A}} \in \mathbb{R}^n : \tilde{A}_1 = 0, \sum_{j<i} p_j \leq \tilde{A}_i \leq \sum_{j<i} \tilde{p}_j \text{ for all } i = 2, \ldots, n \} \). Since \( F(\tilde{\mathbf{A}}, D) = F^D(\tilde{\mathbf{A}}) \), by using our previous results on \( F \) we obtain the following for \( F^D \).

**Corollary 2.8.1.**

1. Critical Path Lemma 2.4.1 applies to \( F^D \).

2. Function \( F^D \) is continuous.

3. There exists an optimal appointment schedule \( \tilde{\mathbf{A}}^* \in \tilde{K} \).

4. There exists an optimal appointment schedule \( \tilde{\mathbf{A}}^* \) satisfying

\[
p_i \leq \tilde{A}_{i+1}^* - \tilde{A}_i^* \leq \sum_{j\leq i} \tilde{p}_j - \sum_{j<i} \tilde{p}_j \text{ for } i = 1, \ldots, n - 1.
\]

5. There exists an optimal appointment vector \( \tilde{\mathbf{A}}^* \in \tilde{K} \) with components non-decreasing, i.e., \( \tilde{A}_i^* \leq \tilde{A}_{i+1}^* \) for all \( i = 1, \ldots, n - 1 \).

6. \( F^D(\tilde{\mathbf{A}}) \) may be computed in \( O(n^2 p^2_{\max}) \) if processing durations are independent and \( \tilde{\mathbf{A}} \) is integer.

**Proof.**

1. Follows directly from Critical Path Lemma 2.4.1.

2. Continuity is preserved by projection onto a coordinate subspace. Therefore, the result follows from Lemma 2.4.2.

3. The feasible set for \( F^D \) is compact since \( 0 \leq D \leq \sum_{i=1}^n p_i \) and the compactness is preserved by projection onto a coordinate subspace. Therefore, the result follows from Lemma 2.4.3.
4. Follows from Lemma 2.4.4 (by changing $1 \leq i \leq n$ to $1 \leq i \leq n - 1$) and the fact that $0 \leq D \leq \sum_{i=1}^{n} \bar{p}_i$.

5. Follows from Non-Decreasing Appointment Dates Lemma 2.4.5 (by changing $1 \leq i \leq n$ to $1 \leq i \leq n - 1$) and the fact that $0 \leq D \leq \sum_{i=1}^{n} \bar{p}_i$.

6. Since $F(\tilde{A}, D) = F^D(\tilde{A})$, we can compute $F^D(\tilde{A})$ exactly the same way we compute $F(\tilde{A})$ with $A_{n+1} = D$. Therefore the result follows from Theorem 2.7.2.

\[\square\]

We next verify that appointment vector integrality also holds for $F^D$.

**Corollary 2.8.2. (Appointment Vector Integrality)** If the processing durations are integer random variables and the due date is integer then there exists an optimal appointment vector which is integer.

*Proof.* Let $\tilde{A}^*$ be any non-integer appointment vector and $\tilde{A}'^*$ the first non-integer component of $\tilde{A}^*$. As before, we define set $J$, $\varphi(x)$, $\Delta$, $\tilde{A}' = (\tilde{A}'_1, \ldots, \tilde{A}'_n)$ and $\tilde{A}'' = (\tilde{A}''_1, \ldots, \tilde{A}''_n)$. We consider any realization $\mathbf{r}$ of the processing durations. Then, Lemmata 2.5.1-2.5.9 follow for $F^D$ (either directly or by taking $A_{n+1}^* = D$).

By Corollary 2.8.1 we know that there exists an optimal appointment schedule which is in the set $\tilde{\mathcal{K}} = \{\tilde{A} \in \mathbb{R}^n : \tilde{A}_1 = 0, \sum_{j<i} p_j \leq \tilde{A}_i \leq \sum_{j<i} \bar{p}_j \text{ for all } i = 2, \ldots, n\}$. Let $\tilde{A}$ denote the set of all such optimal appointment vectors in $\tilde{\mathcal{K}}$, so $\tilde{A}$ is nonempty, bounded and closed, since by Corollary 2.8.1 $F^D$ is continuous. For $\tilde{A} \in \tilde{A}$, we define $I(.)$ as before but by changing $\mathbf{A}$ to $\tilde{A}$ and $\mathbb{Z}^{n+1}$ to $\mathbb{Z}^n$. Then, $I(.)$ is upper semi continuous (usc) on $\tilde{A}$ since upper semi continuity is preserved by projection onto a coordinate subspace.

The fact that $I(.)$ is usc and $\tilde{A}$ is compact implies that there exists an element $\tilde{A}^*$ of $\tilde{A}$ maximizing $I(.)$. By contradiction, assume $\tilde{A}^* \notin \mathbb{Z}^n$. Let $f = \min \{ i : \tilde{A}^*_i = I(\tilde{A}^*) \}$, so for all $j < f$, $\tilde{A}^*_j < I(\tilde{A}^*)$ and thus $\tilde{A}^*_j \in \mathbb{Z}$. Let $\tilde{A}'$ and $\tilde{A}''$ be the schedules derived from $\tilde{A}^*$ as defined at the beginning of Section 2.5 and this proof. By optimality $F(\tilde{A}^*) \leq F(\tilde{A}')$ and $F(\tilde{A}^*) \leq F(\tilde{A}'')$. But $F(\tilde{A}^*)$ changes linearly with $\Delta$ between $\tilde{A}'$ and $\tilde{A}''$ as Lemma 2.5.9 applies to $F^D$. Hence we must have $F(\tilde{A}^*) = F(\tilde{A}') = F(\tilde{A}'')$. Note that $\tilde{A}_i'' \leq \tilde{A}_i' \leq \sum_{k<i} \bar{p}_k$ for all $i = 1, \ldots, n$ and, for every $j \in J$, $\tilde{A}_j'' = \tilde{A}_j' + \Delta < \lceil \tilde{A}_j' \rceil \leq \sum_{k<i} \bar{p}_k$ so $\tilde{A}_i'' \leq \sum_{k<i} \bar{p}_k$ for all $i = 1, \ldots, n$. This shows that $\tilde{A}'' \in \tilde{\mathcal{K}}$ and therefore $\tilde{A}'' \in \tilde{A}$. But
\[ I(\tilde{A}^*) = \tilde{A}_f^* < \tilde{A}_f'' + \Delta = \tilde{A}_f'' = I(\tilde{A}''), \text{ i.e., } I(\tilde{A}^*) < I(\tilde{A}'') \], a contradiction with the definition of \( \tilde{A}^* \).

\[ \square \]

**Remark 2.8.3.** Integrality of \( D \) is crucial for an integer optimal appointment vector. Consider the following example.

\[ F^D(\tilde{A}) = E_p[\alpha_1(C_1 - \tilde{A}_2)^+ + u_1(\tilde{A}_2 - C_1)^+ + \alpha_2(C_2 - D)^+ + u_2(D - C_2)^+] \]

with \( \alpha_1 = u_1 = \alpha_2 = u_2 = 1 \), \( D = \frac{9}{2} \), \( p_2 = 3 \) (deterministic \( p_2 \)), and \( p_1 = 1 \) with probability \( \frac{1}{2} \) and \( p_1 = 2 \) with probability \( \frac{1}{2} \). Then, \( F^D\left(0, \frac{9}{2}\right) = u_2/4 + \alpha_1/2 + \alpha_2/4, F^D\left(0, \frac{3}{2}\right) = o_2/4 + u_1/2 + o_2/4 \) but \( F^D\left(0, \frac{3}{2}\right) = u_1/4 + \alpha_1/4 + o_2/4 \).

The next result is on the submodularity and discrete convexity (\( L^\bullet \)-convexity) of \( F^D \).

Before providing the result we need the definition of \( L^\bullet \)-convexity. A \( L^\bullet \)-convex function is obtained by restriction of a \( L \)-convex function to a coordinate plane [18].

**Definition 2.8.4.** A function \( f : Z^q \to R \cup \{\infty\} \) is said to be \( L^\bullet \)-convex if the function \( \tilde{f} : Z^{q+1} \to R \cup \{\infty\} \) defined by \( \tilde{f}(z, y) = f(z - y1) \) (\( z \in Z^q, y \in Z \)) is an \( L \)-convex function ([18]).

**Corollary 2.8.5.** (\( L^\bullet \)-convexity) If the cost coefficients \((u, o)\) are \( \alpha \)-monotone then \( F^D \) is submodular and \( L^\bullet \)-convex.

**Proof.** Assume that the cost coefficients \((u, o)\) are \( \alpha \)-monotone. Then \( F \) is submodular (by Submodularity Theorem 2.6.7) and \( L \)-convex (by \( L \)-convexity Theorem 2.6.13). Then, \( F^D \) is submodular (since submodularity is preserved by projection onto a coordinate subspace) and \( L^\bullet \)-convex (since \( F^D(\tilde{A}) = F(\tilde{A}, D) \) and \( F \) is \( L \)-convex). \( \square \)

Similarly to \( F \), we can minimize \( F^D \) by using algorithmic results for \( L^\bullet \)-convexity, [19] and [18], with a polynomial number of expected cost computations and submodular set minimizations. As in the case of \( F \), assume the input to our problem consists of the number \( n \) of jobs, the cost vectors \( u \) and \( o \), the horizon \( h \) over which \( F^D \) is to be minimized. We also assume that the processing times are integer and that we have an oracle which computes the expected cost \( F^D(\tilde{A}) \) for any given integer appointment vector \( \tilde{A} \).

**Corollary 2.8.6.** (Polynomial Time Algorithm 1) If the cost vectors \((u, o)\) are \( \alpha \)-monotone and the processing durations are integer then there exists an algorithm which minimize \( F^D \) using polynomial time and polynomial number of expected cost evaluations.
Proof. The Appointment Vector Integrality Corollary 2.8.2 implies that we only need to consider integer appointment vectors to minimize $F^D$. If the cost vectors $(u, o)$ are $\alpha$-monotone then $F^D$ is an $L^\alpha$-convex function by the $L^\alpha$-convexity Corollary 2.8.5. Then $F^D$ can be minimized in $O(\sigma(n) EO n^2 \log([h/2n]))$ time by Iwata’s steepest descent scaling algorithm (Section 10.3.2 of Murota [18]). □

As in the case of $F$, when the processing durations are independent, we can evaluate the expected cost of an integer appointment vector in $O(n^2 p_{\text{max}}^2)$ by Corollary 2.8.1. In the case of independent processing durations, the input to the algorithm in Corollary 2.8.6 includes the distribution of each processing duration $p_i$ and we can minimize $F^D$ in $O(n^9 p_{\text{max}}^2 \log p_{\text{max}})$.

**Corollary 2.8.7. (Polynomial Time Algorithm 2)** If the processing durations are independent, integer-valued random variables and the cost vectors $(u, o)$ are $\alpha$-monotone then we can minimize $F^D$ in $O(n^9 p_{\text{max}}^2 \log p_{\text{max}})$.

Proof. The horizon $h$ can be taken as $n p_{\text{max}} \geq \sum_{i=1}^n p_i$, so $h$ is polynomially bounded in the input size. Corollary 2.8.1 shows that $EO = O(h^2)$ when processing durations are independent. Theorem 4 of Orlin [21] shows that $\sigma(n) = O(n^5)$. The result follows from Corollary 2.8.6. □

### 2.9 No-shows and Emergency Jobs

No-shows and emergency jobs may have important practical applications and implications. For example, no-shows can be quite important in certain outpatient exams such as MRI scans [12]. Similarly, emergencies, such as emergency surgeries or examinations, can be a huge factor affecting the planned appointment schedules. With minor modifications and assumptions, our model can handle no-shows and that arrive when the machine (processor) is busy processing original jobs, emergency jobs, in finding an optimal appointment schedule.

We first discuss no-shows. Suppose that there is some probability $\text{noshow}_i$ that job $i$ will not show up. If job $i$ does not show up then its processing duration becomes zero. On the other hand, if it does show up then its processing duration will be determined by its distribution. Therefore, all we need to do is to update the processing duration distribution $p_i$ of job $i$ to take this no-show possibility into account. We can do so by
multiplying \((1 - \text{noshow}_i)\) with \(Prob\{p_i = k\}\) for all \(k > 0\) and assign \(Prob\{p_i = 0\} = 1 - \sum_{k>0}Prob\{p_i = k\}\).

Emergency jobs arrive after the processing starts without any appointments, and they may need to be processed as soon as possible. We take a non-preemptive approach, i.e., we finish processing of the current job first. We assume that emergency jobs may arrive only during processing of a planned job, i.e., there is no emergency job arrival during idle time or the processing of an emergency job. This is a reasonable assumption when the ratio of total idle time between planned jobs to total processing durations of planned jobs is small and there are not many emergency jobs. Therefore during processing of planned job \(i\), some emergency jobs may arrive, and these emergency jobs will be processed back to back just after job \(i\) and before job \(i+1\). Observe that there will be no idle time between the processing of emergency jobs, so we may think of the duration of these emergency jobs processing as a lengthening of job \(i\)’s processing time. Therefore, the problem reduces to find the new processing duration distribution of job \(i\). Figure 2.3 shows an example of a schedule with emergency jobs.

![Figure 2.3: An Example Schedule with Emergency Jobs](image)

We assume that there can be at most certain number of emergency jobs that can arrive during processing of a planned job, and the distribution of number of emergency jobs arrivals (during each planned job) is given by a discrete probability distribution. Furthermore, processing duration distribution of emergency jobs is also given by a discrete probability distribution. There can be at most \(m^i_{\text{max}}\) emergency jobs that can arrive during the processing of job \(i\). Let \(p_e\) be the discrete processing duration distribution of emergency jobs (we use the same processing duration distribution for each emergency job, but one may take \(p_e\) as the discrete processing duration distribution of emergency jobs arriving during job \(i\)). We denote the total processing duration of emergency jobs that will be processed just after job \(i\)
as \( P^i \). Then, all we need to do is to find the distribution of the new job processing duration \( \tilde{p}_i = p_i + P^i \). Because once we have \( \tilde{p}_i \)'s, we can minimize \( F^D(.) = E_{\tilde{p}} \left[ F^D(.) | \tilde{p} \right] \) as we minimize \( F^D(.) = E_{p} \left[ F(.) | p \right] \), and \( F_E(.) = E_{\tilde{p}} \left[ F(.) | \tilde{p} \right] \) as we minimize \( F(.) = E_p \left[ F(.) | p \right] \), and solve the scheduling problem with emergency jobs.

We now obtain the distribution of \( \tilde{p}_i = p_i + P^i \). Let \( m^i \) \((0 \leq m^i \leq m^i_{\max})\) be the number of emergency jobs arriving during processing of job \( i \). We define \( P^i_k = \sum_{j=1}^{k} p^e \) \((1 \leq k \leq m^i_{\max})\). Distributions \( P^i_k \) \((1 \leq k \leq m^i_{\max})\) can be computed in a recursive manner starting from \( P^i_1 = p^e \) and computing \( P^i_k = P^i_{k-1} + p^e \) for \( 1 < k \leq m^i_{\max} \). Once we have the distribution of \( P^i_k \)'s we can find the distributions of \( P^i \) \((1 \leq i \leq n)\) as follows:

\[
\text{Prob}\{P^i = 0\} = \text{Prob}\{m^i = 0\} + \sum_{k=1}^{m^i_{\max}} \text{Prob}\{m^i = k\} \text{Prob}\{P^i_k = 0\}
\]

\[
\text{Prob}\{P^i = j\} = \sum_{k=1}^{m^i_{\max}} \text{Prob}\{m^i = k\} \text{Prob}\{P^i_k = j\} \quad \text{for } j = 1, 2, \ldots, m^i_{\max} \tilde{p}_{\max}.
\]

The last thing we need to do is to obtain the distribution of \( \tilde{p}_i \), \((\tilde{p}_i = p_i + P^i)\), a single convolution of random variables \( p_i \) (already available) and \( P^i \) (just obtained).

### 2.10 Current Work, Future Work and Conclusion

After developing our modeling framework and proving that we can find an optimal appointment schedule in polynomial time, we focus on practical implementation issues.

Our objective as a function of continuous appointment vector is non-smooth, but we show that the objective is convex and characterized its subdifferential in Chapter 3. We also obtain closed form formulas for the subdifferential as well as for any subgradient. This characterization is useful, it allows is to develop two important extensions.

In the first extension, in Chapter 3, we relax the perfect information assumption on the probability distributions of processing durations, i.e., we assume that processing duration distributions are not known and can only be statistically estimated on the basis of past data or statistical sampling. Our approach is non-parametric, and we assume no (prior) information about processing duration distributions. We develop a sample-based approach to determine the number of independent samples required to obtain a provably near-optimal solution with high confidence, i.e., the cost of the sample-based optimal schedule is with high probability no more than \((1 + \epsilon)\) times the cost of an optimal schedule determined from
knowing the true distributions. This result has important practical implications, as the true processing duration distributions are often not known and only their past realizations or some samples are available.

In another study, Appendix A, we use the subdifferential characterization with independent processing durations and compute a subgradient in polynomial time for any given appointment schedule. This is not a trivial task as the subdifferential formulas include exponentially many terms, and some of the probability computations are complicated. We also obtain an easily computable lower bound on the optimal objective value. Furthermore, we extend computation of the expected total cost (in polynomial time) for any (real-valued) appointment vector. These allow us to use non-smooth convex optimization techniques to find an optimal schedule. Although we already have a polynomial time algorithm to find an optimal appointment schedule, it is not clear at the moment which technique will work faster in practice. We are also considering hybrid algorithms based on both discrete convexity and non-smooth convex optimization combined with a special-purpose integer rounding method. Preliminary versions of these algorithms have been developed. The rounding algorithm takes any fractional solution and rounds it to an integer one with the same or improved objective value. We are planning to implement our algorithms and compare different approaches in computational experiments.

There are many exciting future directions for this research. One is to find an optimal sequence and appointment schedule simultaneously, i.e., given the jobs, determine a sequence and a job appointment schedule minimizing the total expected cost. This problem is likely to be hard, but it may be possible to develop heuristic algorithms with performance guarantees. Studying some special cases for this problem may shed light on the general case. Another one is to put our findings into practice. We are in contact with local healthcare organizations to apply our results with real data and compare the appointment schedules determined by our methods with current practices.

In this chapter, we study a discrete time version of the appointment scheduling problem and establish discrete convexity properties of the objective function. We prove that the objective function is L-convex under mild assumptions on cost coefficients. Furthermore, we show that there exists an optimal integer appointment schedule minimizing the objective. This result is important as it allows us to optimize only over integer appointment schedules without loss of optimality. All these results on the objective function and optimal
appointment schedule enable us to develop a polynomial time algorithm, based on discrete
convexity, that, for a given processing sequence, finds an appointment schedule minimizing
the total expected cost. When processing durations are stochastically independent we evalu-
ate the expected cost for a given processing order and an integer appointment schedule,
efficiently both in theory (in polynomial time) and in practice (computations are quite fast
as shown in our preliminary computational experiments). Independent processing durations
lead to faster algorithms. Our modeling framework can handle a given due date for the to-
tal processing (e.g., end of day for an operating room) after which overtime is incurred,
instead of letting the model choose an end date. We also extend our model and framework
to include no-shows and emergencies. We believe that our framework is sufficiently generic
so that it is portable and applicable to many appointment systems in healthcare as well as
in other areas.
2.11 Bibliography


3 A Sampling-Based Approach to Appointment Scheduling\textsuperscript{1}

We consider the problem of appointment scheduling with discrete random durations of Chapter 2 under the assumption that the duration probability distributions are not known and only a set of independent samples is available e.g., historical data. The goal is to determine an optimal planned start schedule, i.e., an optimal appointment schedule for a given sequence of jobs on a single processor such that the expected total underage and overage costs is minimized. We show that the objective function of the appointment scheduling problem is convex under a simple sufficient condition on cost coefficients. Under this condition we characterize the subdifferential of the objective function with a closed-form formula. We use this formula to determine bounds on the number of independent samples required to obtain provably near-optimal solution with high probability.

3.1 Introduction and Motivation

We consider the appointment scheduling problem with discrete random durations introduced in Chapter 2 but under the assumption that the probability distributions of job durations are not known and the only available information on the durations is a set of independent random samples, e.g., historical data.

We show that the objective function is convex under a simple condition on the cost parameters and characterize its subdifferential, and determine the number of independent samples required to obtain a provably near-optimal solution with high probability.

In the appointment scheduling problem jobs are processed on a single processor in a given sequence and one has to decide the planned starting time of each job, also called

\textsuperscript{1}A version of this chapter has been submitted for publication. Begen M.A., Levi R. and Queyranne M.
A Sampling-Based Approach to Appointment Scheduling.
appointment date. Jobs are not available before their appointment dates. Moreover, the process durations are priori random and are realized only after the appointment dates are set. Due to stochastic processing durations, some jobs may finish earlier, whereas some others may finish later, than the appointment date of the next job. If a job ends earlier than the next job’s appointment date then the system experiences under-age cost due to under-utilization of the processor. On the other hand, when a job finishes later than the next job’s appointment date, the system is exposed to over-age cost due the wait of the next job and/or overtime for the processor. Therefore there is an important trade-off between under-utilization, waiting and overtime, i.e., underage and overage. The goal is to find an optimal appointment schedule, i.e., appointment date vector which minimizes the total expected underage and overage costs. There are important real-world applications of this problem, especially in healthcare such as surgery scheduling, transportation and production, e.g., see Chapter 2 and the references therein. For example in surgery scheduling, we can think of surgeries as the jobs, operating room/surgeon as the processor and the hospital as the scheduler. As observed in practice, surgery durations show variability (e.g., see Figure 2.1 and Figure 4.4) and determining planned start times, i.e., setting appointment dates of surgeries, is an important and challenging task [8]. Surgery appointment schedule has a direct impact on amount of overtime and idle-time of operating room(s). Operating room’s overtime can be costly since it involves staff overtime as well as additional overhead costs, on the other hand, idle-time costs can also be high due to the opportunity cost of unused capacity. Similar trade-off exists in scheduling of container ship arrivals at a container terminal [34]. Another example comes from a production system where it has multiple stages and stochastic leadtimes and the objective is to determine planned leadtimes to minimize expected cost [10].

Researchers studied the appointment scheduling problem for the last 50 years, e.g., see [6], [8], [19], [5], [39]. Existing literature exclusively uses continuous processing durations with full probability characterization, i.e., the probability distributions of processing time of jobs are given as part of the input. Due to the continuous processing durations there are computational difficulties in the computation of the expected total cost. For a given

\(^2\text{To conform with scheduling terminology, we use the term date to denote a point in time. In most applications of appointment scheduling, the appointment “dates” are actually appointment times within the day for which the jobs are being scheduled.}\)
sequence of jobs, only small instances can be solved to optimality, larger instances require
heuristics. Chapter 2 studies a discrete time version of the appointment scheduling problem,
i.e., the processing durations are integer and given by a discrete probability distribution.
This assumption fits many applications, for example, surgeries and physician appointments
are scheduled on minute(s) basis (usually a block of certain minutes). (For instance, one 20
minute physician appointment could be two blocks of 10 minutes.) Chapter 2 establishes
discrete convexity ([27]) properties of the objective function and prove that the objective
function is L-convex ([26]) under a mild assumption on cost coefficients. Furthermore, it
shows that there exists an optimal integer appointment schedule minimizing the objective.
This result is important as it makes possible to optimize only over integer appointment
schedules without loss of optimality. All these results on the objective function and optimal
appointment schedule lead to a polynomial time algorithm, based on discrete convexity
([28]), that, for a given processing sequence, finds an appointment schedule minimizing the
total expected cost. This algorithm invokes a sequence of submodular set function min-
imizations, for which various algorithms are available, see e.g., [11], [25] and [18]. When
processing durations are stochastically independent the expected cost for a given processing
order and an integer appointment schedule is evaluated in polynomial time in Chapter 2.
Independent processing durations lead to faster algorithms. Chapter 2’s modeling frame-
work can include a given due date for the end of the processing of all jobs (e.g., end of day
for an operating room) after which overtime is incurred, instead of letting the model choose
an end date. The framework is also extended to include no-shows and some emergency jobs.

In Chapter 2 the discrete convexity, L-convexity, of the objective function is proved by
assuming integer appointment vectors. The definition of L-convexity includes submodular-
ity and a translation equivalence property. With integer appointment vectors, Chapter 2
establishes L-convexity by proving that the objective function is submodular (under a sim-
ple condition on cost coefficients) and the translation equivalence property is satisfied. In
this chapter, we show that the objective function of the appointment scheduling problem
(as a function of continuous appointment vector) is convex under the same simple condition
on the cost coefficients.

Convexity of the objective function has been discussed (explicitly or implicitly) in sev-
eral papers in literature so far [39, 5, 32, 8], but we believe our analysis is the first rig-
orous treatment of the subject with this simple condition. Under this simple sufficient
condition, the objective function is convex but non-smooth due to the kinks. Due to the non-differentiability of the objective function we work with subgradients (instead of derivatives). In fact, we characterize the set of all subgradients, i.e., the subdifferential at a given appointment date vector, with a closed-form formula. This is unusual since only a single subgradient may be obtained in most applications. We use the subdifferential characterization to relax the perfect information assumption of Chapter 2 on the probability distribution of processing times by establishing a link between the sampling-based solution quality and the number of samples.

Chapter 2 assumes complete information on the job duration distributions, i.e., there is an underlying discrete probability distribution for job durations, and this distribution is available and known fully. This may be the case for some applications. However, for others, the true duration distributions may not be known but their (past) realizations or some samples may be available. One good example for such an application comes from healthcare; hospitals and surgeons usually have some data available on the length of previous surgeries but no one knows what the true distribution for a certain type of surgery is.

When the true distribution is not known then the question is how to use these samples to find a “good solution”. We assume that there is an underlying joint discrete distribution for the job durations, but there is only a set of independent samples available. This may correspond to historical data, for example daily observations of surgery durations. In this chapter, we develop a sampling-based approach and determine the number of independent samples required to obtain a provably near-optimal solution with high probability, i.e., the cost of the sampling-based optimal schedule is with high probability no more than $(1 + \epsilon)$ times the cost of the optimal schedule that is computed based on the true distribution.

Job durations may not necessarily be independent but samples are. In other words, each sample is a vector of durations where each coordinate corresponds to job duration and these vectors are independent. Independence assumption of probability distributions (e.g., job durations, demands of different periods) is common but we do not require it in our analysis.

There has been much interest for studying stochastic models with partial probabilistic characterization. We see that inventory models, especially the newsvendor problem and its multiperiod extension, receive a lot of attention. Depending on how much is known about the true distribution(s) different approaches are possible.
One may know the family of the true distribution but be uncertain about its parameters. This is called the parametric approach, and in this case there is usually an initial prior belief on the uncertainty of the parameter values. This belief is revised with Bayesian updates on realizations of the distribution, e.g., see Ding et al. [9] and the references therein. Liyanage and Shanthikumar [23] introduced operational statistics, an approach that combines parameter estimation and optimization for the case of known family but unknown parameters and priors, see also [7] for more on operational statistics.

If there are no assumptions on the true distribution, i.e., no prior assumptions on its family or its parameters, then the approach becomes non-parametric. Levi et al. [22] use sample average approximation (SAA) (e.g., see [36]) to determine the number of samples required for the SAA solution to be a provably near-optimal (w.r.t. true demand distribution) with high probability. For the multi-period case, they develop a sampling-based dynamic programming framework and obtain similar results. Levi et al. [22] also establish a link between first-order information and relative error with respect to optimal value. Samples can then be used to estimate derivatives, or more generally, subgradients. Godfrey and Powell [13] develop a Concave Adaptive Value Estimation (CAVE) algorithm to approximate the value function of a newsvendor problem by successive concave piecewise linear functions. The CAVE algorithm has good performance in numerical experiments, but no convergence result is given in the paper. Powell et al. [31] extend this work and establish convergence results for separable objective functions with integer break points. Huh and Rusmevichientong [17] propose another non-parametric approach to single and multi-period newsvendor problem but only when sales information is available. (This is called censored demand observation). The authors develop an adaptive policy with an average expected cost converging to the newsvendor cost (determined with the knowledge of true demand distribution) at the rate proportional to the square root of the number of periods. Huh et al. [16] considered a similar model and developed new adaptive policies by using Kaplan-Meier estimator [20]. Another alternative in the non-parametric approach is to work with some partial information on the true distribution, e.g., known moments. For the newsvendor problem the mean and the variance of demand can be used to develop a robust min-max policy, see [35, 12, 30] and the references therein for more on this approach. The Bootstrapping method is another “distribution-free” non-parametric approach. Bookbinder and Lordahl [4] use this method to estimate a quantile of lead-time demand distribution to
determine the reorder point for a continuous review inventory system [3].

Besides the inventory models, researchers use sampling methods for stochastic programs, in particular the SAA method. SAA is one of the most popular approximation methods for stochastic programs, replacing the true distribution with an empirical distribution obtained from random samples. Several papers, e.g., [21], [1], [24], [37], [38], [36] (and the references therein), obtain results on convergence and number of samples required for the SAA method to give small relative errors with high probability. Our modeling technique is not stochastic programming, and we make use of the discrete convexity and polynomial time algorithm results of Chapter 2 to solve the SAA counterpart of the appointment scheduling problem. Furthermore, our analysis is non-parametric, we characterize the subdifferential of the objective and use this information explicitly to establish a link between number of samples and the quality of the SAA solution.

Appointment scheduling reduces to the well-known newsvendor problem when there is only a single job. This was first recognized by Weiss [41]. However, the problem departs from newsvendor characteristics and solution methods in the case of multiple jobs ([32], Chapter 2). In the multi-period newsvendor problem, naturally, decisions are taken at each period sequentially. By contrast, in appointment scheduling, one needs to have a schedule before any processing can start, i.e., one determines all the decision variables (i.e., appointment dates) simultaneously at the beginning of the planning horizon (i.e., at time zero).

We employ an SAA approach for the appointment scheduling problem. We use available (independent) samples to form an empirical distribution and find an optimal solution. For the SAA problem, using subdifferential characterization (Section 3.4) and the well-known Hoeffdings inequality [15] we determine number of samples required to guarantee that there will exist a (sufficiently) small (in terms of the specified accuracy level) subgradient at the SAA solution with high probability (i.e., at least the specified confidence level). As a final step we show that the objective value (w.r.t. true distribution) of the SAA solution is no more than \(1 + \) the accuracy level of the true optimal value with probability at least the confidence level. Our bound for number of required samples is polynomial in number of jobs, accuracy level, confidence level and cost coefficients.

To the best of our knowledge this chapter is the first to address the appointment scheduling problem when the probability distributions of durations are unknown. We develop a
sampling-approach for the appointment scheduling problem which is a stochastic non-linear
integer program. Furthermore, we believe this chapter presents the first rigorous analy-
sis for the convexity of the objective function of appointment scheduling problem with a
simple condition. Last but not least, we characterize the set of all subgradients, i.e., the
subdifferential at a given appointment date vector, with a closed-form formula. We use
the subdifferential characterization to relate SAA solution quality with number of samples
required. As a result we relax the perfect information assumption of Chapter 2 on the
probability distribution of processing times. We believe this subdifferential characteriza-
tion will lead to additional applications, e.g., finding optimum appointment schedules by
using non-smooth optimization methods as in Appendix A.

The rest of this chapter is organized as follows. In Section 3.2, we give the formal
description of appointment scheduling problem. We present the convexity results in Sec-
tion 3.3. Section 3.4 contains the subdifferential characterization. We provide our sampling
analysis in Section 3.5. Finally Section 3.6 concludes the paper. We provide all the proofs
either just after their statements or in Section 3.7.

3.2 Formal Description of Appointment Scheduling
Problem

This section closely follows from Chapter 2. There are $n + 1$ jobs numbered $1, 2, \ldots, n + 1$
that need to be sequentially processed (in the order of $1, 2, \ldots, n + 1$) on a single processor.

An appointment schedule needs to be prepared before any processing can start. That is,
each job is assigned a planned start date. In particular, job $i$ will not be available before its
appointment date (planned start date) $A_i$. When a job finishes earlier than the next job’s
appointment date, the system experiences some cost due to under-utilization, i.e., underage
cost. On the other hand, if a job finishes later than the successor job’s appointment date,
the system experiences overage cost due to the overtime of the current job and the waiting
of the next job. The goal is to find appointment dates, $(A_1, \ldots, A_n)$, that minimize the total
cost. In surgery scheduling, determining good surgery start times is crucial. This is not a
trivial task due to randomness in surgery durations. In finding good surgery start times
one needs to consider the tradeoff between idleness and overtime of resource(s) as well as
patients’ waiting times.
If the processing durations were deterministic the problem is straightforward to solve. However, the processing durations are stochastic and we are only given their joint discrete distribution. We assume, naturally, that all cost coefficients and processing durations are non-negative and bounded. We also assume that processing durations are integer valued.\(^3\)

Job 1 starts on-time, i.e., the start time for the first job is zero, and there are \(n\) real jobs. The \((n+1)\)th job is a dummy job with a processing duration of 0. The appointment time for the \((n+1)\)th job is the total time available for the \(n\) real jobs. We use the dummy job to compute the overage or underage cost of the \(n\)th job.

Let \(\{1, 2, 3, \ldots, n, n+1\}\) denote the set of jobs. We denote the random processing duration of job \(i\) by \(p_i\) and the random vector\(^4\) of processing durations by \(\mathbf{p} = (p_1, p_2, \ldots, p_n, 0)\). Let \(\overline{p}_i\) denote the maximum possible value of processing duration \(p_i\), respectively. The maximum of these \(\overline{p}_i\)'s is \(\overline{p}_{\text{max}} = \max(\overline{p}_1, \ldots, \overline{p}_n)\). The underage cost rate \(u_i\) of job \(i\) is the unit cost (unit per time) incurred when job \(i\) is completed at a date \(C_i\) before the appointment date \(A_{i+1}\) of the next job \(i+1\). The overage cost rate \(o_i\) of job \(i\) is the unit cost incurred when job \(i\) is completed at a date \(C_i\) after the appointment date \(A_{i+1}\). Thus the total cost due to job \(i\) completing at date \(C_i\) is

\[
F_i(\mathbf{A}|\mathbf{p}) = o_i(C_i + A_{i+1})^+ + u_i(A_{i+1} + C_i)^+.
\]

The objective to be minimized is the expected total cost \(F(\mathbf{A}) = E_\mathbf{p}[F(\mathbf{A}|\mathbf{p})]\) where

\(^3\)We can restrict ourselves to integer appointment schedules without loss of optimality by Appointment Vector Integrality Theorem 2.5.10.

\(^4\)We write all vectors as row vectors.
the expectation is taken with respect to random processing duration vector \( p \). We simplify notations by defining the lateness \( L_i = C_i - A_{i+1} \) of job \( i \), its tardiness \( T_i = (L_i)^+ \), and its earliness \( E_i = (-L_i)^+ \). The objective \( F(A) \) can now be written as

\[
F(A) = E_p \left[ \sum_{i=1}^{n} (o_i T_i + u_i E_i) \right] = \sum_{i=1}^{n} (o_i E_p T_i + u_i E_p E_i).
\]

The framework can include a given due date \( D \) for the end of processing (e.g., end of day for an operating room) after which overtime is incurred, instead of letting the model choose a planned makespan \( A_{n+1} \). We assume \( D \) is an integer and that \( 0 \leq D \leq \sum_{i=1}^{n} p_i \). Define \( \tilde{A} = (A_1, A_2, ..., A_n) \) then the new objective becomes

\[
F^D(\tilde{A}) = E_p \left[ \sum_{j=1}^{n-1} \left( o_j (C_j - A_{j+1})^+ + u_j (A_{j+1} - C_j)^+ \right) + o_n (C_n - D)^+ + u_n (D - C_n)^+ \right].
\]

We immediately observe that \( F(\tilde{A}, D) = F^D(\tilde{A}) \), and our results in this chapter apply to both objectives (with or without a due date) equally.

### 3.3 Convexity

In this section, we provide a simple sufficient condition so that \( F \) and \( F^D \) are convex as a function of continuous appointment vector. We start with rewriting \( F(\cdot | p) \) in an equivalent form. We will need this in our convexity proof.

**Lemma 3.3.1. (Identity)**

\[
F(A|p) = \sum_{i=1}^{n} \left[ \alpha_i (C_i - A_{i+1}) + \beta_i (C_i - A_{i+1})^+ + \gamma_i (\max\{C_i, A_{i+1}\} - \sum_{k=1}^{i} p_k) \right]
\]

for any \( \alpha_i \in \mathbb{R} \) \((1 \leq i \leq n)\) where \( \beta_i = (o_i - \alpha_i) \) and \( \gamma_i = [(u_i + \alpha_i) - (u_{i+1} + \alpha_{i+1})] \).

**Proof.** By definition (Eq.3.1) we have \( F(A|p) = \sum_{i=1}^{n} [o_i (C_i - A_{i+1})^+ + u_i (C_i - A_{i+1})^+] \). Let \( \alpha_i \in \mathbb{R} \) \((1 \leq i \leq n)\) then by using the identity \( x - (x)^+ + (x)^+ = 0 \) for any \( x \in \mathbb{R} \) we can write

\[
\begin{align*}
o_i (C_i - A_{i+1})^+ + u_i (C_i - A_{i+1})^+ \\
= o_i (C_i - A_{i+1})^+ + u_i (C_i - A_{i+1})^+ + \alpha_i (C_i - A_{i+1}) - \alpha_i (C_i - A_{i+1})^+ + \alpha_i (A_{i+1} - C_i)^+ \\
= o_i (C_i - A_{i+1}) + (\alpha_i - o_i) (C_i - A_{i+1})^+ + (u_i + \alpha_i) (A_{i+1} - C_i)^+.
\end{align*}
\]
Therefore, for any $\alpha_i \in \mathbb{R}$ ($1 \leq i \leq n$) $F(A|p)$ can be written as

$$F(A|p) = \sum_{i=1}^{n} \left[ \alpha_i (C_i - A_{i+1}) + (o_i - \alpha_i)(C_i - A_{i+1})^+ + (u_i + \alpha_i)(A_{i+1} - C_i)^+ \right].$$

Recall that earliness of job $i$ is $E_i = (A_{i+1} - C_i)^+$. Define $M_i$ as the total idle time of jobs $1, 2, ..., j$. Then $E_i = M_i - M_{i-1}$ with $M_0 = 0$, and $F(A|p)$ can be written as

$$F(A|p) = \sum_{i=1}^{n} \left[ \alpha_i (C_i - A_{i+1}) + (o_i - \alpha_i)(C_i - A_{i+1})^+ + (u_i + \alpha_i)(M_i - M_{i-1}) \right]$$

$$= \sum_{i=1}^{n} \left[ \alpha_i (C_i - A_{i+1}) + (o_i - \alpha_i)(C_i - A_{i+1})^+ + [(u_i + \alpha_i) - (u_{i+1} + \alpha_{i+1})]M_i \right].$$

Next, we prove that $M_i = \max\{C_i, A_{i+1}\} - \sum_{t=1}^{i} p_t$ by induction. The result holds for $i = 1$ because $M_1 = \max\{C_1, A_2\} - p_1 = \max\{p_1, A_2\} - p_1 = (A_2 - p_1)^+ = E_1$ (since $S_1 = 0$ we have $C_1 = p_1$). Assume that the result holds for $i = k$, i.e., $M_k = \max\{C_k, A_{k+1}\} - \sum_{t=1}^{k} p_t$.

We need to show it also holds for $i = k + 1$, i.e., $M_{k+1} = \max\{C_{k+1}, A_{k+2}\} - \sum_{t=1}^{k+1} p_t$.

$$M_{k+1} = M_k + E_{k+1} = M_k + (A_{k+2} - C_{k+1})^+$$

(by definition)

$$= \max\{C_k, A_{k+1}\} - \sum_{t=1}^{k} p_t + (A_{k+2} - C_{k+1})^+$$

(by the inductive assumption)

$$= \max\{C_k, A_{k+1}\} + p_{k+1} - \sum_{t=1}^{k+1} p_t + (A_{k+2} - C_{k+1})^+$$

(add and subtract $p_{k+1}$)

$$= C_{k+1} - \sum_{t=1}^{k+1} p_t + (A_{k+2} - C_{k+1})^+$$

($C_{k+1} = \max\{C_k, A_{k+1}\} + p_{k+1}$)

$$= \max\{C_{k+1}, A_{k+2}\} - \sum_{t=1}^{k+1} p_t$$

where the last equality follows by the identity $(x - y)^+ + y = \max\{x, y\}$. Therefore, $M_i = \max\{C_i, A_{i+1}\} - \sum_{t=1}^{i} p_t$ and

$$F(A|p) = \sum_{i=1}^{n} \left[ \alpha_i (C_i - A_{i+1}) + \beta_i (C_i - A_{i+1})^+ + \gamma_i (\max\{C_i, A_{i+1}\} - \sum_{k=1}^{i} p_k) \right]$$

where $\beta_i = (o_i - \alpha_i)$ and $\gamma_i = [(u_i + \alpha_i) - (u_{i+1} + \alpha_{i+1})]$. This completes the proof. \qed

We recall the definition of $\alpha$-monotonicity, Definition 2.6.5. We prove in Proposition 3.3.3 that $\alpha$-monotonicity is a sufficient condition for the convexity of $F$.

**Definition 3.3.2.** The cost coefficients $(u, o)$ are $\alpha$-monotone if there exists reals $\alpha_i$ ($1 \leq i \leq n$) such that $0 \leq \alpha_i \leq o_i$ and the sequence $u_i + \alpha_i$ is non-increasing.
The condition of $\alpha$-monotonicity is satisfied with many reasonable cost structures such as non-increasing $u_i$’s ($u_{i+1} \leq u_i$ for all $i$) or non-increasing $(o_i + u_i)$’s ($o_{i+1} + u_{i+1} \leq o_i + u_i$ for all $i$). Especially the assumption of non-increasing $u_i$’s fits well with many healthcare applications since idle time is usually a bigger concern earlier in a day than later, e.g., if the first patient fails to show up then the surgeon (and other resources) will be idle until the second patient’s appointment date for sure however if a later patient fails to show then it may be the case the surgeon is still busy with previous patient(s) until the next appointment date. Furthermore, non-increasing $u_i$’s captures an important and commonly used special case, uniform idle cost rate for all jobs ($u_i = u$ for all $i$).

**Proposition 3.3.3.** (Convexity) If $(u, o)$ are $\alpha$-monotone then $F(.|p)$ and $F(.)$ are convex.

**Proof.** $F(A|p) = \sum_{i=1}^n \left[ \alpha_i(C_i - A_{i+1}) + \beta_i(C_i - A_{i+1})^+ + \gamma_i(\max\{C_i, A_{i+1}\} - \sum_{k=1}^i p_k) \right]$ where $\beta_i = (o_i - \alpha_i)$ and $\gamma_i = ([u_i + \alpha_i] - \beta_i)]$ by Identity Lemma 3.3.1. We first show that $(C_i - A_{i+1})$, $(C_i - A_{i+1})^+$ and $\max\{C_i, A_{i+1}\} - \sum_{k=1}^i p_k$ are convex in $A$. These functions are convex in $A$ because $C_i$ is convex in $A$. $C_i$ is convex in $A$ since $C_i = \max_{j \leq i} \{A_j + \sum_{k=j}^i p_k\}$ (by the Critical Path Lemma 2.4.1), i.e., $C_i$ is the maximum of convex (affine) functions of $A$ (therefore it is convex). If $(u, o)$ are $\alpha$-monotone then $\alpha_i \geq 0$, $\beta_i \geq 0$, $\gamma_i \geq 0$ ($1 \leq i \leq n$). Since finite sum of convex functions with non-negative weights is convex both $F(.|p)$ and its expectation $F(.)$ are convex. This completes the proof. \[\square\]

**Remark 3.3.4.** $F$ may fail to be convex in the absence of $\alpha$-monotonicity. To see this, consider the following example with two jobs ($n = 2$), deterministic processing times $p_1 > 0$ and $p_2 > 0$, cost coefficients $o_1 = 0$, $u_1 = 0$ and $o_2 > 0$, $u_2 > 0$. Then $F(A) = F(A|p)$ (due to deterministic processing times) and $F(A) = o_2(C_2 - A_3)^+ + u_2(A_3 - C_3)^+$. Let $A' = (0, 0, p_1 + p_2)$ then $S_1 = 0$, $C_1 = S_2 = p_1$ and $C_2 = S_3 = p_1 + p_2$ so $F(A') = 0$. Similarly, let $A'' = (0, 2p_1, 2p_1 + p_2)$ then $S_1 = 0$, $C_1 = p_1$ and $S_2 = 2p_1$, and $C_2 = S_3 = 2p_1 + p_2$ therefore $F(A'') = 0$. Now, we define $A''' = \frac{1}{2}A' + \frac{1}{2}A'' = (0, p_1, \frac{3}{2}p_1 + p_2)$ then $S_1 = 0$, $C_1 = S_2 = p_1$, $C_2 = p_1 + p_2$ and $S_3 = \frac{3}{2}p_1 + p_2$ so $F(A''') = \frac{1}{2}p_1 u_2 > 0$. But this implies that $F(.)$ is not convex since $\frac{1}{2}p_1 u_2 = F(A') = F(\frac{1}{2}A' + \frac{1}{2}A'') > \frac{1}{2}F(A') + \frac{1}{2}F(A'') = 0$.

Similar convexity result holds for $F^D$.

**Corollary 3.3.5.** (Convexity) If $(u, o)$ are $\alpha$-monotone then $F^D(.|p)$ and $F^D(.)$ are convex.
Proof. We substitute $A_{n+1}$ with $D$ in Identity Lemma 3.3.1 and obtain
\[
F^D(\bar{A}|p) = \sum_{i=1}^{n-1} \left[ \alpha_i(C_i - A_{i+1}) + \beta_i(C_i - A_{i+1})^+ + \gamma_i(\max\{C_i, A_{i+1}\} - \sum_{k=1}^{i} p_k) \right] \\
+ \alpha_n(C_n - D) + \beta_n(C_n - D)^+ + \gamma_n(\max\{C_n, D\} - \sum_{k=1}^{n} p_k)
\]
for any $\alpha_i \in \mathbb{R}$ $(1 \leq i \leq n)$ where $\beta_i = (o_i - \alpha_i)$ and $\gamma_i = [(u_i + \alpha_i) - (u_{i+1} + \alpha_{i+1})]$. Then the result follows from Convexity Proposition 3.3.3 because convexity is preserved by projection onto a coordinate subspace.

3.4 Subdifferential Characterization

We start with a definition of a subgradient and subdifferential.

Definition 3.4.1. A vector $g$ is a subgradient of a convex function $f$ at the point $x$ if $f(y) \geq f(x) + g^T(y - x)$ for all $y$. Subdifferential at a point $x$ is the set of all subgradients at the point $x$, i.e., $\partial f(x) = \{g : f(y) \geq f(x) + g^T(y - x)\}$ [14].

We find a subgradient of the objective function $F$ and characterize the set of all subgradients, i.e., subdifferential of $F$, $\partial F$, at any appointment vector $A$. We start with the alternative representation of $F$ given by Identity Lemma 3.3.1 from which we can identify the smaller blocks of $F$: lateness, tardiness and total idle time of jobs 1, 2, ..., $j$ for each job $j$. By using Minkowski sum and subdifferential calculus rules we first obtain subdifferential of these smaller blocks and then again by using these rules we put these smaller subdifferential sets together to characterize the subdifferential of $F$ with a closed form formula. This characterization allows us to link the quality of the sampling solution with the number of independent samples. We also prove that any subgradient of $F(., D)$ is a subgradient for $F^D$ allowing us to extend our results for $F^D$. 

By Identity Lemma 3.3.1,
\[
F(A|p) = \sum_{j=1}^{n} \left[ \alpha_j(C_j - A_{j+1}) + \beta_j(C_j - A_{j+1})^+ + \gamma_j(\max\{C_j, A_{j+1}\} - \sum_{k=1}^{j} p_k) \right]
\]
for any $\alpha_i \in \mathbb{R}$ $(1 \leq i \leq n)$ where $\beta_i = (o_i - \alpha_i)$ and $\gamma_i = [(u_i + \alpha_i) - (u_{i+1} + \alpha_{i+1})]$ ($\gamma_n = [(u_n + \alpha_n)]$). We assume $\alpha$-monotone $(u, o)$ (by the Convexity Proposition 3.3.3) for the convexity of $F(., p)$ and $F(.,.)$. Recall that $L_j(A|p) = (C_j - A_{j+1})$ (lateness), $T_j(A|p) = (C_j - A_{j+1})^+$ (tardiness) and $M_j(A|p) = \max\{C_j, A_{j+1}\} - \sum_{k=1}^{j} p_k$ (total idle time of jobs 1, 2, ..., $j$). Here we use $(A|p)$ for $L_j, T_j$ and $M_j$ to emphasize the fact
that these quantities are for a given (particular) $p$, hence they are deterministic. Similarly, we will use $(A)$ to denote their expected values, i.e., $L_j(A), T_j(A)$ and $M_j(A)$. We can rewrite $F(A|p)$ as $\sum_{j=1}^{n} (\alpha_j L_j(A|p) + \beta_j T_j(A|p) + \gamma_j M_j(A|p))$ where $\alpha_j, \beta_j, \gamma_j \geq 0$ by α-monotonicity. To characterize the subdifferential $\partial F(A)$, we first derive the subdifferentials of $L_j(A|p), T_j(A|p)$ and $M_j(A|p)$. Then, we find $\partial L_j(A), \partial T_j(A)$ and $\partial M_j(A)$ where $\zeta_j(A) = E_p[\zeta_j(A|p)]$ for $\zeta \in \{L, T, M\}$. After that we obtain the subdifferential of $F_j(A)$ (by Eq(3.2) below) and $\partial(\sum_{j=1}^{n} F_j(A)) = \partial F(A)$.

$$F_j(A) = \alpha_j L_j(A) + \beta_j T_j(A) + \gamma_j M_j(A) \quad (3.2)$$

We start the analysis with the definition of Minkowski sum (e.g., see [33]) since the sums in our subdifferential derivations are Minkowski sums. The Minkowski sum of sets $X_1$ and $X_2$ is defined as

$$X_1 + X_2 = \{x = x_1 + x_2 : x_1 \in X_1, x_2 \in X_2\}.$$ 

More generally, if $I$ is a finite set, $r_i$ is a real number and $X_i$ is a set ($i \in I$) then the Minkowski sum $\sum_{i \in I} r_i X_i$ is $\sum_{i \in I} r_i X_i = \{x = \sum_{i \in I} r_ix_i : x_i \in X_i \text{ for all } i\}$. In particular, for any real $r \in \mathbb{R}$, $rX = \{rx : x \in X\}$. We will use two particular subdifferential calculus rules in our derivations. Rule 1 and Rule 2 follow from Theorem 4.1.1 and Corollary 4.4.4 of [14] respectively. Let $f, f_1, f_2, ..., f_m$ be finite convex functions from $\mathbb{R}^n$ to $\mathbb{R}$, $I$ be a finite set and $r_i$ ($i \in I$) be a non-negative real number. Then the rules are:

Rule 1: $\partial(\sum_{i \in I} r_i f_i) = \sum_{i \in I} r_i \partial f_i \quad (3.3)$

Rule 2: If $f = \max_{1 \leq i \leq m} f_i$ and all $f_i$’s are differentiable then $\partial f = \text{co}\{\nabla f_i : f_i = f\}$

where $\text{co}$ stands for convex hull and the summation on the right side of the equation in Rule 1 is a Minkowski sum.

Lemma 3.4.2 allows us to consider $\partial \Psi(A|p)$ in finding $\partial \Psi(A)$ where $\Psi \in \{L_j, T_j, M_j, F, F_j\}$. We use the notations $\partial \Psi(A) = E_p[\partial \Psi(A|p)]$ and $\text{Prob}(p)$ to represent the probability of realization $p$. Since $\Psi(.)$ is, in all these cases, finite and convex, we have the following result.

**Lemma 3.4.2.** The relation $\partial(E_p[\Psi(A|p)]) = E_{p}[\partial \Psi(A|p)]$ holds where

$$E_p[\partial \Psi(A|p)] = \sum_{p} \text{Prob}(p) \partial \Psi(A|p) = \{s \in \mathbb{R}^{n+1} : \exists s^p \in \partial \Psi(A|p) \forall p \text{ and } s = \sum_{p} \text{Prob}(p)s^p\}.$$
Proof. $\Psi(A|p)$ is finite and convex everywhere for any $p$ and there are finitely many realizations of $p$. Therefore by Rule 1 and Rule 2 we obtain the claimed result as follows:

$$
\partial(E_p[\Psi(A|p)]) = \partial(\sum_p \text{Prob}(p)\Psi(A|p)) = \sum_p \text{Prob}(p)\partial\Psi(A|p) = E_p[\partial\Psi(A|p)].
$$

Lemma 3.4.2 is useful as it allows us to work with $\partial\Psi(A|p)$ for any $\Psi \in \{L_j, T_j, M_j, F_j\}$ and obtain $\partial\Psi(A)$ by taking its expectation. By Eq(3.2) and Lemma 3.4.2,

$$
\partial F(A) = \sum_{j=1}^{n} [\alpha_j E_p[\partial L_j(A|p)] + \beta_j E_p[\partial T_j(A|p)] + \gamma_j E_p[\partial M_j(A|p)]]
$$

where, as before, all sums are Minkowski sums. So, once we find $\partial$ and obtain $\partial^j$ and $\partial^k$, we now obtain $\partial F(A)$. Consider $L_j(A|p) = (C_j - A_{j+1})$. By Critical Path Lemma 2.4.1, $C_j = \max_{k \leq j} \{A_k + \sum_{t=k}^{j} p_t\}$ so $L_j(A|p) = \max_{k \leq j} \{A_k + \sum_{t=k}^{j} p_t - A_{j+1}\}$.

Recall that the notation $\{.,p\}$ is used to emphasize the fact that the quantity of interest is deterministic once the job duration vector $p$ is given. In order to find $\partial L_j(A|p)$, we need to know which $k$’s ($k \leq j$) maximize $\{A_k + P_{kj}\}$ where $P_{kj} = \sum_{t=k}^{j} p_t$. To represent the set of such maximizers for job $j$ we define

$$
I_{j} = \arg \max_{k \leq j} \{A_k + P_{kj}\}.
$$

A remark is here in order. Note that $I_{j}$ depends on $A$ and $p$, and it is deterministic for any given (particular realization of) $p$ and a random variable otherwise. Let $1_i$ denote a unit vector in $\mathbb{R}^{n+1}$ where the $i^{th}$ component is 1 and all other components are 0.

Then $L_j(A|p) = \max_{k \leq j} \{A_k + P_{kj} - A_{j+1}\}$ and by Rule 2 we obtain

$$
\partial L_j(A|p) = \text{co}\{1_k - 1_{j+1} : k \in I_j\}.
$$

Similar to $\partial L_j(A|p)$, we now obtain $\partial T_j(A|p)$. In addition to $I_{j}$, we also need the sign of $\max_{k \leq j} \{A_k + P_{kj}\} - A_{j+1}$ since $\partial T_j(A|p) = (L_j(A|p))^+$. Let

$$
I_j^\varrho = \{k \in I_j : A_k + P_{kj} \varrho A_{j+1}\} \text{ where the relation } \varrho \in \{>,<,=\}.
$$

By extending $\partial L_j(A|p)$ with the sign of $\max_{k \leq j} \{A_k + P_{kj}\} - A_{j+1}$ we obtain $\partial T_j(A|p)$:

$$
\partial T_j(A|p) = \begin{cases} 
\text{co}\{1_k - 1_{j+1} : k \in I_j^\varrho\} & \text{if } \max_{k \leq j} A_k + P_{kj} > A_{j+1} \\
\text{co}\{(0) \cup \{1_k - 1_{j+1} : k \in I_j^\varrho\}\} & \text{if } \max_{k \leq j} A_k + P_{kj} = A_{j+1} \\
\{0\} & \text{if } \max_{k \leq j} A_k + P_{kj} < A_{j+1}
\end{cases}
$$

65
We note that exactly two of the sets $I_j^\uparrow, I_j^\downarrow, I_j^\leftarrow$ are empty. This allows us to represent $\partial T_j(A|p)$ as

$$\partial T_j(A|p) = \text{co}(\{0\} \cup \{1_k - 1_{j+1} : k \in I_j^\uparrow\}) + \text{co}(\{0\} \cup \{1_k - 1_{j+1} : k \in I_j^\downarrow\}). \quad (3.7)$$

Next, we obtain $\partial M_j(A|p)$. Recall that $M_j(A|p) = \max\{C_j, A_{j+1}\} - P_{1j}$ and $C_j = \max_{k \leq j} \{A_k + P_{kj}\}$, then $M_j(A|p) = \max \{\max_{k \leq j} \{A_k + P_{kj}\}, A_{j+1}\} - P_{1j}$. By using Rule 2 (similarly to $\partial T_j(A|p)$), we obtain $\partial M_j(A|p)$.

$$\partial M_j(A|p) = \begin{cases} 
\text{co}(1_k : k \in I_j^\uparrow) & \text{if } \max_{k \leq j} A_k + P_{kj} > A_{j+1} \\
\text{co}(\{1_{j+1}\} \cup \{1_k : k \in I_j^\downarrow\}) & \text{if } \max_{k \leq j} A_k + P_{kj} = A_{j+1} \\
1_{j+1} & \text{if } \max_{k \leq j} A_k + P_{kj} < A_{j+1}.
\end{cases}$$

Since exactly two of the sets $I_j^\uparrow, I_j^\downarrow, I_j^\leftarrow$ are empty, we can represent $\partial M_j(A|p)$ compactly as

$$\partial M_j(A|p) = \text{co}(\{0\} \cup \{1_k - 1_{j+1} : k \in I_j^\uparrow\}) + \text{co}(\{1_{j+1}\} \cup \{1_k : k \in I_j^\downarrow\}). \quad (3.8)$$

We now can obtain $\partial L_j(A)$, $\partial T_j(A)$ and $\partial M_j(A)$ starting with $\partial L_j(A)$. Recall that by Eq(3.5), $L_j(A|p) = \text{co}(1_k - 1_{j+1} : k \in I_j)$ and by Lemma 3.4.2, we have $\partial L_j(A) = E_p[\partial L_j(A|p)] = \sum_p \text{Prob}(p) \partial L_j(A|p)$. Therefore,

$$\partial L_j(A) = \sum_p \text{Prob}(p) \text{co}(1_k - 1_{j+1} : k \in I_j). \quad (3.9)$$

There are potentially $\overline{p}_\max^m$ many realizations of $p$, and this number may be very large. However, we observe that all the vectors appearing in the convex hull of $L_j(A|p)$ are $(1_k - 1_{j+1})$ for some $k \leq j$. Therefore the convex hull of $L_j(A|p)$ will be a convex combination of the vectors in some subset of $\{(1_1 - 1_{j+1}), (1_2 - 1_{j+1}), ..., (1_j - 1_{j+1})\}$. The following result makes this observation precise.

**Lemma 3.4.3.** Let $r_1, r_2, ..., r_m \geq 0$ be reals. If $X$ is a convex set and then $\sum_{i=1}^m (r_i X) = (\sum_{i=1}^m r_i) X$.

**Remark 3.4.4.** The convexity of $X$ is essential, to see this take the non-convex set $X = \{0, 1\}$ with $r_1 = r_2 = 1$ then $\sum_{i=1}^2 (r_i X) = \{0, 1, 2\} \neq \{0, 2\} = (\sum_{i=1}^2 r_i) X$.

Lemma 3.4.3 enables us to combine all realizations giving the same convex hull $X$, i.e., instead of considering all possible realizations we consider the non-empty subsets of
\{(1_1 - 1_{j+1}), (1_2 - 1_{j+1}), \ldots, (1_j - 1_{j+1})\}. We define \([j] = \{1, 2, \ldots, j - 1, j\}\) and use \(\mathcal{P}^*([j])\) to denote all the non-empty subsets of \([j]\). For any \(S \in \mathcal{P}^*([j])\), let \(\text{Prob}\{\Phi = S\} = \sum_{p: \Phi = S} \text{Prob}\{p\}\) for \(\Phi \in \{I_j, \bar{I}_j, \bar{I}_j\}\) and \(j = 1, \ldots, n\). Then the next Lemma shows how to obtain \(\partial L_j(A)\).

**Lemma 3.4.5.** The subdifferential \(\partial L_j(A)\) is given by

\[
\sum_{S \in \mathcal{P}^*([j])} \text{Prob}\{I_j = S\} \text{co}\{(1_k - 1_{j+1} : k \in S)\}.
\]

**Proof.** Eq(3.9) gives \(\partial L_j(A) = \sum_p \text{Prob}\{p\} \text{co}\{1_k - 1_{j+1} : k \in I_j\}\). We obtain the desired result by the equalities below:

\[
\sum_p \text{Prob}\{p\} \text{co}\{1_k - 1_{j+1} : k \in I_j\} = \sum_{S \in \mathcal{P}^*([j])} \text{Prob}\{I_j = S\} \text{co}\{1_k - 1_{j+1} : k \in S\}
\]

where \(\mathbb{1}\{I_j = S\}\) is 1 if \(\{I_j = S\}\) and 0 otherwise (i.e., \(\mathbb{1}\) is the indicator function), and the last equality follows from the definition of \(\text{Prob}\{I_j = S\}\) and Lemma 3.4.3. \(\square\)

We obtain similar results for \(\partial T_j(A)\) and \(\partial M_j(A)\) in the next two lemmata.

**Lemma 3.4.6.** The subdifferential \(\partial T_j(A)\) is given by

\[
\sum_{S \in \mathcal{P}^*([j])} [\text{Prob}\{I_j = S\} \text{co}\{1_k - 1_{j+1} : k \in S\} + \text{Prob}\{I_j = \bar{S}\} \text{co}\{(0) \cup \{1_k - 1_{j+1} : k \in S\}\}].
\]

**Lemma 3.4.7.** The subdifferential \(\partial M_j(A)\) is given by

\[
\sum_{S \in \mathcal{P}^*([j])} \left[\text{Prob}\{I_j = S\} \text{co}\{1_k - 1_{j+1} : k \in S\} + \text{Prob}\{I_j = \bar{S}\} \text{co}\{1_k : k \in S \cup \{j + 1\}\}\right] + \left(1 - \sum_{S \in \mathcal{P}^*([j])} \text{Prob}\{I_j = S\}\right) 1_{j+1}.
\]
For later purposes we represent the convex hulls in $\partial L_j(A)$, $T_j(A)$ and $M_j(A)$ in a different form. By using Lemma 3.4.5 above we may express $\partial L_j(A)$ as

$$\partial L_j(A) = \left\{ \sum_{s \in P^*(\{j\})} \text{Prob}[I_j = S] \sum_{k \in S} (1_k - 1_{j+1})X_{kj}^L(S) : \right.$$

$$\left. \sum_{k \in S} X_{kj}^L(S) = 1 \ \forall S \in P^*(\{j\}), \ X_{kj}^L(S) \geq 0 \ \forall S \in P^*(\{j\}) \ \forall k \in S \right\}.$$  (3.10)

where $X_{kj}^L(S)$ is the non-negative variable representing the weight of the term $(1_k - 1_{j+1})$ in a convex combination determining an element of $\text{co}\{(1_k - 1_{j+1}) : k \in S\}$.

Similarly to Eq(3.10) and Eq(3.11), by using Lemma 3.4.6 we obtain:

$$\partial T_j(A) = \left\{ \sum_{s \in P^*(\{j\})} \text{Prob}[I_j = S] \sum_{k \in S} (1_k - 1_{j+1})X_{kj}^T>(S) :$$

$$+ \sum_{k \in S} X_{kj}^T>(S) = 1 \ \forall S \in P^*(\{j\}), \ \sum_{k \in S} X_{kj}^T>(S) \leq 1 \ \forall S \in P^*(\{j\})$$

$$X_{kj}^T>(S), X_{kj}^T<(S) \geq 0 \ \forall S \in P^*(\{j\}) \ \forall k \in S \right\}.$$  (3.11)

where $X_{kj}^T>(S)$ and $X_{kj}^T<(S)$ are non-negative variables representing the weight of the terms $(1_k - 1_{j+1})$ in a convex combination determining an element of $\text{co}\{(1_k - 1_{j+1}) : k \in S\}$ and $\text{co}\{(0) \cup (1_k - 1_{j+1}) : k \in S\}$ respectively. Note that the convexity constraint in the second line of $\partial T_j(A|p)$, $\sum_{k \in S} X_{kj}^T<(S) \leq 1$, is an inequality since 0 may be a subgradient.

Similarly to Eq(3.10) and Eq(3.11), by using Lemma 3.4.7 we express $\partial M_j(A)$ as the following.

$$\partial M_j(A) = \left\{ \sum_{s \in P^*(\{j\})} \left( \text{Prob}[I_j = S] \sum_{k \in S} (1_k - 1_{j+1})X_{kj}^M>(S) + \right.$$

$$\sum_{k \in S \cup \{j+1\}} (1_k)X_{kj}^M=(S \cup \{j+1\}) \right) :$$

$$\left. \left(1 - \sum_{s \in P^*(\{j\})} \text{Prob}[I_j = S] \right)1_{j+1} : \
$$

$$\sum_{k \in S} X_{kj}^M>(S) = 1 \ \forall S \in P^*(\{j\}), \ X_{kj}^M>(S) \geq 0 \ \forall S \in P^*(\{j\}) \ \forall k \in S,$$

$$\sum_{k \in S \cup \{j+1\}} X_{kj}^M=(S \cup \{j+1\}) = 1 \ \forall S \in P^*(\{j\}), \ X_{kj}^M=(S) \geq 0 \ \forall S \in P^*(\{j\}) \ \forall k \in S,$$

$$X_{kj}^M=(S \cup \{j+1\}) \geq 0 \ \forall S \in P^*(\{j\}) \ \forall k \in S \cup \{j+1\} \right\}.$$  (3.12)

where $X_{kj}^M>(M)$ and $X_{kj}^M(S)$ are non-negative variables representing the weight of the terms $(1_k - 1_{j+1})$ and $(1_k)$ in a convex combination determining an element of $\text{co}\{(1_k - 1_{j+1}) : k \in S\}$ and $\text{co}\{(1_k) : k \in S \cup \{j+1\}\}$ respectively.
For clarity, we collect the variables $X_{ij}^L(S), X_{ij}^{T>}(S), X_{ij}^{T=}(S), X_{ij}^M>(S), X_{ij}^M=(S)$ into a single vector $X^j$, and express the feasible set $\Theta^j$ of $X^j$ in a compact form.

$$X^j = ((X_{ij}^v(S)), (X_{kl}^M=(S \cup \{l + 1\})): v \in \{L, T>, T=, M>\}, 1 \leq i \leq j \leq n + 1,$$

$$1 \leq k < l \leq n + 1, \ S \in P^*([j]), \ i \in S, \ k \in S \cup \{l + 1\}).$$

$$\Theta^j = \left\{ X^j \geq 0: \sum_{i \in S} X_{ij}^v(S) = 1, \sum_{i \in S} X_{ij}^{T=}(S) \leq 1, \sum_{k \in S \cup \{l + 1\}} X_{kl}^M=(S \cup \{l + 1\}) = 1, \forall v \in \{L, T>, M>\}, \forall S \in P^*([j]), \forall i \in S, \forall k \in S \cup \{l + 1\}\right\}.$$  

We next collect all $X^j$ vectors into a single vector $X$ and express the feasible set $\Theta$ of $X$:

$$X = (X^j)_{j \in [n+1]}$$

$$\Theta = \times_{j \in [n+1]} \Theta^j = \{ X = (X^j)_{j \in [n+1]} : X^j \in \Theta^j \ \forall j \in [n + 1] \}. \quad (3.13)$$

Now we obtain $\partial F(A)$.

**Proposition 3.4.8.** We may express $\partial F(A)$ in a closed-form formula given by Eq(3.15).

*Proof.* Since $\alpha_j, \beta_j, \gamma_j \geq 0 \ (j = 1, ..., n + 1)$, by Rule 1 and Eq(3.2) we get $\partial F_j(A) = \alpha_j \partial L_j(A) + \beta_j \partial T_j(A) + \gamma_j \partial M_j(A)$. We obtain $\partial F(A)$ as the Minkowski sum of $F_j(A)$’s ($j = 1, ..., n + 1$).

$$\partial F(A) = \sum_{j=1}^{n} \partial F_j(A) = \sum_{j=1}^{n} \left[ \alpha_j \partial L_j(A) + \beta_j \partial T_j(A) + \gamma_j \partial M_j(A) \right]. \quad (3.14)$$

We gather the values of $\partial L_j(A), \partial T_j(A)$, and $\partial M_j(A)$ from Eq(3.10), Eq(3.11) and Eq(3.12) respectively and use Eq(3.14) to obtain the closed-form expression:

$$\partial F(A) = \left\{ \sum_{j=1}^{n} \alpha_j \sum_{S \in P^*([j])} \text{Prob}\{I_j = S\} \left( \sum_{i \in S} (1_i) X_{ij}^L(S) - 1_{j+1} \right) + \sum_{j=1}^{n} \beta_j \sum_{S \in P^*([j])} \text{Prob}\{I_j > = S\} \left( \sum_{i \in S} (1_i) X_{ij}^{T=}(S) - 1_{j+1} \right) + \sum_{j=1}^{n} \beta_j \sum_{S \in P^*([j])} \text{Prob}\{I_j ^> = S\} \left( \sum_{i \in S} (1_i) X_{ij}^{T>}(S) - 1_{j+1} \right) + \sum_{j=1}^{n} \gamma_j \sum_{S \in P^*([j])} \text{Prob}\{I_j ^> = S\} \left( \sum_{i \in S} (1_i) X_{ij}^M>(S) - 1_{j+1} \right) + \sum_{j=1}^{n} \gamma_j \sum_{S \in P^*([j])} \text{Prob}\{I_j ^> = S\} \sum_{i \in S \cup \{j + 1\}} (1_i) X_{ij}^M>(S \cup \{j + 1\}) + \sum_{j=1}^{n} \gamma_j (1 - \sum_{S \in P^*([j])} \text{Prob}\{I_j ^> = S\})1_{j+1} : X \in \Theta \right\}. \quad (3.15)$$
We next express \( \partial F(\mathbf{A}) \) component by component for a particular \( \mathbf{X} \in \Theta \), i.e., a coordinate of a subgradient at the point \( \mathbf{A} \) for a particular \( \mathbf{X} \in \Theta \). Let \( g(\mathbf{X}, \mathbf{A}) \) be the element of \( \partial F(\mathbf{A}) \) defined by the vector \( \mathbf{X} \). Then \( g(\mathbf{X}, \mathbf{A}) = (g_1(\mathbf{X}, \mathbf{A}), g_2(\mathbf{X}, \mathbf{A}), \ldots, g_{n+1}(\mathbf{X}, \mathbf{A})) \) where \( g_k(\mathbf{X}, \mathbf{A}) \) is the \( k^{th} \) component of \( g(\mathbf{X}, \mathbf{A}) \). Corollary 3.4.9 gives an expression for \( g_k(\mathbf{X}, \mathbf{A}) \).

**Corollary 3.4.9.** We may express \( g(\mathbf{X}, \mathbf{A}) \) in a closed-form formula given by Eq(3.16).

**Proof.** We observe that all vectors appearing in the convex combination defining \( \partial F(\mathbf{A}) \) are \((1_1 - 1_{j+1})\) for some \( 1 \leq i < j + 1 \leq n + 1 \) and \((1_1)\) for some \( 1 \leq i \leq n + 1 \). Therefore the \( k^{th} \) component \( g_k \) of \( g \in \partial F(\mathbf{A}) \) may only get nonzero contributions from vectors \((1_k - 1_{j+1})\) for all \( j + 1 > k \), vectors \((1_1 - 1_k)\) for all \( i < k \) and vector \((1_k)\). To see this from a different perspective, consider \( A_k \): \( A_k \) appears in the terms \((C_{k-1} - A_k), (C_{k-1} - A_k)^{+}\) and may appear in \( \max\{C_{k-1}, A_k\} - \sum_{i=1}^{k-1} p_i, (C_{j-1} - A_j) \) and \((C_{j-1} - A_j)^{+}\) for \( j > k \). We derive \( g_k \) by using Eq(3.15). Let \( \mathbf{X} \in \Theta \), then the \( k^{th} \) component of \( g(\mathbf{X}, \mathbf{A}) \), namely, \( g_k(\mathbf{X}, \mathbf{A}) \), is

\[
\begin{align*}
\sum_{j=k}^{n} \alpha_j & \sum_{S \in P^*(\{j\})} \text{Prob}\{I_j = S\} \ X_{kj}^L(S) - \alpha_{k-1} \sum_{S \in P^*(\{k-1\})} \text{Prob}\{I_{k-1} = S\} \\
+ \sum_{j=k}^{n} \beta_j & \sum_{S \in P^*(\{j\})} \text{Prob}\{I_j^+ = S\} X_{kj}^{T^+}(S) - \beta_{k-1} \sum_{S \in P^*(\{k-1\})} \text{Prob}\{I_{k-1} = S\} \\
+ \sum_{j=k}^{n} \beta_j & \sum_{S \in P^*(\{j\})} \text{Prob}\{I_j^- = S\} X_{kj}^{T^-}(S) - \beta_{k-1} \sum_{S \in P^*(\{k-1\})} \text{Prob}\{I_{k-1} = S\} \sum_{i \in S} X_{i,k-1}^T(S) \\
+ \sum_{j=k}^{n} \gamma_j & \sum_{S \in P^*(\{j\})} \text{Prob}\{I_j^\gamma = S\} X_{kj}^T(S) - \gamma_{k-1} \sum_{S \in P^*(\{k-1\})} \text{Prob}\{I_{k-1} = S\} \\
+ \sum_{j=k}^{n} \gamma_j & \sum_{S \in P^*(\{j\})} \text{Prob}\{I_j^- = S\} X_{kj}^{M^+}(S \cup \{j + 1\}) \\
+ \gamma_{k-1} (1 - \sum_{S \in P^*(\{k-1\})} \text{Prob}\{I_{k-1} = S\}) 
\end{align*}
\] (3.16)

**Remark 3.4.10.** Note that \( \sum_{S \in P^*(\{k-1\})} \text{Prob}\{I_{k-1} = S\} = 1 \). For our analysis in this chapter, we do not require the values of the probabilities \( \text{Prob}\{I_j = S\} \), \( \text{Prob}\{I_j^\gamma = S\} \) and \( \text{Prob}\{I_j^- = S\} \) (for \( S \in P^*(\{j\}) \) and \( j \in [n+1] \)). However these values may be needed for other research, and indeed these probabilities are computed and used in Appendix A.

Subgradients for \( F^D \). Proposition 3.4.11 allows us to use any subgradient of \( F(\mathbf{\tilde{A}}, D) \) for \( F^D(\mathbf{\tilde{A}}) \).
Proposition 3.4.11. \( \text{proj}(\partial F(\tilde{A}, D)) \subseteq \partial F^D(\tilde{A}) \) where \( \text{proj} \) is the projection given as \( \text{proj}(x_1, x_2, ..., x_n, x_{n+1}) = (x_1, x_2, ..., x_n) \).

Therefore subgradient \( \text{proj}(g(\tilde{A}, D)) \) is a subgradient for \( F^D(\tilde{A}) \) and hence we can extend our results to \( F^D \).

Remark 3.4.12. One may wish to find a minimum norm subgradient at a point \( A \) as it provides an optimality test (a point \( A^* \) is optimal if and only if \( 0 \in \partial F(A^*) \)) and also the negative minimum norm subgradient is a descent direction (e.g., see [2]). By Eq(3.16) the minimum norm subgradient may be computed with a linear program (LP) in \( l_1 \) norm and as a quadratic program (QP) in \( l_2 \) norm.

\[
\begin{align*}
\text{LP} & \quad \text{min} \sum_{k=1}^{n+1} z_k \\
& \quad \text{subject to} \\
& \quad z_k \geq g_k(X, A) \quad (1 \leq k \leq n+1) \\
& \quad z_k \geq -g_k(X, A) \quad (1 \leq k \leq n+1) \\
& \quad X \in \Theta
\end{align*}
\]

Decision variable \( z_i \) is used to represent the absolute value \( |g_i| \) in the \( l_1 \) norm. Now, we give the QP formulation.

\[
\begin{align*}
\text{QP} & \quad \text{min} \sum_{k=1}^{n+1} g_k^2(X, A) \\
& \quad \text{subject to} \\
& \quad X \in \Theta
\end{align*}
\]

This QP has linear constraints but a quadratic objective function.

3.5 Sampling Approach

In this section, we relax the perfect information assumption of job durations distribution in Chapter 2. We assume that there exists an underlying (true) discrete joint distribution for the job durations but the distribution is not known. Instead there is a set of independent samples available. For example, in many practical scenarios one has daily historical data on surgery durations. Job durations may not necessarily be independent but samples are. (Each sample is a vector of all job durations on surgeries.) We develop a sampling-based approach to determine the number of independent samples required to obtain a provably near-optimal solution with high probability. That is, with high probability the cost (w.r.t.
the true distribution) of the sampling-based schedule is no more than \((1 + \epsilon)\) times the cost of optimal schedule that is computed based on the true distribution.

Let \(\epsilon\) be the accuracy level, \(1 - \delta\) the confidence level and \(N = N(\epsilon, \delta, u, o)\) the number of samples. Define \(p^k = (p^k_1, p^k_2, ..., p^k_n)\) as the \(k^{th}\) observation of \(N\) samples. We use \(\sim\) to denote quantities obtained from samples. Let \(\hat{p} = \hat{p}(N)\) be the empirical joint probability distribution obtained from \(N\) independent observations of \(p\), i.e., \(\text{Prob}\{\hat{p} = p^k\} = \frac{1}{N}\) for \(1 \leq k \leq N\). We denote a true optimal appointment vector with \(A^*\), i.e., \(A^*\) is a minimizer of \(F_p(A) = E_p(F(A|p))\). We use the subscript \(p\) emphasize the fact that the quantities are obtained with respect to the true distribution \(p\). Similarly, let \(\tilde{A} = \tilde{A}(N)\) be a minimizer of \(F_p(A) = E_{\hat{p}}(F(A|\hat{p}))\). Again we use the subscript \(\hat{p}\) to emphasize the fact that the quantities are obtained with respect to the sampling distribution \(\hat{p}\). For subgradients, we express their \(k^{th}\) component as \(g_k(X, A)_p\) for \(F_p(\cdot)\) and \(g_k(X, A)_{\hat{p}}\) for \(F_{\hat{p}}(\cdot)\) at the point \(A\).

We start our analysis by proving that we can minimize \(F_{\hat{p}}(\cdot)\) (and \(F^D_{\hat{p}}(\cdot)\)) in polynomial time. This follows from Theorem 2.7.1 (and Corollary 2.8.6) of Chapter 2. Then, with an application of Hoeffdings’ inequality, we establish a connection between the probability of an event with respect to \(p\) and \(\hat{p}\) as a function of sample size \(N\) for a given accuracy level \(\epsilon'\) (absolute difference of the probabilities w.r.t. \(p\) and \(\hat{p}\)) and a confidence level \(1 - \delta'\).

After that we provide a similar result for a family of events \(F\) to hold simultaneously. Our next result uses subdifferential characterization to show the existence of a \(g \in \partial F_p(\tilde{A})\) such that \(|g_k| < \epsilon' K'\) with high probability where \(K' = K'(n, u, o)\) is some constant. Then we prove that if there exists \(g \in \partial F_p(\tilde{A})\) such that \(|g_k| < \epsilon\nu/3(n + 1)n\) for all \(1 \leq k \leq n + 1\) then \(F_p(\tilde{A}) \leq (1 + \epsilon)F_p(A^*)\) where \(0 < \epsilon \leq 1\) and \(\nu = \min\{u_1, u_2, ..., u_n, o_1, o_2, ..., o_n\}\). This is achieved with an application of Jensen inequality and a version of Lemma 5.1 of [22] (Lemma 3.7.2). We conclude by stating our main result which determines the number of samples required to achieve \((1 + \epsilon)\) approximation with probability at least \(1 - \delta\).

**Corollary 3.5.1. (Polynomial Time Algorithm)** If the cost vectors \((u, o)\) are \(\alpha\)-monotone and the processing durations are integer then \(F_{\hat{p}}(\cdot)\) (and \(F^D_{\hat{p}}(\cdot)\)) can be minimized in \(O(n^5 Nn n^2 \log([p_{\text{max}}/2]))\) time.

**Proof.** Theorem 2.7.1 implies that \(F_{\hat{p}}(\cdot)\) can be minimized in \(O(\sigma(n) EO n^2 \log([h/2n]))\) where \(\sigma(n)\) is the number of function evaluations required to minimize a submodular set function over an \(n\)-element ground set and \(EO\) is the time needed for an expected cost.
evaluation. We find the expected cost for $F_{\hat{p}}(.)$ in $O(nN)$ by computing the total cost for each realization (takes $O(n)$ time) and then take the average of $N$ total cost realizations, i.e., sample average approximation. Finally, Theorem 4 of [29] shows that $\sigma(n) = O(n^5)$. The result for $F_{\hat{D}}(.)$ follows similarly from Corollary 2.8.6.

Polynomial Time Algorithm Corollary 3.5.1 shows that for a given $N$ samples of job durations we can solve the SAA counterpart of the appointment scheduling problem efficiently. The remaining task is to find the sufficient number of samples $N$ (for a given accuracy level and confidence level) such that the SAA optimal solution (w.r.t. the true distribution) will have a cost no more than $(1+\text{the accuracy level})$ times optimal cost with probability at least the confidence level.

Let $O$ be any event depending on the processing times $p = (p_1, p_2, ..., p_n)$, $O = O(p_1, p_2, ..., p_n) = O(p)$. Let $\text{Prob}_p\{O(p)\}$ denote the true probability of $O$. Let $\text{Prob}_{\hat{p}}\{O(p)\}$ denote an estimate of $\text{Prob}_p\{O(p)\}$ when true distribution of $p$ is not known, and the empirical probability distribution $\hat{p}$, based on $N$ independent samples, is used in the estimation. We define an indicator function as

$$\mathbb{I}\{O(p^k)\} = \begin{cases} 1 & \text{if event } O \text{ occurs with realization } p^k \\ 0 & \text{otherwise} \end{cases}$$

then $\mathbb{I}\{O(p^k)\}$ is Bernoulli distributed with parameter $\text{Prob}_p\{O(p)\}$. We define our estimate $\text{Prob}_{\hat{p}}\{O(p)\}$ as

$$\text{Prob}_{\hat{p}}\{O(p)\} = \frac{1}{N} \sum_{k=1}^{N} \mathbb{I}\{O(p^k)\}.$$ 

**Remark 3.5.2.** Note that $N\text{Prob}_{\hat{p}}\{O(p)\}$ is the sum of $N$ independent Bernoulli random variables with parameter $\text{Prob}_p\{O(p)\}$, therefore $N\text{Prob}_{\hat{p}}\{O(p)\}$ is binomially distributed with parameters $\text{Prob}_p\{O(p)\}$ and $N$.

We use Hoeffdings’ inequality to obtain the number of samples $N$ required such that

$$\text{Prob}\{ |\text{Prob}_p\{O(p)\} - \text{Prob}_{\hat{p}}\{O(p)\}| \leq \varepsilon' \} > 1 - \delta'$$

for any given accuracy level $\varepsilon' > 0$ and confidence level $0 < \delta' < 1$. Direct application of Hoeffdings’ inequality for Bernoulli random variables (Theorem 4.5 in [40]) yields

$$N > \frac{1}{2} (\varepsilon')^2 \ln(2/\delta').$$

Using union bounds we obtain a similar result for a family of events to hold simultaneously.
Lemma 3.5.3. Let $\mathcal{F}$ be the set of (possibly dependent) events $O_1, O_2, ..., O_{|\mathcal{F}|-1}, O_{|\mathcal{F}|}$ where each $O_k \in \mathcal{F}$ depends on the processing times $p = (p_1, p_2, ..., p_n)$. Let $0 < \varepsilon', \delta' < 1$. If $N > \frac{1}{2} (1/\varepsilon')^2 \ln(2/\delta')$ then

$$
\text{Prob}\{ |\text{Prob}_p\{O_k(p)\} - \text{Prob}_p\{O_k(p)\}| \leq \varepsilon' \forall k = 1, 2, ..., |\mathcal{F}| \} > 1 - |\mathcal{F}|\delta'.
$$

We characterized the subdifferential of $F$, $\partial F_p(.)$, in Section 3.4 with a closed-form expression Eq(3.15). We also derived a formula, Eq(3.16), to represent a component of any subgradient, $g_k(X, .)_p$. The formulas for $\partial F_p(.)$ and $g_k(X, .)_p$ are identical to the Eq(3.15) and Eq(3.16) respectively except that each $\text{Prob}_p\{\cdot\}$ term is replaced by the corresponding $\text{Prob}_p\{\cdot\}$ term.

We show in Lemma 3.5.4 that if we take a sufficiently large number of samples then $|g_k(X, \hat{A})_p - g_k(X, \hat{A})_{\hat{p}}|$ will be small with high probability for some $X \in \Theta$. This implies that there exists a small $g \in \partial F_p(\hat{A})$.

Recall that $\hat{A}$ is an optimal appointment vector for $F_p$. Therefore there exists $\hat{X} \in \Theta$ such that $g_k(\hat{X}, \hat{A})_{\hat{p}} = 0$ for all $1 \leq k \leq n + 1$. If we show that $|g_k(\hat{X}, \hat{A})_p - g_k(\hat{X}, \hat{A})_{\hat{p}}| < \varepsilon' K'$ then $|g_k(\hat{X}, \hat{A})_p - 0| < \varepsilon' K'$, and hence there exists $g \in \partial F_p(\hat{A})$ such that $|g_k| < \varepsilon' K'$ for all $1 \leq k \leq n + 1$. We now show that $|g_k(\hat{X}, \hat{A})_p - g_k(\hat{X}, \hat{A})_{\hat{p}}| < \varepsilon' K'$ with probability at least $1 - |\mathcal{F}|\delta'$ where $|\mathcal{F}| = 5n^2 + 5$ and $K' = n(9o_{\max} + 4u_{\max})$ where $o_{\max} = \max(o_1, o_2, ..., o_n)$ and $u_{\max} = \max(u_1, u_2, ..., u_n)$.

Lemma 3.5.4. If $N > \frac{1}{2} (1/\varepsilon')^2 \ln(2/\delta')$ then $|g_k(X, \hat{A})_p| < \varepsilon' K'$ with probability at least $1 - |\mathcal{F}|\delta'$ where $X \in \Theta$, $|\mathcal{F}| = 5n^2 + 5$ and $K' = n(9o_{\max} + 4u_{\max})$.

Remark 3.5.5. If $u_i = u$ for all $i = 1, 2, ..., n$ then $K' = n(4o_{\max} + 2u) + 2u$, i.e.,

$$
|g_k(\hat{X}, \hat{A})_p - g_k(\hat{X}, \hat{A})_{\hat{p}}| \leq \varepsilon'(n(4o_{\max} + 2u) + 2u) \quad (1 \leq k \leq n + 1)
$$

with probability at least $1 - |\mathcal{F}|\delta'$ where $|\mathcal{F}| = (n + 1)(4n + 2) = 5n^2 + 5$.

The last piece we need is a connection between $F_p(\hat{A})$ and $F_p(A^*)$ if there exists $g \in \partial F_p(\hat{A})$ such that $|g_k| < \varepsilon$. Before this result we need a Lemma to obtain a lower bound function for $F_p$.

Lemma 3.5.6. Let $\bar{p}_i = E[p_i]$, $\bar{p} = (\bar{p}_1, \bar{p}_2, ..., \bar{p}_n)$, $\bar{C}_1 = \bar{p}_1$ and $\bar{C}_i = \max(\bar{C}_{i-1}, A_i) + \bar{p}_i$. We define $\bar{f}(A) = (\sum_{i=1}^{n} [(\bar{C}_i - A_{i+1})^+ + (A_{i+1} - \bar{C}_i)^+]$ and $\bar{A} \in \arg\min_A \bar{f}(A)$. If cost coefficients $(u, o)$ are $\alpha$-monotone then $\bar{A} = (0, \bar{p}_1, \bar{p}_1 + \bar{p}_2, ..., \sum_{j=1}^{n} \bar{p}_j)$ and $F_p(A) \geq \bar{f}(A) \geq \frac{\varepsilon}{n} ||A - \bar{A}||_1$. 74
Remark 3.5.7. The following example with \( n = 2 \) jobs shows that this lower bound is tight, that is we may have \( \mathcal{F}_p(A) = \frac{\nu}{n} \|A - \tilde{A}\|_1 \). Let processing times \( p = (1, 4) \) be deterministic, \( u_1 = u_2 = o_1 = o_2 = 1 \) (therefore \( \nu = 1 \)). Then \( \tilde{A} = (0, 1, 5) \). For \( A = (0, 4, 8) \) \( \mathcal{F}(A) = 3 = \frac{1}{2} \sum_{i=1}^{n} |A_i - \tilde{A}_i| \).

The last step we need before our main result is to prove that for a suitably chosen \( \hat{A} \) we can obtain \( \mathcal{F}_p(\hat{A}) \leq (1 + \epsilon)\mathcal{F}_p(A^\ast) \) for any \( 0 < \epsilon \leq 1 \). This result follows easily by using a version of Lemma 5.1 of [22] (Lemma 3.7.2 in Section 3.7).

Lemma 3.5.8. Let \( 0 < \epsilon \leq 1 \). If there exists \( g \in \partial \mathcal{F}_p(\hat{A}) \) such that \( |g_k| < \nu/(3(n+1)n) \) for all \( 1 \leq k \leq n+1 \) then \( \mathcal{F}_p(\hat{A}) \leq (1 + \epsilon)\mathcal{F}_p(A^\ast) \).

Proof. If \( |g_k| < \nu/(3(n+1)n) \) for all \( 1 \leq k \leq n+1 \) then \( ||g||_1 \leq \nu/(3n) \). We then directly apply Lemma 3.7.2 with \( \mathcal{J}(A) = \frac{\nu}{n} ||A - \tilde{A}\|_1 \) (\( \mathcal{F}_p(A) \geq \frac{\nu}{n} ||A - \tilde{A}\|_1 \) by Lemma 3.5.6) and \( \alpha = \nu/(3n) \) obtain the desired result. \( \square \)

Combining Lemmata 3.5.3, 3.5.4 and 3.5.8 yields our main result for the sampling method.

Theorem 3.5.9. Let \( 0 < \epsilon \leq 1 \) (accuracy level) and \( 0 < \delta < 1 \) (confidence level) be given. If \( N > \left( 4.5(1/\epsilon)^2(n^2(n+1)(9o_{\text{max}}+4u_{\text{max}}))/\nu^2 \ln(2(5n^2+5)/\delta) \right) \) then \( \mathcal{F}_p(\hat{A}) \leq (1 + \epsilon)\mathcal{F}_p(A^\ast) \) with probability at least \( 1 - \delta \).

Remark 3.5.10. In the case of uniform underage cost coefficients (\( u_i = u \) for all \( i \)) the bound in Theorem 3.5.9 becomes \( \left( 4.5(1/\epsilon)^2(n^2(n+1)((4o_{\text{max}}+2u)+2u)/\nu^2 \ln(2(5n^2+5))/\delta) \right) \). Furthermore, the bound is similar but has a slightly higher polynomial (w.r.t. the number of jobs \( n \)) compared to the bound obtained for the multi-period newsvendor problem in [22] (w.r.t. the number of periods \( T \)). This is expected since in the appointment scheduling problem one needs to make all the decisions (i.e., determine the planned start times of all jobs) at once (before any processing starts), whereas in the inventory problem one decides sequentially at each period.

3.6 Conclusion

We consider the appointment scheduling problem with discrete random durations of Chapter 2 but without assuming any (prior) knowledge about the probability distribution of job
 durations. We show that the objective function is convex under a simple sufficient condition.

We work with subgradients of the objective function due to its non-differentiability. In fact we characterize the set of all subgradients, i.e., the subdifferential at a given appointment date vector with a closed-form formula. This is unusual since only a single subgradient may be obtained in most applications. We use the subdifferential characterization to relax the perfect information assumption of Chapter 2 on the probability distribution of processing times. We assume that there is an underlying (true) joint discrete distribution for the job durations, and only its independent samples are available, e.g., daily historical observations of surgery durations. Job durations may not necessarily be independent but samples are. In other words, we assume that job duration distribution is not known, i.e., no (prior) information about the distribution except that independent samples are available. We develop a sampling-based approach to determine the number of independent samples required to obtain a provably near-optimal solution with high probability, i.e., the cost of the sampling-based optimal schedule is with high probability no more than \((1 + \epsilon)\) times the cost of optimal schedule if the true distribution were known.

3.7 Proofs

**Proof. (Lemma 3.4.3)** If at most one of \(r_1, r_2, ..., r_m\) is non-zero then there is nothing to prove. Now suppose that there are at least two \(r_i > 0\), and w.l.o.g assume that \(r_1, r_2 > 0\). We first prove the result for \(m = 2\) then generalize it by induction. \(r_1X = \{r_1x : x \in X\}\) and \(r_1X + r_2X = \{r_1x + r_2y : x, y \in X\}\) by definition.

We first show that \(r_1X + r_2X \subseteq (r_1 + r_2)X\). Let \((a + b) \in (r_1X + r_2X)\) where \(a \in r_1X\) and \(b \in r_2X\). If \(a \in r_1X\) then \(\frac{a}{r_1} \in X\). Similarly if \(b \in r_2X\) then \(\frac{b}{r_2} \in X\). Since \(X\) is convex, \((\frac{a}{r_1} \lambda + \frac{b}{r_2} (1 - \lambda)) \in X\) for any \(0 \leq \lambda \leq 1\). Now let \(\lambda = \frac{r_1}{r_1 + r_2}\) then \((\frac{a}{r_1} \lambda + \frac{b}{r_2} (1 - \lambda)) = (\frac{a}{r_1} \frac{r_1}{r_1 + r_2} + \frac{b}{r_2} \frac{r_2}{r_1 + r_2}) = \frac{a + b}{r_1 + r_2}\) this implies \(a + b \in (r_1 + r_2)X\) and therefore \(r_1X + r_2X \subseteq (r_1 + r_2)X\).

Next, we show that \((r_1 + r_2)X \subseteq r_1X + r_2X\). Let \(a \in (r_1 + r_2)X\) then \(\frac{a}{r_1 + r_2} \in X\), and therefore \(r_1 \frac{a}{r_1 + r_2} \in r_1X\) and \(r_2 \frac{a}{r_1 + r_2} \in r_2X\). Hence \(r_1 \frac{a}{r_1 + r_2} + r_2 \frac{a}{r_1 + r_2} \in (r_1X + r_2X)\). But \(r_1 \frac{a}{r_1 + r_2} + r_2 \frac{a}{r_1 + r_2} = a\) therefore \((r_1 + r_2)X \subseteq r_1X + r_2X\). This completes the proof for \(m = 2\).

Next, assume that the result holds for \(m = k > 2\), i.e., \(\sum_{i=1}^{k}(r_iX) = (\sum_{i=1}^{k} r_i)X\).
We need to show that it also holds for \( m = k + 1 \), i.e., \( \sum_{i=1}^{k+1} (r_i X) = (\sum_{i=1}^{k+1} r_i) X \). Let \( r = \sum_{i=1}^{k} r_i \). Then
\[
\sum_{i=1}^{k+1} (r_i X) = \sum_{i=1}^{k} (r_i X) + r_{k+1} X = (\sum_{i=1}^{k} r_i) X + r_{k+1} X = r X + r_{k+1} X = (\sum_{i=1}^{k+1} r_i) X.
\]
where the second equality follows by the inductive assumption and fourth equality is due to our result for \( m = 2 \). Therefore the proof is complete.

**Proof.** (Lemma 3.4.6)

By Eq. (3.7), the fact that \( r(X + Y) = rX + rY \) (for \( r \in \mathbb{R} \) and, sets \( X \) and \( Y \) and Lemma 3.4.2 we obtain
\[
\partial T_j(A) = \sum_{p} \text{Prob}(\{ 1_k - 1_{j+1} : k \in I_j^+ \}) + \text{Prob}(\{ 0 \} \cup \{(1_k - 1_{j+1}) : k \in I_j^+ \})
\]
\[
= \sum_{p} \text{Prob}(\{ 1_k - 1_{j+1} : k \in I_j^+ \}) + \sum_{p} \text{Prob}(\{ 0 \} \cup \{1_k - 1_{j+1} : k \in I_j^+ \})
\]
Then by using the definition of \( \text{Prob}(I_j^+ = S) \) and Lemma 3.4.3 (similarly to Lemma 3.4.5) we get
\[
\sum_{p} \text{Prob}(\{ 1_k - 1_{j+1} : k \in I_j^+ \}) = \sum_{p} \text{Prob}(\{ 1_k - 1_{j+1} : k \in S \}) \text{Prob}(I_j^+ = S)
\]
\[
= \sum_{p} \sum_{s \in P^*(\{j\})} \text{Prob}(\{ 1_k - 1_{j+1} : k \in S \}) \text{Prob}(I_j^+ = S)
\]
Next by using the identity of
\[
\sum_{x} \text{Prob}(x) = \sum_{x} \text{Prob}(x : x \in X) + \sum_{x} \text{Prob}(x : x \notin X),
\]
the definition of \( \text{Prob}(I_j^+ = S) \) and Lemma 3.4.3 we rewrite \( \sum_{p} \text{Prob}(\{ 0 \} \cup \{1_k - 1_{j+1} : k \in I_j^+ \}) \) as
\[
= \sum_{p} \text{Prob}(p : I_j^+ = \{0\}) + \sum_{p} \text{Prob}(p : I_j^+ \neq \{0\}) \text{Prob}(\{ 0 \} \cup \{1_k - 1_{j+1} : k \in I_j^+ \})
\]
\[
= \sum_{p} \text{Prob}(p : I_j^+ \neq \{0\}) \sum_{s \in P^*(\{j\})} \text{Prob}(\{ 0 \} \cup \{1_k - 1_{j+1} : k \in S \}) \text{Prob}(I_j^+ = S)
\]
\[
= \sum_{s \in P^*(\{j\})} \sum_{p} \text{Prob}(p \{ 0 \} \cup \{1_k - 1_{j+1} : k \in S \}) \text{Prob}(I_j^+ = S)
\]
Therefore we finally obtain
\[
\partial T_j(A) = \sum_{S \in \mathcal{P}^*(\{j\})} [\text{Prob}(I_j^c = S) \text{co}\{1_k - 1_{j+1} : k \in S\} + \text{Prob}(I_j = S) \text{co}(\{0\} \cup \{1_k - 1_{j+1} : k \in S\})].
\]

Proof. (Lemma 3.4.7) Similarly to Lemma 3.4.6, by Eq(3.8) and Lemma 3.4.2 we obtain
\[
\partial M_j(A) = \sum_p \text{Prob}(p) \left[ \text{co}\{1_k - 1_{j+1} : k \in I_j^c\} + \text{co}(\{1_{j+1}\} \cup \{1_k : k \in I_j^c\}) \right]
= \sum_p \text{Prob}(p) \text{co}\{1_k - 1_{j+1} : k \in I_j^c\} + \sum_p \text{Prob}(p) \text{co}(\{1_{j+1}\} \cup \{1_k : k \in I_j^c\}).
\]
As in Lemma 3.4.6,
\[
\sum_p \text{Prob}(p) \text{co}\{1_k - 1_{j+1} : k \in I_j^c\} = \sum_{S \in \mathcal{P}^*(\{j\})} \text{Prob}(I_j^c = S) \text{co}\{1_k - 1_{j+1} : k \in S\}
\]
and we can rewrite \(\sum_p \text{Prob}(p) \text{co}(\{1_{j+1}\} \cup \{1_k : k \in I_j^c\})\) as
\[
\sum_p \text{Prob}(p : I_j^c \neq \emptyset) \text{co}\{1_k : k \in I_j^c \cup \{j+1\}\} + \sum_p \text{Prob}(p : I_j^c = \emptyset) 1_{j+1}
= \sum_{S \in \mathcal{P}^*(\{j\})} \sum_p \text{Prob}(p : I_j^c \neq \emptyset) 1\{I_j^c = S\} \text{co}\{1_k : k \in I_j^c \cup \{j+1\}\}
+ \sum_{S \in \mathcal{P}^*(\{j\})} \sum_p \text{Prob}(p : I_j^c = \emptyset) 1\{I_j^c = S\} 1_{j+1}
= \sum_{S \in \mathcal{P}^*(\{j\})} \left[ \text{Prob}(I_j^c = S) \text{co}\{1_k : k \in S \cup \{j+1\}\} \right] + (1 - \sum_{S \in \mathcal{P}^*(\{j\})} \text{Prob}(I_j^c = S)) 1_{j+1}.
\]
Hence we obtain
\[
\partial M_j(A) = \sum_{S \in \mathcal{P}^*(\{j\})} \left[ \text{Prob}(I_j^c = S) \text{co}\{1_k - 1_{j+1} : k \in S\} + \text{Prob}(I_j^c = S) \text{co}\{1_k : k \in S \cup \{j+1\}\} \right] + (1 - \sum_{S \in \mathcal{P}^*(\{j\})} \text{Prob}(I_j^c = S)) 1_{j+1}.
\]

Proof. (Proposition 3.4.11) Let \(y \in \partial F(\tilde{A}, D)\). Then by subgradient inequality we have
\[
F(B, B_{n+1}) \geq F(\tilde{A}, D) + (B - \tilde{A}, B_{n+1} - D)y^t
\]
for all \(B = (B_1, ..., B_n) \in \mathbb{R}^n\) and \(B_{n+1} \in \mathbb{R}\). This inequality holds for all \((B, B_{n+1}) \in \mathbb{R}^{n+1}\) and in particular \((B, D)\). Thus we obtain
\[
F(B, D) \geq F(\tilde{A}, D) + (B - \tilde{A}, D - D)y^t.
\]
This gives \(F^D(B) \geq F^D(\tilde{A}) + (B - \tilde{A})g^t\) since \(F(B, D) = F^D(B)\) and \(F(\tilde{A}, D) = F^D(\tilde{A})\) where \(g = \text{proj}(y) = (y_1, ..., y_n)\).
\(F^D(B) \geq F^D(\tilde{A}) + (B - \tilde{A})g^t\) implies that \(g = \text{proj}(y) \in \partial F^D(\tilde{A})\).
Proof. (Lemma 3.5.3) The proof is by induction on $|F|$. Let $1 \leq k \leq |F|$ and

$$Y_k = \{ \left| \text{Prob}_p\{O_k(p)\} - \text{Prob}_{\hat{p}}\{O_k(p)\} \right| \leq \varepsilon' \}.$$

For $F = 2$ the result holds since

$$\text{Prob}\{Y_1 \cap Y_2\} = 1 - \text{Prob}\{Y_1 \cap Y_2\} = 1 - \text{Prob}\{Y_1 \cup Y_2\} \geq 1 - (\text{Prob}\{Y_1\} + \text{Prob}\{Y_2\}) \quad \text{(since Prob}\{Y_1 \cup Y_2\} \leq \text{Prob}\{Y_1\} + \text{Prob}\{Y_2\}) \geq 1 - 2\delta' \quad \text{(since Prob}\{Y_1\}, \text{Prob}\{Y_2\} < \delta').$$

Suppose the result is true for $|F| = k$, i.e., $\text{Prob}\{\cap_{i=1}^k Y_i\} \geq 1 - k\delta'$. Let $Y = \bigcap_{i=1}^k Y_i$. Then,

$$\text{Prob}\{Y \cap Y_{k+1}\} = 1 - \text{Prob}\{Y \cap Y_{k+1}\} = 1 - \text{Prob}\{Y \cup Y_{k+1}\} \geq 1 - (\text{Prob}\{Y\} + \text{Prob}\{Y_{k+1}\}) \quad \text{(as Prob}\{Y \cup Y_{k+1}\} \leq \text{Prob}\{Y\} + \text{Prob}\{Y_{k+1}\}) \geq 1 - (k + 1)\delta' \quad \text{(as Prob}\{Y\} < k\delta' \text{ and Prob}\{Y_{k+1}\} < \delta').$$

Therefore the result is also true for $|F| = k + 1$, and hence the proof is complete. \hfill \Box

Proof. (Lemma 3.5.4). Since $\hat{A}$ is an optimal appointment vector for $F_{\hat{p}}$ there exists $\hat{X} \in \Theta$ such that $g_k(\hat{X}, \hat{A})_{\hat{p}} = 0$ for all $1 \leq k \leq n + 1$. If $|g_k(\hat{X}, \hat{A})_{\hat{p}} - g_k(\hat{X}, \hat{A})_{\hat{p}}| < \varepsilon'K'$ then this implies that $|g_k(\hat{X}, \hat{A})_{\hat{p}} - g_k(\hat{X}, \hat{A})_{\hat{p}}| < \varepsilon'K'$, and hence there exists $g \in \partial F_{\hat{p}}(\hat{A})$ such that $|g_k| < \varepsilon'K'$ for all $1 \leq k \leq n + 1$. We now show that $|g_k(\hat{X}, \hat{A})_{\hat{p}} - g_k(\hat{X}, \hat{A})_{\hat{p}}| < \varepsilon'K'$. We start by taking the difference $|g_k(\hat{X}, \hat{A})_{\hat{p}} - g_k(\hat{X}, \hat{A})_{\hat{p}}|$ term by term and factor out $\hat{X}$ terms by using Eq(3.16).
\[ |g_k(\tilde{X}, \tilde{A})_p - g_k(\tilde{X}, \tilde{A})_{\tilde{p}}| \]
\[
= \left| \sum_{j=k}^{n} \alpha_j \sum_{S \in P^*(\{j\})} \tilde{X}_{kj}^L(S) \left( \text{Prob}_p \{I_j = S\} - \text{Prob}_{\tilde{p}} \{I_j = S\} \right) 
- \alpha_{k-1} \sum_{S \in P^*(\{k-1\})} \left( \text{Prob}_p \{I_{k-1} = S\} - \text{Prob}_{\tilde{p}} \{I_{k-1} = S\} \right) 
+ \sum_{j=k}^{n} \beta_j \sum_{S \in P^*(\{j\})} \tilde{X}_{kj}^T(S) \left( \text{Prob}_p \{I_j^T = S\} - \text{Prob}_{\tilde{p}} \{I_j^T = S\} \right) 
- \beta_{k-1} \sum_{S \in P^*(\{k-1\})} \tilde{X}_{k-1j}^T(S) \left( \text{Prob}_p \{I_{k-1}^T = S\} - \text{Prob}_{\tilde{p}} \{I_{k-1}^T = S\} \right) 
+ \sum_{j=k}^{n} \gamma_j \sum_{S \in P^*(\{j\})} \tilde{X}_{kj}^M(S) \left( \text{Prob}_p \{I_j^M = S\} - \text{Prob}_{\tilde{p}} \{I_j^M = S\} \right) 
- \gamma_{k-1} \sum_{S \in P^*(\{k-1\})} \tilde{X}_{k-1j}^M(S \cup \{n+1\}) \left( \text{Prob}_p \{I_{k-1}^M = S\} - \text{Prob}_{\tilde{p}} \{I_{k-1}^M = S\} \right) \right|. 
\]

The term \(-\alpha_{k-1} \sum_{S \in P^*(\{k-1\})} \left( \text{Prob}_p \{I_{k-1} = S\} - \text{Prob}_{\tilde{p}} \{I_{k-1} = S\} \right)\) disappears since \(\sum_{S \in P^*(\{k-1\})} \text{Prob}_p \{I_{k-1} = S\} = 1 = \sum_{S \in P^*(\{k-1\})} \text{Prob}_{\tilde{p}} \{I_{k-1} = S\}\). By using triangular inequality we obtain
\[ |g_k(\hat{X}, \hat{A})_p - g_k(\tilde{X}, \tilde{A})_{\tilde{p}}| \]
\[ \leq \left| \sum_{j=k}^{n} \alpha_j \sum_{S \in \mathcal{P}^*([j])} \hat{X}^L_{kj}(S) \left( \text{Prob}_p\{I_j = S\} - \text{Prob}_{\tilde{p}}\{I_j = S\} \right) \right| \]
\[ + \left| \sum_{j=k}^{n} \beta_j \sum_{S \in \mathcal{P}^*([j])} \hat{X}^T_{kj}(S) \left( \text{Prob}_p\{I_j^+ = S\} - \text{Prob}_{\tilde{p}}\{I_j^+ = S\} \right) \right| \]
\[ + \left| \beta_{k-1} \sum_{S \in \mathcal{P}^*([k-1])} \left( \text{Prob}_p\{I_{k-1}^+ = S\} - \text{Prob}_{\tilde{p}}\{I_{k-1}^+ = S\} \right) \right| \]
\[ + \left| \sum_{j=k}^{n} \beta_j \sum_{S \in \mathcal{P}^*([j])} \hat{X}^T_{kj}(S) \left( \text{Prob}_p\{I_j^- = S\} - \text{Prob}_{\tilde{p}}\{I_j^- = S\} \right) \right| \]
\[ + \left| \beta_{k-1} \sum_{S \in \mathcal{P}^*([k-1])} \sum_{s \in S} \hat{X}^T_{ik-1}(S) \left( \text{Prob}_p\{I_{k-1}^- = S\} - \text{Prob}_{\tilde{p}}\{I_{k-1}^- = S\} \right) \right| \]
\[ + \left| \sum_{j=k}^{n} \gamma_j \sum_{S \in \mathcal{P}^*([j])} \hat{X}^T_{kj}(S) \left( \text{Prob}_p\{I_j^- = S\} - \text{Prob}_{\tilde{p}}\{I_j^- = S\} \right) \right| \]
\[ + \left| \gamma_{k-1} \sum_{S \in \mathcal{P}^*([k-1])} \left( \text{Prob}_p\{I_{k-1}^- = S\} - \text{Prob}_{\tilde{p}}\{I_{k-1}^- = S\} \right) \right| \]
\[ + \left| \sum_{j=k}^{n} \gamma_j \sum_{S \in \mathcal{P}^*([j])} \hat{X}^M_{kn}(S \cup \{n+1\}) \left( \text{Prob}_p\{I_n^- = S\} - \text{Prob}_{\tilde{p}}\{I_n^- = S\} \right) \right| \]
\[ + \left| \gamma_{k-1} \sum_{S \in \mathcal{P}^*([k-1])} \left( \text{Prob}_p\{I_{k-1}^- = S\} - \text{Prob}_{\tilde{p}}\{I_{k-1}^- = S\} \right) \right| \]. \quad (3.17)

We now find an upper bound for \( |g_k(\hat{X}, \hat{A})_p - g_k(\tilde{X}, \tilde{A})_{\tilde{p}}| \) by obtaining an upper bound for each \(|.|\) term in Eq(3.17). We do so by using the fact that \( \tilde{X} \in \Theta \) and rewriting some of the probability terms. Note that we will show this for the first and the third terms as the remaining bounds are obtained similar to either of the first or the third. We start with the first term in Eq(3.17).

\[ \left| \sum_{j=k}^{n} \alpha_j \sum_{S \in \mathcal{P}^*([j])} \hat{X}^L_{kj}(S) \left( \text{Prob}_p\{I_j = S\} - \text{Prob}_{\tilde{p}}\{I_j = S\} \right) \right| \]
\[ = \left| \sum_{j=k}^{n} \alpha_j \sum_{S \in \mathcal{P}^*([j])} \hat{X}^L_{kj}(S) \left( 1\{k \in S\} \text{Prob}_p\{I_j = S\} - 1\{k \in S\} \text{Prob}_{\tilde{p}}\{I_j = S\} \right) \right| , \]
since \( \hat{X}^L_{kj}(S) = 0 \) if \( k \notin S \). Let \( \mathcal{P}^+([j]) = \{ S \in \mathcal{P}^*([j]) \mid \text{Prob}_p\{I_j = S\} - \text{Prob}_{\tilde{p}}\{I_j = S\} \geq 0 \} \). Then by definition of \( \mathcal{P}^+([j]) \), the fact that \( 0 \leq \hat{X}^L_{kj}(S) \leq 1 \) and triangular inequality
we obtain

\[
\left| \sum_{j=k}^{n} \alpha_j \sum_{S \in \mathcal{P}^+([j])} \hat{X}_{kj}^L(S) \left(1 \{ k \in S \} \text{Prob}_p \{ I_j = S \} - 1 \{ k \in S \} \text{Prob}_p \{ I_j = S \} \right) \right| \\
\leq \left| \sum_{j=k}^{n} \alpha_j \sum_{S \in \mathcal{P}^+([j])} \hat{X}_{kj}^L(S) \left(1 \{ k \in S \} \text{Prob}_p \{ I_j = S \} - 1 \{ k \in S \} \text{Prob}_p \{ I_j = S \} \right) \right| \\
\leq \left| \sum_{j=k}^{n} \alpha_j \sum_{S \in \mathcal{P}^+([j])} \left(1 \{ k \in S \} \text{Prob}_p \{ I_j = S \} - 1 \{ k \in S \} \text{Prob}_p \{ I_j = S \} \right) \right| \\
= \left| \sum_{j=k}^{n} \alpha_j \left( \text{Prob}_p \{ k \in I_j, I_j \in \mathcal{P}^+([j]) \} - \text{Prob}_p \{ k \in I_j, I_j \in \mathcal{P}^+([j]) \} \right) \right| \\
\leq \sum_{j=k}^{n} \alpha_j \max_{k} \left( \text{Prob}_p \{ k \in I_j, I_j \in \mathcal{P}^+([j]) \} - \text{Prob}_p \{ k \in I_j, I_j \in \mathcal{P}^+([j]) \} \right) \\
\leq \sum_{j=k}^{n} \alpha_j \varepsilon' \leq \varepsilon' \alpha_{\max} n.
\]

Similarly,

\[
\left| \sum_{j=k}^{n} \beta_j \sum_{S \in \mathcal{P}^+([j])} \hat{X}_{kj}^T(S) \left( \text{Prob}_p \{ I_j^> = S \} - \text{Prob}_p \{ I_j^> = S \} \right) \right| \leq \varepsilon' \beta_{\max} n, \\
\left| \sum_{j=k}^{n} \beta_j \sum_{S \in \mathcal{P}^+([j])} \hat{X}_{kj}^T(S) \left( \text{Prob}_p \{ I_j^< = S \} - \text{Prob}_p \{ I_j^< = S \} \right) \right| \leq \varepsilon' \beta_{\max} n, \\
\left| \sum_{j=k}^{n} \gamma_j \sum_{S \in \mathcal{P}^+([j])} \hat{X}_{kj}^T(S) \left( \text{Prob}_p \{ I_j^\gamma = S \} - \text{Prob}_p \{ I_j^\gamma = S \} \right) \right| \leq \varepsilon' \gamma_{\max} n, \\
\left| \sum_{j=k}^{n} \gamma_j \sum_{S \in \mathcal{P}^+([n])} \hat{X}_{kj}^M(S \cup \{ n + 1 \}) \left( \text{Prob}_p \{ I_n^= = S \} - \text{Prob}_p \{ I_n^= = S \} \right) \right| \leq \varepsilon' \gamma_{\max} n.
\]
We now find an upper bound for the third term in Eq (3.17).

\[
\left| \beta_{k-1} \sum_{S \in \mathcal{P}^*([k-1])} \left( \text{Prob}_p(I^*_1 \cap I_{k-1}^*; S) - \text{Prob}_p(I_{k-1}^* = S) \right) \right|
\]

\[
= \left| \beta_{k-1} \sum_{S \in \mathcal{P}^*([k-1])} \left( \text{Prob}_p(I_{k-1}^* = S \text{ and } P_{i,k-1} > A_k - A_i : i \in S) \right.
\]

\[
- \left. \text{Prob}_p(I_{k-1}^* = S \text{ and } P_{i,k-1} > A_k - A_i : i \in S) \right) \right|
\]

\[
= \left| \beta_{k-1} \sum_{S \in \mathcal{P}^*([k-1])} \left( \text{Prob}_p(I_{k-1}^* = S \text{ and } P_{i,k-1} > A_k - A_i) \right) \right|
\]

\[
- \text{Prob}_p(I_{k-1}^* = S \text{ and } P_{i,k-1} > A_k - A_i \leq \mathbb{1}\{i \in S\}) \right|
\]

\[
= \left| \beta_{k-1} \sum_{i=1}^{k-1} \left( \text{Prob}_p(i \in I_{k-1}^* \text{ and } P_{i,k-1} > A_k - A_i) - \text{Prob}_p(i \in I_{k-1}^* \text{ and } P_{i,k-1} > A_k - A_i) \right) \right|
\]

\[
= \left| \beta_{k-1} \sum_{i=1}^{k-1} \left( \text{Prob}_p(i \in I_{k-1}^*) - \text{Prob}_p(i \in I_{k-1}^*) \right) \right|
\]

\[
\leq \beta_{k-1} \sum_{i=1}^{k-1} \epsilon' \leq \epsilon' \beta_{k-1} n.
\]

Similarly, we get

\[
\left| \gamma_{k-2} \sum_{S \in \mathcal{P}^*([k-2])} \sum_{i \in S} X_{i,k-1}^S \left( \text{Prob}_p(I_{k-1}^* = S) - \text{Prob}_p(I_{k-1}^* = S) \right) \right| \leq \epsilon' \beta_{k-1} n.
\]

\[
\left| \gamma_{k-2} \sum_{S \in \mathcal{P}^*([k-2])} \left( \text{Prob}_p(I_{k-1}^* = S) - \text{Prob}_p(I_{k-1}^* = S) \right) \right| \leq \epsilon' \gamma_{k-1} n.
\]

\[
\gamma_{k-2} \sum_{S \in \mathcal{P}^*([k-2])} \left( \text{Prob}_p(I_{k-1}^* = S) - \text{Prob}_p(I_{k-1}^* = S) \right) \right| \leq \epsilon' \gamma_{k-1} n.
\]

Therefore we can bound \(|g_k(\hat{X}, \hat{A}_p - g_k(\tilde{X}, \tilde{A}_p)| \) from above:

\[
|g_k(\hat{X}, \hat{A}_p - g_k(\tilde{X}, \tilde{A}_p)| \leq \epsilon' n(\alpha_{\max} + 4\beta_{\max} + 4\gamma_{\max}) \quad (1 \leq k \leq n + 1).
\]

Since the cost coefficients \((u, o)\) are \(\alpha\)-monotone we have \(0 \leq \alpha_i \leq \alpha_{\max}, \beta_i \leq \alpha_{\max}\) and 
\(\gamma_i \leq u_{\max} + \alpha_{\max} - \gamma_{\max}. \) Therefore \((\alpha_{\max} + 4\beta_{\max} + 4\gamma_{\max}) \leq (9\omega_{\max} + 4u_{\max})\) so we can take 
\(K' = n(9\omega_{\max} + 4u_{\max})\). We also determine \(|\mathcal{F}|\), the maximum number of events we need to compute \(|g_k(\hat{X}, \hat{A}_p - g_k(\tilde{X}, \tilde{A}_p)| \) for all \(k\). For each \(k\), we have \((5(n - k + 1) + 4(k - 1))\) therefore at most \(5n\) events. Since \(k \leq n + 1\) we have \(|\mathcal{F}| = (n + 1)(5n) = 5n^2 + 5\). This completes the proof. \(\square\)
Proof. (Lemma 3.5.6) Fix \( A \). Let \( h(\mathbf{p}) = F(A|\mathbf{p}) = \sum_{i=1}^{n} [\alpha_i(C_i - A_{i+1})^+ + u_i(A_{i+1} - C_i)^+] \). We claim that \( h \) is convex. Recall that by Identity Lemma 3.3.1, we can rewrite \( F(A|\mathbf{p}) \) and hence \( h(\mathbf{p}) \) as

\[
h(\mathbf{p}) = F(A|\mathbf{p}) = \sum_{i=1}^{n} \left[ \alpha_i(C_i - A_{i+1}) + \beta_i(C_i - A_{i+1})^+ + \gamma_i(\max\{C_i, A_{i+1}\} - \sum_{k=1}^{i} p_k) \right]
\]

for any \( \alpha_i \in \mathbb{R} \) (\( 1 \leq i \leq n \)) where \( \beta_i = (\alpha_i - \alpha_i) \) and \( \gamma_i = [(u_i + \alpha_i) - (u_{i+1} + \alpha_{i+1})] \). Recall that \( C_i = \max_{k \leq i} \{A_k + \sum_{t=k}^{i} p_t\} \) (by the Critical Path Lemma 2.4.1) so \( C_i \) is convex in \( \mathbf{p} \). By \( \alpha \)-monotonicity \( \alpha_i, \beta_i \geq 0 \) hence the terms \( \alpha_i(C_i - A_{i+1}) \) and \( \beta_i(C_i - A_{i+1})^+ \) are convex in \( \mathbf{p} \). Furthermore, the term \( \gamma_i(\max\{C_i, A_{i+1}\} - \sum_{k=1}^{i} p_k) \) is convex (in fact linear) in \( \mathbf{p} \). Therefore \( h(\mathbf{p}) \) is convex.

Recall that \( \nu = \min\{u_1, u_2, ..., u_n, o_1, o_2, ..., o_n\} \) and \( \tilde{C}_i \)'s are the completion times, but they are deterministic since we are using expected values, \( \tilde{p}_i \)'s, for the processing times. We next show that \( F_p(A) \geq \tilde{f}(A) \) by applying Jensen’s inequality to \( h(\mathbf{p}) \) and applying Identity Lemma 3.3.1 to \( F(A|\tilde{p}) \).

\[
F_p(A) = E_p[h(\mathbf{p})] \geq F(A|E\mathbf{p}) = F(A|\tilde{p}) = \sum_{i=1}^{n} \left[ \alpha_i(\tilde{C}_i - A_{i+1}) + \beta_i(\tilde{C}_i - A_{i+1})^+ + \gamma_i(\max\{\tilde{C}_i, A_{i+1}\} - \sum_{k=1}^{i} \tilde{p}_k) \right] = \sum_{i=1}^{n} [\nu(\tilde{C}_i - A_{i+1})^+ + (A_{i+1} - \tilde{C}_i)^+] = \tilde{f}(A).
\]

Next we obtain \( \tilde{A} \in \arg \min_{A} \tilde{f}(A) \). Note that \( \tilde{f}(A) \geq 0 \) for all \( A \). Set \( \tilde{A}_1 = 0, ..., \tilde{A}_{i+1} = \sum_{j=1}^{i} \tilde{p}_j, ..., \tilde{A}_{n+1} = \sum_{j=1}^{n} \tilde{p}_j, \) i.e., \( \tilde{A}_{i+1} - \tilde{A}_i = \tilde{p}_i \) and \( \tilde{A}_{i+1} = \sum_{k=1}^{i} \tilde{p}_k \) for \( 1 \leq i \leq n \). Then \( \tilde{A}_{i+1} = \tilde{C}_i \) for all \( i = 1, ..., n \) and \( \tilde{f}(\tilde{A}) = 0 \). Therefore \( \tilde{A} = (0, \tilde{p}_1, ..., \sum_{j=1}^{n} \tilde{p}_j) \) is indeed optimal for \( \tilde{f} \).

We next show \( F_p(A) \geq \frac{\nu}{n}||A - \tilde{A}||_1 \) by showing \( \tilde{f}(A) \geq \frac{\nu}{n}||A - \tilde{A}||_1 \). Note that \( \sum_{i=1}^{n} [(\tilde{C}_i - A_{i+1})^+ + (A_{i+1} - \tilde{C}_i)^+] = \sum_{i=1}^{n} |(\tilde{C}_i - A_{i+1})| \), and the result would follow if we show \( \sum_{i=1}^{j} |(\tilde{C}_i - A_{i+1})| \geq |A_{j+1} - \tilde{A}_{j+1}| = |A_{j+1} - \sum_{i=1}^{j} \tilde{p}_i| \) for all \( j = 1, 2, ..., n \). We now show \( \sum_{i=1}^{j} |(\tilde{C}_i - A_{i+1})| \geq |A_{j+1} - \sum_{i=1}^{j} \tilde{p}_i| \) for all \( j = 1, 2, ..., n \). We distinguish two cases.
First, suppose that \( A_{j+1} \leq \sum_{i=1}^{j} \tilde{p}_t \). Since \( \sum_{i=1}^{j} \tilde{p}_t \leq \tilde{C}_j \), we have \( A_{j+1} \leq \tilde{C}_j \). Therefore
\[
\sum_{i=1}^{j} |(\tilde{C}_i - A_{i+1})| \geq |(\tilde{C}_j - A_{j+1})| \geq |\sum_{t=1}^{j} \tilde{p}_t - A_{j+1}| = |A_{j+1} - A_{j+1}|.
\]
The second case is where \( A_{j+1} \leq \sum_{i=1}^{j} \tilde{p}_t \). Then
\[
\sum_{i=1}^{j} |(\tilde{C}_i - A_{i+1})| \geq \sum_{i=1}^{j} (A_{i+1} - \tilde{C}_i) = \max(\tilde{C}_j, A_{j+1}) - \sum_{i=1}^{j} \tilde{p}_t \geq A_{j+1} - \sum_{i=1}^{j} \tilde{p}_t = |A_{j+1} - \tilde{A}_{j+1}|,
\]
where the first equality follows from Identity Lemma 3.3.1. Hence we obtain
\[
\sum_{i=1}^{j} |(\tilde{C}_i - A_{i+1})| \geq |A_{j+1} - \tilde{A}_{j+1}| \text{ for all } 1 \leq j \leq n.
\]
Therefore for every \( j = 1, ..., n \)
\[
\sum_{i=1}^{n} |(\tilde{C}_i - A_{i+1})| \geq \sum_{i=1}^{j} |(\tilde{C}_i - A_{i+1})| \geq |A_{j+1} - \tilde{A}_{j+1}|
\]
and hence
\[
n \tilde{f}(A) = n\nu(\sum_{i=1}^{n} |(\tilde{C}_i - A_{i+1})| + (A_{i+1} - \tilde{C}_i)) = n\nu(\sum_{i=1}^{n} |\tilde{C}_i - A_{i+1}|) \geq \nu \|A - \tilde{A}\|_1
\]
as desired. Therefore \( F_p(A) \geq \frac{\nu}{n} \|A - \tilde{A}\|_1 \). This completes the proof. \(\square\)

**Definition 3.7.1.** (Definition 3.3 of [22]) Let \( f : \mathbb{R}^m \rightarrow \mathbb{R} \) be convex. A point \( y \) is an \( \alpha \)-point if there exists \( g \in \partial f(y) \) such that \( \|g\|_1 \leq \alpha \).

**Lemma 3.7.2.** (A version of Lemma 5.1 of [22]). Let \( f : \mathbb{R}^m \rightarrow \mathbb{R} \) be convex, finite with a global minimizer \( y^* \). Assume that there exists \( \tilde{f} \) such that \( f \geq \tilde{f} = \lambda \|y - \tilde{y}\|_1 \) for some \( \lambda > 0 \) and \( \tilde{y} \in \mathbb{R}^m \). If \( \tilde{y} \) is an \( \alpha \)-point for \( \alpha = \lambda \epsilon/3 \) then \( f(\tilde{y}) \leq (1 + \epsilon)f(y^*) \).

**Proof.** Let \( L = f(y^*)/\lambda \). Consider the norm \( l_1 \) ball \( B = B(\tilde{y}, L) \), then \( y^* \in B(\tilde{y}, L) = \{\lambda \|y^* - \tilde{y}\| \leq f(y^*)\} \). Subgradient inequality at \( \tilde{y} \) combined with Cauchy-Schwartz inequality yields
\[
f(\tilde{y}) - f(y^*) \leq \alpha \|\tilde{y} - y^*\|_1 \quad \text{(since Cauchy-Schwartz inequality also holds for \( l_1 \) norm).}
\]
We also have
\[
||\tilde{y} - y^*||_1 \leq ||\tilde{y} - \tilde{y}||_1 + ||\tilde{y} - y^*||_1 \leq f(\tilde{y})/\lambda + L = f(\tilde{y})/\lambda + f(y^*)/\lambda.
\]
So we obtain \( f(\tilde{y}) - f(y^*) \leq \alpha(f(\tilde{y})/\lambda + f(y^*)/\lambda) \) and hence \( f(\tilde{y}) \leq f(y^*)(\lambda + \alpha)/(\lambda - \alpha) \).
If we choose \( \alpha \leq \lambda \epsilon/3 \) the result follows. \(\square\)

85
3.8 Bibliography


88


4 Incentive-Based Surgery Scheduling: Determining Optimal Number of Surgeries

We study the problem of determining the number of surgeries for an operating room (OR) block where surgery durations are random, there are significant idle and overtime costs for running an OR and the incentives of the parties involved (hospital and surgeon) are not aligned. We explore the interaction between the hospital and the surgeon in a game theoretic setting, present empirical findings on surgery durations and suggest, under reasonable assumptions, payment schemes that the hospital may offer to the surgeon to reduce its (idle and especially overtime) costs.

4.1 Introduction

Healthcare is one of the biggest industries in North America. Canada was expected to spend $148 billion on healthcare in 2006 [8], which accounts for more than 10% of its GDP. In the United States the situation is similar, in 2006 it accounted for 15.3% of GDP [6]. Healthcare challenges, including rising costs and demand, are continually becoming more acute not only in Canada and the United States, but in almost every country in the world [5].

In [10], Glouberman and Mintzberg develop a novel framework to analyze healthcare management. According to this framework, we can think of healthcare as an industry like any other, but with some additional unique characteristics. Unlike other businesses, either private or public, no one is in complete charge (e.g., of a hospital), and there are several decision makers with conflicting objectives. For instance, managers make resource allocation decisions, but it is doctors who decide what to do with those resources.

---

1A version of this chapter will be submitted for publication. Begen M.A., Ryan C. and Queyranne M. Incentive-Based Surgery Scheduling: Determining Optimal Number of Surgeries.
Healthcare is more easily categorized as belonging to the public sector than the private sector in most countries. In his article [7], Dixit states the following about public sector agencies: “Public sector agencies have some special features, most notably a multiplicity of dimensions – of tasks, of the stakeholders and their often conflicting interests about the ends and the means, and of the tiers of management and front-line workers”. The framework in [10] makes it easier to understand this statement and why the healthcare system is difficult to manage. We can think of hospitals as essentially independent blocks of an healthcare system. According to [10], there are four different management groups – four worlds, if you will – within a hospital, as shown in Figure 4.1.

Figure 4.1: Healthcare Players and Tasks

Each group in Figure 4.1 has its own objectives, and has some scope to make their own decisions. Doctors and nurses deliver clinical operations, and hence focus on downstream considerations; that is, closer to dealing with the actual health of patients. Managers and trustees are responsible for budgeting and raising funds for the hospital, so their concerns upstream. On the other hand, employees (managers and nurses) work in the hospital, while doctors and trustees work out of the hospital, since they are not technically employees of the hospital. In Canada and the United States, although some doctors are salaried hospital employees, most doctors are private entrepreneurs who have admission privileges at a hospital, work on a fee-for-service basis and appear when the patient needs a cure or treatment [5].
The significance of this framework for our study is the fact that in order to provide a medical service, such as a surgery, all four groups have a unique contribution to make. Conversely, the actions of any one of the groups has an effect on the ability of the others to perform their duties. The picture gets even more complicated when we think of government and insurance companies. In [10], Glouberman and Mintzberg conclude that these decision-makers must achieve a certain level of integration to provide effective healthcare management. Other studies, including Calmes and Shusterich [3] and Marco [15], reach a similar conclusion for operating room management. Operating rooms are one of the most essential places of a hospital, and also one of the costliest. These authors, among others (see for instance, [11, 13]) point out that ORs are one of the most difficult places to manage in a hospital, and it is imperative to improve collaboration between the players (of an OR) for any advancement in OR management. In this chapter, we focus on this small but important part of healthcare operations – surgery scheduling and, more specifically setting the number of surgeries for an OR.

In particular, we study the interactions between a surgeon and a hospital in determining number of surgeries for an OR where surgery durations are random, and there are significant costs for idle time and overtime. More specifically, we explore the commonly observed situation reported in the literature Olivares et al. [18] and observed empirically (Section 4.2) that surgeons over-schedule their allotted OR time, i.e., they schedule too many surgeries for their OR time. We argue that this observation can be explained by the incentive of surgeons to take advantage of fee-for-service payment structure for surgeries performed combined with the fact surgeons do not bear overtime costs at the hospital level. This creates a cost which is borne by the hospital who operates the OR and pays surgery support staff. Thus in our model we discuss that the hospital has an incentive to limit the number of surgeries performed by surgeons to reduce overtime expenditures. We explore this misalignment of incentives – for the surgeon to over-schedule and the hospital to control overtime costs – in a game theoretic setting.

Only recently some systematic attention has been given in the literature to incentive issues in health care management. For instance, researchers have studied physician-patient, government-physician, and hospital-physician relationships in the context of principal-agent modeling framework [23]. There are also studies such as [19] that look at a bigger picture and explore patient-physician-third party payer relationships. A more recent study [12]
develops a framework to empirically estimate the parameters of a principal-agent model to
design a payment system for dialysis providers. Other authors focus on expansion of OR’s
[13] and stakeholder interactions in OR’s [14] in basic game theoretic environments. In
[14], it is stated that many interactions between surgeons, anesthetists, nurses and hospital
management can be seen as a repeated game. We also mention two empirical studies
[12, 18] which estimate cost parameters in a principal agent model and newsvendor model
respectively for making decisions about compensation and capacity. Of particular interest
to the present work is a suggestion raised in Olivares et al. [18] that the amount of schedule
overruns are mainly caused by incentive conflicts and over-confidence. Our chapter takes a
systematic look at this very question, providing a model by which these incentive conflicts
can be identified and effectively analyzed.

To the best of our knowledge our chapter presents the first systematic study of deter-
mining the number of surgeries for an OR block, investigating the interaction between the
surgeon and hospital (management) in a game- theoretic setting. Our research has been
motivated by our observations from applied healthcare projects such as [20], literature, e.g.,
[18], and empirical findings (Section 4.2).

The organization of the chapter is as follows. In Section 4.2, we define and motivate
the problem, give an overview of the surgery scheduling process, present data and discuss
findings (empirical, literature-based, anecdotal-based) on the underestimating of surgery du-
rations, and hence on overtime in an OR (block). Section 4.3 presents our model, including
notation and a thorough discussion of assumptions. In this section, we define the objective
functions of the surgeon and the hospital and explore their properties. In Section 4.4, we
demonstrate a misalignment of incentives between the hospital and surgeon. Section 4.5
provides alternative contracting scenarios whereby this misalignment of incentives can be
aligned, and take care to characterize sufficient conditions for when these schemes are cost
effective for the hospital. This section also includes a discussion of welfare considerations
from the perspective of an upstream planner (in the Canadian system, the provincial gov-
ernment) who is concerned for maximizing social welfare, including impacts on patients.
We consider a different formulation of our model in Section 4.6 which makes alternative
assumptions about the independence of surgeries, and in this framework we explore the
impact on some of our results in the presence of a risk-averse surgeon. All the proofs of our
analytical results are placed in Section 4.8.
4.2 Problem Description and Motivation

The objective of our study is to determine the number of elective (i.e., non-emergency, scheduled) surgeries for an OR block where surgery durations are random, there are significant idle and overtime costs, and the incentives of the hospital and surgeon (parties involved in the scheduling process) are not necessarily aligned.

We start with an overview of a surgery scheduling process. In practice, scheduling surgeries in a medical facility is a complex and important process, and the choice of schedule directly impacts the number of patients treated for each specialty, cancelation of surgeries, utilization of resources, wait times, and the overall performance of the system [20]. The surgery scheduling process for elective cases is usually considered as a three-level process [1, 2, 18], which we now describe.

The first level defines and assigns the OR time among the surgical specialties, usually called mix planning. A surgical OR block schedule is developed at the second level. An OR block schedule is simply a table that assigns each specialty surgery time in ORs on each day. The times are called blocks. The OR block schedule is sometimes called the master surgical schedule (see Figure 2 of [20] for a sample OR block schedule). Finally, in the third level we schedule individual cases on a daily basis, also known as patient mix. We can classify these levels as strategic, tactical and operational stage of the surgery scheduling process respectively. Figure 4.2 gives an overview of the process in terms of decisions, decision maker and decision level.

<table>
<thead>
<tr>
<th>Decision level</th>
<th>Decision maker</th>
<th>Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strategic</td>
<td>Health Authority</td>
<td>Budget and Surgical Mix (specialties and % of time, i.e., capacity per specialty)</td>
</tr>
<tr>
<td>Tactical</td>
<td>Hospital Management</td>
<td>Block Schedule (blocks for each specialty/surgeon)</td>
</tr>
<tr>
<td>Operational</td>
<td>Surgeons</td>
<td>Patient Schedule (scheduling of patients into a block)</td>
</tr>
</tbody>
</table>

Figure 4.2: Surgery Scheduling Process
In the first level, the budget often determines the available OR time, and there could be several factors determining the proportion of time to be assigned to each surgical specialty. For instance, waiting times (or number of patients waiting for a certain type of surgery) and seniority of surgeons might be used to define the amount of OR time required by each specialty. In the second level, the OR time assigned to each specialty is used to build the surgical block schedule, assigning days of the week and operating rooms, this taking into consideration the availability of both OR’s and post-surgery resources such as recovery beds [20, 1]. The third level has more of an operational focus. Individual surgeries are scheduled within an assigned block in the overall OR block schedule. It is at this level where one determines the number of surgeries to perform in a block, the sequence of the surgeries performed and the planned start times (appointment times) of the surgeries. It is primarily at this level that variability in surgery durations plays a key role. It is also important to note that surgeons may have surgical privileges in one or more hospitals and they are often the decision-makers that manage the third stage of the surgical scheduling process [20]. Figure 4.3 shows the decisions taken, and their usual order, in this operational level of surgery scheduling.

![Figure 4.3: The Third Level of Surgery Scheduling Process](image)

Ideally, one should consider all three decisions not in isolation but within a unified framework of analysis. However, the practical applications and mathematical challenges force practitioners and academics to work on these problems individually. For instance, even the problem of determining planned start times of surgeries is difficult on its own (e.g., see Chapter 2 and the references therein). In this chapter, we address the first decision of level 3, namely determining the number of surgeries. We do not consider sequencing (which
is a challenging problem in itself) since we will assume identical surgeries. We also do not consider appointment scheduling because our main focus is to explain the incentive issues between the surgeon and hospital in setting the number of surgeries and not their schedule. A practical justification for this is the common practice for all patients expecting a surgery on a given day to arrive at the hospital in the morning and await their surgeries. Thus, the surgeon has no idle time between surgeries scheduled in an OR block.

We consider the problem in a Canadian context, where hospitals receive funding to make ORs and supporting staff and equipment available. As explained in Blake and Donald [2], governments in Canada have historically managed the amount of healthcare services by putting rigid constraints on hospital budgets. By doing so they control hospital spending and resources and therefore indirectly the actions of physicians. In our setting, the hospital is a cost minimizer, and it receives funding (e.g., from a provincial government) to run ORs. We consider two types of cost for a hospital: idle time and overtime costs. Idle time cost may be seen as an opportunity cost of OR being idle, this is especially important in a Canadian context due to important political and social issues related to the length of surgical waiting lists [22]. Overtime costs are also significant, since the cost of operating beyond the regular OR time is often quite costly. One source of cost is in paying nurses, who are paid overtime by the hospital when they work beyond shift hours. On the other hand, the surgeon is a private entrepreneur who has privileges at the hospital and works on a fee-for-service basis. It is commonly believed in practice that surgeons tend to underestimate surgery durations and hence perform many surgeries in their allotted OR time, more than what may be ideal for the hospital. Empirical findings provides some evidence for this belief, as we show next.

Due to the randomness of surgery durations\(^2\) (Figure 4.4 and [24]), one cannot predict the precise time required for a surgery, instead it must be estimated. Usual practice is to take surgeon’s surgery duration prediction in OR bookings. In Canada, usually (if not always) it is the surgeons who keep track of the patients that require a surgical procedure, decide on the order in which they will be performed and determine the schedule of their OR block.

Anecdotal (our discussions with surgeons, anesthetists and OR booking managers, ob-

\(^2\)The data used in this chapter comes from a local hospital in Vancouver, BC. The data cover a period in 2007 and 2008 with over 5000 elective surgeries. The hospital has over fifteen surgical specialties and seven ORs.
servations in projects with hospitals and health authorities) and data based evidence ([18], Figure 4.5) suggest that surgeons are often overly optimistic about the duration of surgeries that they perform, i.e., surgeons think that they can perform surgeries quicker or they tend to underestimate their durations.

The hospital may have historical data on the surgeon and a specific type of surgery, and hospital’s OR booking manager may sometimes interfere the surgeon’s predictions if the manager thinks that surgeon is underestimating the durations. However, this is not common since “surgeons are the most mobile and least easily replaced healthcare professionals” as stated in [13]. This is due in part to the fact that surgeons are a highly mobile and scarce resource in the Canadian healthcare industry and thus wield a lot of power. In a city with multiple hospitals or private clinics their power is even more enhanced, due to their mobility between hospitals.

Figure 4.5 depicts a comparison of actual and booked/scheduled duration of surgeries. If surgery durations were perfectly estimated we would expect all surgeries be on the 45 degree line. However we see that majority (based on the data we collected, in 81%) of the cases actual durations were longer than booked/scheduled durations.

Figure 4.5 shows that duration of individual surgeries are often underestimated. One may ask how this phenomenon actually effects the daily overall performance of an OR block, i.e., amount of overtime for an OR as well as the likelihood of an OR to go overtime. To answer this question, we look at the data at an operating room level. Our data comes from a hospital with seven ORs and on average five ORs run per day. For each OR, we
compute daily average of scheduled and overtime OR minutes. We summarize our findings in Figure 4.6. The figure also shows the percentage of overtime, i.e., the ratio of overtime OR minutes and scheduled OR minutes. We see from this figure that the overtime amount is well over 20% for each OR. Total average daily overtime minutes from all ORs add up to 167 minutes.

We also find the percentage of days that each OR has an overtime to estimate the probability of daily overtime for each OR. We give these probabilities in Table 4.1. These numbers are well above 75%, suggesting that overtime for an OR is very likely.

Table 4.1: Estimates of Daily Overtime Probability per OR

<table>
<thead>
<tr>
<th>Operating Rooms</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.91</td>
<td>0.95</td>
<td>0.97</td>
<td>0.77</td>
<td>0.75</td>
<td>0.93</td>
<td>0.91</td>
</tr>
</tbody>
</table>

These empirical findings – significant amount and high likelihood of overtime – suggest that the cost of overtime can be substantial; overtime pay rates (of hospital personnel) are more expensive than regular pay rates. In addition, excessive overtime can cause job satisfaction losses, fatigue and other work-related problems with hospital employees. If an
OR can be managed in such a way that overtime is decreased then it is hoped that this will translate to immediate and significant cost savings. Additionally, savings from reduction in overtime costs may be used to increase hospital resources such as regular OR time, recovery and intensive care beds.

Then the question becomes how can we reduce overtime? Before we propose a solution, we discuss reasons for overtime. The current method of assigning surgeries to OR time works roughly as follows: the surgeon provides an OR booking manager with a list of surgeries with estimated durations. If the estimated durations are less than the allotted time then the booking manager accepts the schedule and coordinates the appropriate surgical support staff for each individual operation. Now, Figure 4.5 shows underestimation of individual surgery durations by surgeons, i.e., surgeons are quite optimistic on how quickly they perform surgeries. By underestimating the duration of surgeries, the surgical plan presented to the booking manager may contain more surgeries than can be actually accommodated in the OR block. Thus, reducing overtime crucially depends on the way that surgical plans are devised by surgeons and approved by hospital management staff.
A first step in remedying this situation is understanding why surgeons book more surgeries than they can realistically complete in a given OR block. One reason is that surgeons have a desire to serve as many of their patients as possible for altruistic reasons – a surgeon sees his patients suffer and hopes they can be healed as promptly as possible. Another more structural reason derives from the remuneration scheme for surgeons. In Canada, most surgeons are paid by the provincial government based on the number and type of surgeries they perform, irrespective of how much time each surgery takes. If we assume surgeons are profit maximizers this payment scheme puts emphasis of performing as many surgeries as possible and devalues costs associated with the overuse of hospital resources. The intuition is simple: suppose having an additional hour in the OR allows the surgeon to take on one more surgery. The surgeon may be quite willing to take on this overtime, since the benefit of performing the surgery is a fixed amount that may be worth significantly more than the disutility of one hour of working overtime for the surgeon. This may often be the case, especially since the surgeon need not consider the overtime costs of support staff and materials when deciding if working one more hour is desirable. We argue that the optimism we see in the data regarding the estimated duration of surgeries reflects this incentive, since it directly effects the number of surgeries they are able to book and execute in their block.

Hospitals have an interest in influencing surgeons to perform fewer surgeries and more accurately estimate their durations, since this would mean less overtime costs. This raises two important considerations for the hospital. The first is to decide themselves how many surgeries they would prefer be performed in the OR block. It is reasonable to assume that hospital would be better off with a surgeon who optimizes the use of resources (and the most important resource is the OR time) rather than one who is interested in profit maximizing, i.e., hospital’s ideal number of surgeries may be less than the profit maximizer surgeon’s number. We propose that the hospital should determine its own ideal number of surgeries in order to minimize its own cost.

Thus leads us, however, to the second key issue: how can the surgical booking procedure be adjusted so that hospitals can influence surgeons to decide on a surgery plan that better reflects the costs of the hospital. Indeed, the surgeon may not cooperate with being dictated to doing fewer surgeries (e.g., she/he has a better outside alternative) and thus the hospital must consider how to design a contract that will be accepted by the surgeon and also save
money for the hospital (with respect to current “high-overtime” situation).

In this chapter, we characterize analytically the number of surgeries that minimizes hospital costs, find conditions when this number is less than surgeon’s preference, and suggest contracts to remedy this misalignment between the hospital and surgeon on determining the number of surgeries in a given OR block. We consider the surgeon as an agent and the hospital as a principal and use a simple principal-agent model of analysis common in applied economics. (For a review of principal-agent theory we refer the reader to [9] and the references therein.)

In principal-agent theory, when there is no information asymmetry between players the first best (i.e., best possible outcome) can be achieved with a ”forcing contract”, i.e., the principal can dictate the agent what to do (i.e., how many surgeries to be performed). With or without information asymmetry between the players, a residual claimancy contract can be used to achieve the first best if the agent is risk neutral and such a contract is feasible (e.g., there is a single agent who has unlimited wealth). We can think of a residual claimancy contract in our setting as the hospital renting the operating room to the surgeon, at the hospital’s opportunity cost of the room. The surgeon then makes his or her scheduling decision having internalized all costs and benefits of the decision. The incentive problem is completely resolved.

In our analysis, both players are risk neutral (although we study briefly the case of a risk averse surgeon in Section 4.6.2), and we consider two cases regarding the information asymmetry: no information asymmetry (hospital has access to surgery duration distribution) and information asymmetry (the hospital has access to only the mean of surgery durations).

We propose two payment schemes that will achieve the first best depending on how much information that the hospital has on surgery durations. If the hospital has access to only the mean duration of surgeries then it may choose a three-part contract (Section 4.5.3), and if the hospital has access to entire distribution of surgeries then it may choose either take-it-or-leave-it offer at the optimal number of surgeries or a three-part contract (Section 4.5.2). The three-part contract can be seen as a residual claimancy contract whereas take-it-or-leave-it offer can be thought as a forcing contract.
4.3 The Model

We start with a short description of our notation and define the objective functions of the hospital and the surgeon. We will make several important assumptions to refine our model, help focus on the incentive issues involved and avoid over-complication. Each assumption will be discussed and motivated, and the more restrictive assumptions will be noted.

As discussed, the scenario is a surgeon working out of a hospital by using its OR facilities and support staff. The hospital receives funding (e.g., from a provincial government) to make its operating rooms available for surgeries at minimum possible cost. The surgeon works in an operating room reserved (determined by an OR block schedule) for her/his use in the hospital. The scheduled time, i.e., the length of the OR block allocated for this surgeon will be denoted $d$ and is given exogenously in our model.

The surgeon is paid a fee-for-service rate of $r$ dollars per surgery directly by the provincial government. It is important to stress that we assume that the surgeon is not directly paid by the hospital (as in the Canadian health care system). Indeed, this is a distinctive feature of our model as compared to a classical principal-agent framework where the principal compensates the agent.

The surgeon decides the number $n$ of surgeries to schedule during her/his allotted time. We assume that there is a long list of people waiting to receive the given surgery, and thus no shortage of demand for operations. This is quite reasonable under most (if not all) types of (elective) surgeries [21]. The fact the surgeon chooses $n$ and not an “effort level” is another distinctive feature in our setting which is not considered in standard principal-agent problems. Here we may think of the surgeon’s choice as a rough proxy for effort. The number of surgeries $n$ is the key decision variable, and exploring precisely how it is determined is the distinguishing feature of our analysis.

As is common in practice, we assume that every surgery scheduled must be performed on the scheduled day even if this causes the total duration of all $n$ surgeries to exceed $d$. An important extension of our analysis, which is not addressed here, would be to consider the possibility of cancelations after a certain cut-off time.

Each surgery $i$ has a random duration $t_i$. We assume that the support of the pdf is contained in the positive real line $\mathbb{R}^+$ and each has identical finite mean $\mu$. We also assume that the random variables $t_i$ are independent. Let $T(n) = \sum_{i=1}^{n} t_i$ denote the random
duration of $n$ surgeries. It is a random real-valued function of $n$. The (random) overtime of $n$ surgeries can thus be expressed as $\max\{0, T(n) - d\}$ and similarly $\max\{0, d - T(n)\}$ represents the (random) idle time.

The above two assumptions – that of identical mean for each surgery and independence – are worthy of further discussion. First, by assuming identical means we may think of the surgeon scheduling $n$ elective surgeries of a similar type; for instance, all hernia operations. We see this practice in certain specializations of surgeons, e.g., ophthalmology. Another motivation for this assumption is to simplify the model to avoid consideration of surgery sequence. If surgeries have varying means the question of sequence becomes paramount and the problem becomes more combinatorial in nature. Nonetheless, since we assume that surgeries may have different distributions (under the condition they have the same mean) then sequence is still an issue. For instance, a schedule of five “high” variance surgeries will have different properties from a sequence of “low” variance surgeries. Thus, we assume that the sequence is given and the surgeon simply chooses $n$ consecutive elements of that sequence. An alternative assumption is that the $t_i$ are independent and identically distributed (iid) in which case sequence is irrelevant. The results for this case are essentially identical to those found here and so we adopt the former assumption. In either case, the goal is to focus on the incentives which drive the choice of the number of surgeries and avoiding extraneous complexities at this point.

Second, we address the assumption of independence of surgery durations. Indeed, one might argue that surgeon fatigue creates a dependence in surgery durations, and this is a relevant criticism of our model. By assuming independence we effectively assume that all variation in surgery duration depends on the specifics of each surgery case. We can derive similar results to those found here under the assumption that all have the same (random) duration $t$, in other words, there is complete dependency among surgery durations. Thus, the total duration of $n$ surgeries performed in one day by the surgeon has random value $nt$. The results we derive in this setting are similar in spirit to those discussed below and yield many of the same general findings. However, our approach to the analysis is different and we believe of separate interest. Details are included in Section 4.6. Thus, our analysis covers the extreme cases of independence and complete dependence, and thus one might imagine that similar insight might arise for intermediate cases.

We start our analysis by assuming both the hospital and surgeon are risk neutral (we
extend our analysis with a risk averse surgeon in the case where all surgeries have the same duration $t$ in Section 4.6). The hospital is not-for-profit and closer to being a public rather than a private organization. Therefore we believe risk-neutrality for the hospital is a reasonable assumption. We begin by assuming that the surgeon is risk neutral for simplicity of our arguments and analytical tractability. One reason this assumption might make sense is that the surgeon performs surgeries on many days during a month or year, therefore the profit deriving from a single day is small in comparison to her overall compensation. On the other hand, since our model concerns the decisions of a surgeon for a single day, it is reasonable to consider that a surgeon might be risk averse.

We assume that the hospital’s expected cost function of opening an OR room to a surgeon to use for duration $d$ is a function of the the form

$$C(n) = E_{T(n)}[o_H \max\{0, T(n) - d\} + u_H \max\{0, d - T(n)\}]$$

where $E_{T(n)}[\cdot]$ denotes the expectation operator on the random variable $T(n)$. The cost coefficient $o_H$ is the cost per unit time of going overtime after regular working duration $d$, whereas the cost coefficient $u_H$ is the cost per unit time of idle time cost of the OR. We can think of $o_H$ as the overtime cost rate of the hospital, e.g., staffing and equipment costs after regular working hours. On the other hand, $u_H$ may be thought as the opportunity cost of an idle OR. Besides facility and operating costs, this cost may include a component to reflect the utility loss of patients waiting for a surgery (especially when there are long surgical waiting lists as in Canada). This, however, is modeled more directly when we consider welfare considerations in Section 4.5. Underage costs may also be seen to include costs related to the possibility that unused capacity might motivate budget cuts to the hospital from the provincial government. Note that our cost function does not track normal operating costs for running the operating room within the scheduled $d$ hours and in particular there is no direct per-unit cost incurred by the hospital per surgery. Since the hospital will incur these costs in any instance, we focus on the problem of minimizing the costs related to the over- and under-utilization of resources.

This cost function is reminiscent of standard newsvendor costs having a cost for overage and underage. The difference here is that in the standard newsvendor setting, demand is random and the newsvendor chooses capacity. Here the situation is reversed: the capacity $d$ is fixed and the choice variable $n$ impacts demand on that capacity (in this case, OR
Similar models have been explored in the literature, most notably in the “inverse newsvendor” model in [4]. The difference in our model is that demand is not chosen directly, but through the choice of $n$, which determines the number of random values which amount to total demand.

Using linearity of expectation and the identity

$$x = \max\{0, x\} - \max\{0, -x\}$$

we can express $C(n)$ as:

$$C(n) = -u_H \mu n + (o_H + u_H)E_T(n)[\max\{0, T(n) - d\}] + u_H d. \quad (4.1)$$

This form is more useful in the analysis that follows in Section 4.4.

The expected profit of the surgeon for performing $n$ surgeries in an operating room scheduled for duration $d$ is assumed to be of the form

$$\pi(n) = rn - E_T(n)[o_S \max\{0, T(n) - d\} + u_S \max\{0, d - T(n)\}]$$

The quantity $o_S$ is the cost per unit of time of performing surgeries for hours in excess of scheduled surgery time $d$. This can represent an opportunity cost for the surgeon to work outside of scheduled hours, possibly reflecting alternate sources of income or leisure time. The quantity $u_S$ is the cost per unit time worked less than the scheduled time, and can represent lost revenues from surgeries that might have been scheduled and indirectly loss of goodwill amongst patients to the surgeon for longer wait times. A similar transformation of above yields the following more amenable form of $\pi(n)$:

$$\pi(n) = (r + u_S \mu)n - (o_S + u_S)E_T(n)[\max\{0, T(n) - d\}] - u_S d \quad (4.2)$$

One important feature of the expected profit function of the surgeon is that when the total duration of the surgeries is precisely $d$, the surgeon experiences no costs. The same is true for the hospital, making $d$ a significant value for both the hospital and surgeon. Clearly, this is a special case of a more general setting where we might imagine that the surgeon experiences costs when total duration is different from some other value, say $l$, where $l \neq d$. We assume for analytical tractability that $l = d$, although this assumption might not hold in practice. Considering how these results extend to the case $l \neq d$ is one possible direction for future study.
Clearly, all the cost coefficients $o_H$, $u_H$, $o_S$ and $u_S$ can be challenging to estimate, which is a common problem of models of this type. We assume that hospital knows all these cost coefficients and the surgeon knows only his or hers. This is, undoubtedly, a strong assumption of this model. Finally, we assume that all cost coefficients and the fee-for-service rate $r$ are nonnegative.

4.4 Misalignment of Incentives

We now discuss the process of determining the number of surgeries to perform in time $d$ under the assumptions stated above. Our goal is to understand why surgeons tend to underestimate surgery durations and hence schedule more number of surgeries than what might be ideal for hospital in their allotted time by demonstrating that the ideal number of surgeries for the surgeon is (under some stated conditions) larger than the preferred number of the hospital, which needs to take into account overtime costs. We now proceed with the analysis.

4.4.1 Deciding the Number of Surgeries

First we focus on the decision of the surgeon. Let $n_S$ be the preferred number of surgeries scheduled by the surgeon when unrestricted by the hospital. In other words, $n_S$ is chosen to optimize the profit function $\pi$. We begin describing the properties of the surgeon’s profit function $\pi$. For our first result we need the following definitions. The first definition concerns convexity properties of $\pi$. We treat $n$ as an integer variable, and thus use the following notion of discrete convexity:

**Definition 4.4.1.** Let $f : \mathbb{Z} \rightarrow \mathbb{R}$. The first differences of $f$ are denoted $\Delta f(n) = f(n + 1) - f(n)$ and second differences by $\Delta^2 f(n) = \Delta f(n + 1) - \Delta f(n)$. Then $f$ is discretely convex if its first differences are nondecreasing or equivalently if its second differences are nonnegative, i.e., $\Delta^2 f(n) \geq 0$ for all $n \in \mathbb{Z}$. We say $f$ is discretely concave if $\Delta^2 f(n) \leq 0$ for all $n \in \mathbb{Z}$.

The second definition concerns the distribution of surgery durations and is due to [16]:

**Definition 4.4.2.** A random variable $X$ is new better than used in expectation (NBUE) if $E[X] \geq E[X - k|X \geq k]$ for all $k$. 

106
Proposition 4.4.3 (Discrete concavity of $\pi$). The surgeon’s profit function $\pi$ is discretely concave when $t_i$ is NBUE for all $i$.

Having established the discrete concavity of $\pi$ we use the following necessary and sufficient condition for $n_S$ to be integer optimal:

$$\pi(n_S) \geq \pi(n_S - 1) \text{ and } \pi(n_S) \geq \pi(n_S + 1),$$

i.e., $\Delta \pi(n_S - 1) \geq 0$ and $\Delta \pi(n_S) \leq 0$.

Necessary and sufficient conditions for the optimality of $n_S$ in terms of the cost data of the surgeon now follow. We use the following convenient notation. Recall that $O(n) = \max\{0, T(n) - d\}$ is the (random) overtime for scheduling $n$ surgeries. Let $\theta(n) = E_{T(n)}O(n)$ be the expected overtime. Then, by using Eq(4.2) we may write $\pi(n) = (r + u_S\mu)n - (o_S + u_S)\theta(n) - u_Sd$. Thus, $n_S$ is optimal for $\max_{n \geq 0} \pi(n)$ if and only if:

$$\Delta \theta(n_S - 1) \leq \frac{r + u_S\mu}{o_S + u_S} \leq \Delta \theta(n_S). \quad (4.3)$$

The condition for optimality in Eq(4.3) is reminiscent of the classic newsvendor solution based on critical fractiles. We can interpret $r + u_S\mu$ as the expected marginal benefit of undertaking an additional surgery. On the other hand, $(o_S + u_S)\Delta \theta(n_S)$ may be interpreted as the marginal expected overtime cost of an additional surgery. Thus, Eq(4.3) says that at the optimal choice of $n_S$ the expected marginal cost and benefit of surgery $n_S$ must be comparable (indeed, if $n$ is allowed to be continuous then they must be equal).

Turning now to the hospital’s decision, let $n_H$ be the preferred number of surgeries scheduled by the hospital when it knows the surgery duration mean $\mu$ of $t$ and can force the surgeons to perform the number of surgeries it prefers. In other words, $n_H$ is chosen to minimize the cost function $C$. Our goal is to provide optimality conditions for $n_H$ similar to Eq(4.3). The following result describes the convexity properties of $C$.

Corollary 4.4.4 (Discrete convexity of $C$). The hospital’s cost function $C$ is discretely convex when $t_i$ is NBUE for all $i$.

Since $C$ is discretely convex we use the following optimality conditions to characterize $n_H$:

$$C(n_H) \leq C(n_H - 1) \text{ and } C(n_H) \leq C(n_H + 1),$$

i.e., $\Delta C(n_H - 1) \leq 0$ and $\Delta C(n_H) \geq 0$.
Next we state the necessary and sufficient conditions for the optimality of \( n_H \) in terms of the cost data of the hospital:

\[
\Delta \theta(n_H - 1) \leq \frac{u_H \mu}{o_H + u_H} \leq \Delta \theta(n_H). \tag{4.4}
\]

This condition can be similarly interpreted as above. Note that the value \( u_H \mu \) can be interpreted as the marginal expected benefit of undertaking an additional surgery (assuming the total duration with the additional surgery does not exceed \( d \)), which does not depend on the \( r \).

We now turn to one of the motivating questions of this study: how do the incentives of the hospital and surgeon interact and is there a misalignment of these incentives? We give conditions whereby there truly is a misalignment of incentives between the parties, and attempt to explain why this misalignment arises. The main result describes conditions where \( n_S \geq n_H \); in other words, when each party has a different optimal number of surgeries and the surgeon prefers to perform more surgeries than the hospital. The surgeon’s objective is to maximize his/her profits whereas the hospital’s objective can be thought as utilizing the OR time as much as possible, i.e., minimizing the cost of idle and overtime.

**Theorem 4.4.5.** The optimal number of surgeries from the hospital’s perspective \( n_H \) is less or equal than the surgeon’s preferred number of surgeries \( n_S \), i.e.,

\[ n_H \leq n_S \]

if and only if

\[ \frac{r + u_S \mu}{o_S + u_S} \geq \frac{u_H \mu}{o_H + u_H}. \tag{4.5} \]

The resulting condition Eq(4.5) has a straightforward interpretation. We may think of \( \rho_S = \frac{r + u_S \mu}{o_S + u_S} \) as the ratio of expected marginal benefit for the surgeon to perform a surgery to per-unit-time cost of overtime. A similar interpretation holds for the hospital’s ratio \( \rho_H = \frac{u_H \mu}{o_H + u_H} \) where \( u_H \mu \) is the impact on cost when one more surgery is scheduled and \( o_H + u_H \) represents a per-unit-cost of overtime. Thus when the “marginal ratio” \( \rho_S \) of the surgeon exceeds that of the hospital \( \rho_H \) then the surgeon schedules more surgeries than the hospital prefers. Since in practice it is likely that \( n_S \geq n_H \) this indicates that the marginal ratios of the surgeon and hospital satisfy \( \rho_S \geq \rho_H \).

It is reasonable to assume that in practice, hospitals are less sensitive to undertime than overtime, i.e., \( u_H \leq o_H \), and surgeons are more sensitive to undertime than overtime, i.e.,
\( u_S \geq o_S \). This observation makes easier to see why \( \rho_S \geq \rho_H \) holds and surgeons’ preferences on number of surgeries may be higher than hospitals’ choice.

### 4.5 Contracts

We now turn to the question of how we might address the misalignment of incentives described in the previous section. Thus, we turn from explaining some of the observations detailed empirically in Section 4.2 towards considering ways to reduce over-use of operating room resources. This can be achieved through the alignment of incentives of the hospital and surgeon via designing mechanisms or contracts. The type of mechanism required to align incentives depends on several important factors. We discuss them briefly.

One important factor in designing mechanisms is the amount of information that each of the players have. There are several types of information involved in this problem, which are essentially knowledge about the cost coefficients, surgery durations and functional form of the utilities. As above we assume both the hospital and surgeon have common knowledge of the mean surgery duration \( \mu \) and the functional form of the utilities of each player, as well as their own cost coefficients. One thing that differentiates the later scenarios we discuss is whether the hospital has complete information about the distributions of the surgeries. In all of our models we assume that the hospital has knowledge of the surgeon’s cost coefficients.

A second important factor is the degree to which the hospital can monitor the actions of the surgeon. In other words, whether the hospital can observe \( n \), the number of surgeries booked by the surgeon, and also the overtime and idle time. We are aware that in some hospitals the surgeon must present their schedules to an OR manager, who observes \( n \) and then schedules the support staff and equipment for the surgeon. Nonetheless, one might well imagine a scenario where hospital management is less informed as to the number of surgeries. We assume here that the hospital can always observe the number of surgeries planned by the surgeon and any idle or overtime. This is a reasonable assumption in any well-run hospital.

A third important factor is consideration of whether a third party – possibly a government or health authority – has some control over the design of the contract. Indeed, one factor missing in the discussion to this point is a very important one – the effective
treatment of patients. We have mentioned the possibility that the hospital or surgeon’s concern for patients can be captured in their cost coefficients, but this is a rather indirect way to understand the impact on patient care. At the end of this section we explore welfare considerations that attempt to look explicitly at the impact on patient care and the overall efficiency of the system.

In the following subsections we describe several types of contracts that arise under different assumptions on information, monitoring and power, and the role of government.

4.5.1 Hospital has Complete Information and Coercive Power

The best situation from the point of view of the hospital is when the surgeon performs $n_H$ surgeries. However, the requirements to ensure this outcome are quite strong. First of all, in order to compute $n_H$ the hospital needs full knowledge of the distribution of the $t_i$’s. In particular, we would need to know the distribution of $T(n)$ for each $n$, and this depends on the joint distribution of $t_1, \ldots, t_n$. As we can see this is a strong informational requirement for the hospital. Nonetheless, we may assume this to be the case, since the hospital has access to historical information about the surgeries, possibly at least as much information as the surgeon does. Certainly, the hospital tracks OR usage by various physicians and likely tracks surgery types and durations (such as in the data set we used in Section 4.2). Nonetheless, one might argue that the surgeon herself/himself has private information about the specifics of each case which is independent of the historical information and is thus not available to the hospital. This represents an information asymmetry between surgeon and hospital. This issue is partially addressed below in another contracting scenario. Nonetheless, studying further how asymmetry of information would impact our results is an area for future research.

The other factor, besides information, that may prevent the surgeon from taking the hospital’s recommendation of $n_H$ surgeries is that the surgeon may have some power in determining how many surgeries get scheduled. In general, the surgeon has a best outside alternative to performing surgeries at the hospital in question, which yields her some level of utility $\pi_0$, which we can safely assume is less than $\pi(n_S)$ (since otherwise our surgeon should quit!). We assume that this outside alternative is common knowledge to both the surgeon and the hospital, and it may, for instance, be the option that a surgeon can work in some other hospital or possibly a private medical clinic. If $\pi_0 \leq \pi(n_H)$ then the surgeon would be
willing to perform $n_H$ surgeries if granted use of the operating room, because her/his next best alternative leaves her/him worse off, and so the hospital achieves its desired number of surgeries. In the other case, i.e., $\pi_0 \geq \pi(n_H)$, the situation is more complicated. If the hospital has the power to force the surgeon to perform exactly $n_H$ surgeries, then the hospital is most happy, whereas the surgeon is less well off compared to his/her alternative. This is only a sustainable option if the hospital has strong coercive power.

4.5.2 Take-It-or-Leave-It Offer

We now suppose that the hospital cannot coerce the surgeon into performing $n_H$. This is observed in practice when the skills of a surgeon are in great demand. As mentioned above, “surgeons are the most mobile and least easily replaced health care professionals” [13], and we assume that the hospital has a shortage of surgeons and cannot easily replace one surgeon with another. Furthermore, suppose that $\pi(n_H) < \pi_0 \leq \pi(n_S)$ and hence the surgeon’s most profitable activity is to perform surgeries at the given hospital, just not as few as $n_H$ surgeries.

We ask the following basic question: Can the hospital offer some incentive to induce the surgeon to perform fewer surgeries than $n_S$ in a way that is cost-effective for the hospital? We answer this question in two settings. The first, described in this subsection, is where the hospital retains complete information about surgery durations but no longer has coercive power. The second setting, described in the next section, is where the hospital no longer has complete information about surgery durations and must induce the surgeon to make an appropriate choice simply through adjusting her/his compensation. In both cases we assume that the hospital can monitor the choice of $n$ by the surgeon, knows the surgery expected duration $\mu$, and knows the surgeon’s cost coefficients $u_S$ and $o_S$.

The first setting can be modeled as a simple bilateral externality, a standard model in the microeconomics literature (see Chapter 11.B of [17]). The overtime and idle time costs of the hospital are influenced by the decision of the surgeon, who need not consider these costs when making her/his optimal choice. As we showed in the previous section, if the surgeon is unconstrained in her/his choice, she/he opts for $n_S$ surgeries, which is usually not optimal for the hospital. The hospital experiences a loss of $C(n_S) - C(n_H)$ from its optimum under this scenario. Assuming, as we do here, that the surgeon has the right to schedule surgeries as she/he sees fit (i.e., the hospital has no coercive power of the surgeon’s
decision), the hospital will need to offer some compensation $B > 0$ to induce the surgeon to adjust her/his surgeries. The surgeon will agree to performing $n$ surgeries if and only if $\pi(n) + B \geq \pi(n_S)$. Since we may assume that the hospital will offer the smallest bonus possible to achieve the reduction of surgeries to $n$ in number, we have $B = \pi(n_S) - \pi(n) \geq 0$. Thus, the cost of the surgeon under this bonus scheme is precisely:

$$\min_{n \geq 0, B} \{C(n) + B : \pi(n) + B \geq \pi(n_S)\} = \min_{n \geq 0} C(n) - \pi(n) - \pi(n_S) \quad (4.6)$$

Let $n_G \in \arg \min_{n \geq 0} \{C(n) - \pi(n)\}$ denote an optimal solution to the above problem. Note that this can seen as a kind of socially optimal choice, since it maximizes the overall welfare $\pi(n) - C(n)$ of the two parties (of course, it does not consider the direct impact on patients, which is discussed below). Since the bonus $B = \pi(n_S) - \pi(n_G)$ compensates the surgeon sufficiently, the surgeon will always choose to perform $n_G$ surgeries and take bonus $B$.

To clarify, the end result of this analysis is that the hospital computes $n_G$ based on information at its disposal by solving Eq(4.6). Then the offer to surgeon is simple: if the surgeon schedules $n_G$ surgeries, s/he receives a compensation of $B$ dollars from the hospital. Otherwise, the surgeon is free to determine the number of surgeries s/he prefers but will not receive the bonus (however, again the bonus is defined so this latter case never occurs). One may see this as a “top-down” approach to surgery scheduling, all the computational work (computing $n_G$ and $B$) is undertaken by the hospital and the surgeon’s choice is straightforward.

### 4.5.3 Three-Part Contract

The contract in the previous subsection was predicated on the assumption that the hospital has complete information about the duration of surgeries. In addition, the hospital needs to undertake computation of the preferred number of surgeries and bonus, with the possibility (due to inaccurate calculation) that the take-it-or-leave-it offer may still be rejected. As mentioned previously, it is probable that the surgeons have some private information about the duration of surgeries due to their personal knowledge of the patients and their histories. Thus, it may be preferable to consider a “decentralized” contract which attempts to align the incentive of the surgeon to that of the hospital. In other words, the hospital may design a compensation scheme whereby the surgeon’s decisions themselves weigh the importance of the hospital’s cost structure.
This can be achieved by the following three-part contract, which is specified up to some policy parameter we denote as \( \alpha > 0 \). The three parts to the contract are as follows:

1. a fixed sum \( B_\alpha \) which passes from hospital to surgeon

2. a surgery unit cost \( \gamma_\alpha \) which is charged by the hospital for each surgery booked by the surgeon

3. a per-unit time overtime penalty \( \omega_\alpha \) which is charged by the hospital for each unit of overtime incurred by the surgeon.

Ranges for values of these parameters which achieve an alignment of incentives are discussed below. Note that to administer this contract the hospital needs to be able to monitor the actions of the surgeon. Indeed, to charge the surgery unit cost, the number of surgeries needs to be observed, and overtime fees can only be calculated if the hospital carefully monitors overtime. This latter monitoring is, of course, should already be a practice of the hospital since they need to compensate nurses for overtime and thus have an interest in monitoring this quantity.

One interpretation of the contract is as follows. The surgeon is given some budget \( B_\alpha \) and she/he can use this budget to rent time for a surgery at rate \( \gamma_\alpha \) and is penalized for overuse of resources at a rate of \( \omega_\alpha \) per unit time. Thus, we see that the cost implications of a surgery for the hospital are in some sense passed to the surgeon, and she is in turn compensated at the budget level \( B_\alpha \) to defray these cost considerations and still remain interested in performing surgeries. The payoff of the surgeon on this contract will be

\[
\pi_\alpha(n) = B_\alpha - \gamma_\alpha n - \omega_\alpha E_T(n) \left[ \max\{0, T(n) - d\} \right] + \pi(n)
\]

We now proceed in specifying the three elements of the contract, which are related to the choice of parameter \( \alpha \), for which the hospital can ensure the surgeon can perform \( n_H \) surgeries. Of course, the surgeon will need to be compensated in order to participate. The participation constraint of the surgeon is given by

\[
\pi_\alpha(n) \geq \pi_0
\]

where \( n \) is the optimal number of surgeries for the surgeon under a contract with parameter \( \alpha \).
The values of the three contract parameters which can align incentives are as follows:

\[ \gamma_{\alpha} = r + (u_S - \alpha u_H) \mu, \ \omega_{\alpha} = \alpha (o_H + u_H) - (o_S + u_S) \] and any fixed sum \( B_{\alpha} \) which lies in the range \( B_{\alpha} \geq \pi_0 + (u_S - \alpha u_H)d + \alpha C(n_H) \). These values are chosen so that the surgeon’s profit has the form

\[ \pi_{\alpha}(n) = B_{\alpha} - (u_S - \alpha u_H)d - \alpha C(n). \]

which is simply a linear transformation of the hospital’s objective \( C(n) \). It is then straightforward to see that a surgeon facing profit function \( \pi_{\alpha}(n) \) will choose \( n = n_H \). This is precisely that number of surgeries which minimizes the hospital’s costs, and the hospital has achieved its goal. The surgeon will participate in this three-part contract for any value \( \alpha > 0 \) since the surgeon’s profit will be

\[
\begin{align*}
\pi_{\alpha}(n_H) &= B_{\alpha} - (u_S - \alpha u_H)d - \alpha C(n_H) \\
&\geq \pi_0 + (u_S - \alpha u_H)d - (u_S - \alpha u_H)d + \alpha (C(n_H) - C(n)) \\
&\geq \pi_0
\end{align*}
\]

The result derives from the bound which we put on the fixed sum: \( B_{\alpha} \geq \pi_0 + (u_S - \alpha u_H)d + \alpha C(n_H) \).

A few comments are in order about this contract. First, note that the variable fees \( \gamma_{\alpha} \) and \( \omega_{\alpha} \) can be determined without knowing the full surgery time distributions. Thus under this contract, the hospital can ensure \( n_H \) surgeries are performed without full information about those surgeries. Thus, in contrast to the take-it-or-leave-it offer described in the previous subsection, the computational burden now rests with the surgeon and not the hospital. We can thus see this as a “bottom-up” approach to surgery scheduling – the hospital passes the necessary information to the surgeon via the three components of the contract, and the surgeon is left to decide.

One further consideration is necessary here, which is whether the hospital benefits from offering these contracts. Note that the total cost to the hospital is now:

\[
C_{\alpha}(n_H) = B_{\alpha} + C(n_H) - \gamma_{\alpha}n_H - \omega_{\alpha}E_{T(n)}[\max\{0, T(n_H) - d\}]
\]

We can rewrite \( C_{\alpha}(n_H) \) by plugging in the values of \( \gamma_{\alpha} \) and \( \omega_{\alpha} \) as

\[
C_{\alpha}(n_H) = B_{\alpha} + (1 - \alpha)C(n_H) + \pi(n_H) + (\alpha u_H - u_S)d.
\]
The hospital benefits from the new scheme if $C_\alpha(n_H) \leq C(n_S)$ where we assume $n_S$ surgeries are performed if no intervention is made. This implies an upper bound on the bonus, i.e.,:

$$B_\alpha \leq C(n_S) - (1 - \alpha)C(n_H) + \pi(n_H) - (\alpha u_H - u_S)d.$$  

The possible values of $B_\alpha$, i.e., $\pi_0 + (u_S - \alpha u_H)d + \alpha C(n_H) \leq B_\alpha \leq C(n_S) - (1 - \alpha)C(n_H) + \pi(n_H) - (\alpha u_H - u_S)d$ yield a range of fixed-fee compensations that yield feasible contracts. We note that the bonus could be determined through bargaining by the two parties, and its value would fall somewhere between these bounds.

### 4.5.4 Implementing the Two Contracts

We now discuss briefly some of the challenges that might be faced when implementing either of these two contracts. First of all, there is an important challenge in specifying the parameters of the model. Specifically, information on the cost parameters of both the surgeon and hospital may not be explicitly known, and particularly not to both parties (which is assumed under both contracts). The task of determining $o_H$ and $u_H$, because it deals with the overall costs of the hospital, may present challenges. Indeed, there may be lack of consensus on the values of these parameters amongst the various decision-makers in the hospital. Overtime costs, nonetheless, seem more accessible than idle time costs. Overtime costs may be approximated by direct costs of staffing overtime wages. Idle time costs are more indirect, and take into account lost value from idle resources. To implement either contract we foresee that important discussions would need to held to establish consensus on the values of these parameters.

A second issue, which is mostly unrelated to specifying the parameters of the model, is that of the political feasibility of adopting these contracts. One attractive feature of the take-it-or-leave-it offer is that the contract is relatively simple to understand and has the appearance of a “win-win” situation. The surgeon receives a bonus for performing $n_G$ surgeries and there are no fees involved. On the other hand, since most of the computation in this setting rests with the hospital, surgeons may feel disempowered in having the hospital in some sense decide the preferred surgery level. The potential disutility that may arise due to a sense of disempowerment is not covered by our model, but may be a consideration in practice.

On the other hand, in the three-part contract surgeons retain their decision-making
role. However, the downside here is that the surgeon now experiences fees and penalties for overtime, and so it seems less a clear “win-win” situation than the alternate contract. The three-part contract proposes to treat surgeons much like independent entrepreneurs who must rent and pay for overuse of facilities and resources, which seems a less advantageous setup than the current situation where surgeons retain “privileges” in the operating room. Thus, there may be some disutility deriving from this perceived loss of privilege that is not considered in our model but again it may be significant in practice.

4.5.5 Welfare Considerations

One element missing from the above analysis is a concern for the welfare of patients. When isolating attention on the incentives of hospitals and surgeons, it is possible that patients are adversely affected. In the Canadian healthcare system another agent, the provincial government, is responsible for the health care system as a whole. They are interested in balancing the interests of patients, hospitals and physicians. We assume that the provincial government is a social welfare maximizer with the following utility function:

\[ W(n) = \pi(n) - C(n) + \delta n \]

where \( \delta \) is a measure of the per unit “social value” of a performed surgery. This is consistent with earlier assumptions in our model that the surgeries are elective surgeries of a similar type, hence having identical mean \( \mu \) and remuneration \( r \). It is true that some surgeries may have more social value than others; for instance, saving the life of a child by a surgery may carry more social value than saving someone with many other health complications and whose quality of life after the surgery would only be marginally improved. This is of course an ethical discussion and value judgement. We avoid such discussions, and take the view that the provincial government is not privy to the details of each individual case and thus takes \( \delta \) as their valuation of each surgery.

Letting \( u_W = u_H + u_S \) and \( o_W = o_H + o_S \) we can rewrite \( W(n) \) as:

\[ W(n) = (r + \delta + u_W \mu)n - (o_W + u_W)E_{T(n)}[\max\{0, T(n) - d\}] - u_W d \]

As above, \( W \) is discretely concave, and an optimal number of surgeries from a social welfare perspective \( n_W \) satisfies:

\[ \Delta \theta(n_W - 1) \leq \frac{r + \delta + u_W \mu}{o_W + u_W} \leq \Delta \theta(n_W). \] (4.7)
Note that $n_W$ can be bigger, smaller, or equal to $n_H$ and $n_S$ depending on the cost parameters. For instance, if the central planner places high value on surgeries (for instance, due to political pressures) $\delta$ may be large enough so that $n_W$ is in fact greater than $n_S$. It is straightforward to find bounds on $\delta$ that guarantee this to be the case.

The case with the most intuitive appeal is where $n_H \leq n_W \leq n_S$, which indicates that surgeons do more surgeries than socially optimum, and hospitals hope to perform fewer surgeries than socially optimal value. To align the surgeon’s incentives and induce the cooperation of the surgeon, the central planner could again offer a three-part contract similar to the one above. The major difference is that the contract needs to ensure participation of both the surgeon and hospital in this case. Also, different contracts arise if revenues from overuse of the facilities or per-unit surgery charges can either accrue to the hospital or to the central planner directly. Precise details are omitted but the analysis follows very similar reasoning to that in Section 4.5. A social planner, the provincial government in our case, can use Eq(4.7) with their estimates $\delta$ to judge if and how they would like to intervene via designing a contract to manage the relationship between hospitals and surgeons.

### 4.6 Dependent Surgeries with Identical Realizations

In this section we return to one of the important assumptions in our model, that of independent surgery durations. Here we assume that surgery durations are fully dependent and given by the outcome of a single random variable. Although this setting is also restrictive, it can be seen as the opposite extreme of the independent case. Of interest is the fact that we can obtain similar results, and thus under both models the conclusions and insights are similar. This suggests that the findings of our analysis could apply to intermediate cases or surgery duration dependence.

Another reason for considering this case is that in this framework we are able to say something about risk aversion. Using the previous model we were unable to establish results in the case of risk aversion, and this can be remedied here.

In this section, we assume that each surgery has the same random duration $t$, a random variable with probability distribution function (pdf) $f$ and cumulative density function (cdf) $F$. We assume that the support of the pdf is contained in the positive real line $\mathbb{R}^+$. Thus, the total duration of $n$ surgeries performed in one day by the surgeon has random value $nt$. 
Furthermore, we assume that \( n \) is a continuous decision variable and can take any value in \( \mathbb{R}^+ \). This is an abstraction from reality, where only an integer number of surgeries ought to be considered. However, this is not a restrictive assumption in this case since we have a single dimensional decision variable \( n \) and one can always take \([n]\), i.e., round down, or \([n]\), i.e., round up, after the analysis and choose the better one.

### 4.6.1 Preliminaries and Misalignment of Incentives

The cost function of the hospital and the profit function of the surgeon are defined as before, the only difference is that the total duration is now \( nt \) and \( E_t[\cdot] \) denotes the expectation operator on the random variable \( t \). As before, \( \mu \) is the mean of \( t \). We assume that the hospital and surgeon have the following objectives – the cost of the hospital

\[
C(n) = E_t[o_H \max \{0, nt - d\} + u_H \max \{0, d - nt\}] \tag{4.8}
\]

and the profit of the surgeon

\[
\pi(n) = rn - E_t[o_S \max \{0, nt - d\} + u_H \max \{0, d - nt\}] \tag{4.9}
\]

In this section we assume that hospital knows \( f \) and all the cost coefficients whereas surgeon knows \( f \) and only his/her cost coefficients. As before we characterize \( n_H \) and \( n_S \).

To state the results we first define the function:

\[
\varphi(x) = \int_0^x t f(t) dt = x F(x) - G(x) \tag{4.10}
\]

where \( G(x) = \int_0^x F(t) dt \).

**Proposition 4.6.1. (Convexity of \( C \), characterization of \( n_H \))**

1. The hospital’s cost function \( C \) is (strictly) convex when \( o_H + u_H > 0 \).

2. The optimal solution, \( n_H \), to the optimization problem \( \min \{C(n) : n \geq 0\} \) is the unique solution of \( n \) to the following equation:

\[
\varphi\left(\frac{d}{n_H}\right) = \frac{o_H \mu}{o_H + u_H} \tag{4.11}
\]

**Proposition 4.6.2. (Concavity of \( \pi \), characterization of \( n_S \))**

1. The surgeon’s profit function \( \pi \) is strictly concave when \( o_S + u_S > 0 \).
2. The optimal solution, \( n_S \), to the optimization problem \( \max \{ \pi(n) : n \geq 0 \} \) is the unique solution of \( n \) to the following equation:

\[
\varphi \left( \frac{d}{n} \right) = \frac{os\mu - r}{os + u_S}
\]  

(4.12)

We next obtain a result similar to Theorem 4.4.5.

**Theorem 4.6.3.** The optimal number of surgeries from the hospital’s perspective \( n_H \) is less or equal than the surgeon’s preferred number of surgeries \( n_S \); i.e.,

\[
n_H \leq n_S
\]

if and only if

\[
\frac{o_H}{o_H + u_H} \geq \frac{os - \frac{r}{\mu}}{os + u_S}
\]  

(4.13)

A remark is in order here. The conditions of Theorems 4.4.5 and 4.6.3 are the same. To see this reorganize the terms in Eq(4.13) and rewrite them as given in Eq(4.5):

\[
\frac{o_H}{o_H + u_H} \geq \frac{os - \frac{r}{\mu}}{os + u_S} \quad \text{(by equation 4.13)}
\]

\[
\frac{o_Hu}{o_H + u_H} \geq \frac{os\mu - r}{os + u_S} \quad \text{multiply both sides with } \mu
\]

\[
\mu + \frac{r - os\mu}{os + u_S} \geq \frac{o_Hu}{o_H + u_H} \quad \text{(add } \mu \text{ to both sides and switch the terms)}
\]

\[
\frac{r + us\mu}{os + u_S} \geq \frac{u_Hu}{o_H + u_H} \quad \text{(condition given in Theorem 4.4.5)}
\]

We may rewrite inequality Eq(4.13) as

\[
\frac{1}{o_H} \frac{o_H}{o_H + u_H} \geq \frac{os - \frac{r}{\mu}}{os + u_S} \frac{1}{os}, \quad \text{i.e.,} \quad \frac{1}{1 + \frac{u_H}{os}} \frac{1}{\frac{1}{os}} \geq \frac{1 - \frac{r}{os}}{1 + \frac{u_S}{os}}
\]

Note that \( \mu os \) is the average cost of doing a surgery after time \( d \) for the surgeon, and \( r \) is the revenue from a surgery. The ratio \( \frac{r}{\mu os} \) is clearly strictly positive. Furthermore this ratio should be less than one because otherwise, i.e., \( \frac{r}{\mu os} > 1 \), it implies that surgeon would never stop doing surgeries.

On the other hand, we argue that the ratios \( \frac{u_H}{o_H} \) and \( \frac{u_S}{os} \) may be comparable even though the individual cost coefficients may not, i.e., \( u_H \) vs. \( u_S \) and \( o_H \) vs. \( os \). If they are comparable then as \( \frac{r}{\mu os} \) approaches 1 it becomes more attractive for the surgeon to work overtime suggesting \( n_H \leq n_S \).
4.6.2 Risk-Averse Surgeon

We now revisit our assumption that the surgeon is risk neutral and extend some of our findings to the risk-averse setting. When the surgeon is risk neutral his/her expected profits for undertaking \( n \) surgeries is:

\[
\pi(n) = rn - E_t[os \max\{0, nt - d\} + us \max\{0, d - nt\}].
\]

When we consider risk aversion, the expected profit function will include a (strict) concave increasing utility function \( v \), which is a one variable real-valued function mapping dollar amount to utility. The expected utility of undertaking \( n \) surgeries is assumed to be of the form

\[
E_t[v(\pi(n, t))]
\]

where

\[
\pi(n, t) = \begin{cases} 
    rn - us(d - nt) & \text{if } nt \leq d \\
    rn - os(nt - d) & \text{if } nt > d 
\end{cases}
\]

is the dollar amount of profit for \( n \) surgeries with time realization \( t \).

A remark on how the utility function is constructed is in order. The utility of undertaking \( n \) surgeries when time is realized as \( t \) is \( v(\pi(n, t)) \). In other words, utility is a function of the dollar amount of profit \( \pi(n, t) \). Note, however, that in defining \( \pi(n, t) \) we already made a conversion of opportunity cost of time into dollars when we defined the coefficients \( us \) and \( os \). The fact that the surgeon now has a general increasing concave utility function \( v \) does not change this determination of the coefficients \( us \) and \( os \). That is, we maintain, even under risk aversion, the fact that each time unit is worth \( us \) dollars before time \( d \) and \( os \) dollars after. In other words, the dollar value of time is a piecewise linear function under both risk neutrality and risk aversion.

The fundamental question we consider in this setting is how would a risk-averse surgeon choose his/her optimal number of surgeries, and would this value, for instance, be less than a risk neutral surgeon with the same profit function \( \pi(n, t) \). To make things precise we let \( n_R \) denote the optimal number of surgeries a risk averse surgeon would plan for a time period \( d \). That is,

\[
n_R \in \arg \max_n E_t[v(\pi(n, t))]
\]

and our question of interest is whether \( n_R \leq n_S \) where \( n_S \) is as defined in Proposition 4.6.2.
Finding \( n_R \) directly might be quite challenging depending on the structure of the utility function \( v \), so this bound in itself, if true, can be quite illuminating.

Another reason for our interest in the question “is \( n_R \leq n_S \)?” is due to its possible implication for compensation \( B \). If it turns out that \( n_R \) is in fact smaller than \( n_S \) then this may indicate that a risk-averse surgeon needs a smaller compensation \( B \) than that of risk neutral surgeon with identical costs in order to align his/her incentives with the hospital. The intuition is simple: the less a surgeon schedules surgeries on his/her own volition the less it would take to compensate him/her to perform \( n_H \) surgeries. Despite this intuition we were unable to prove the result analytically and plan to take it up in future research. We feel, nonetheless, that the question of whether \( n_R \leq n_S \) has independent interest, and thus pursue it further here.

We will establish some sufficient conditions for which \( n_R \leq n_S \). First we give an intuitive discussion for why this indeed might be the case, however, this intuitive line of reasoning yields only motivation and not concrete proof. Next we establish the result when the time distribution takes on only two values – \( t_L \) and \( t_H \) – where \( t_L \) is the time for a “routine” surgery and \( t_H \) for a surgery with “complications”. This simplification of the surgery duration distribution yields attractive conditions under which \( n_R \leq n_S \) holds. Finally, we present a sufficient condition on the cost coefficients for \( n_R \leq n_S \) for a general case, i.e., for a general surgery duration distribution.

4.6.3 Intuitive Discussion

The basis of our intuition for \( n_R \leq n_S \) comes from the following oversimplified notion of risk aversion: given the same expected monetary return from two lotteries, the lottery with smaller variance would be favored by a risk-averse decision maker. In particular, if one lottery has a smaller expected return and a greater variance than another then intuitively the latter is preferred by both a risk-averse and risk-neutral decision maker. Of course, in general, the validity of this statement depends on more than just the first two moments of the lottery distribution (and often on first or second order stochastic dominance) and possibly the degree of concavity of the the utility function [17], but the intuition is nonetheless clear.

As a brief aside, we should point out the well known result that if the utility function exhibits constant absolute risk aversion (CARA), i.e., (negative) exponential utility function, and the payoff of a lottery is normally distributed then maximizing the expected
utility function is the same as maximizing a term involving the mean and the variance (and some other given/known parameters), i.e., it is determined entirely by the first two moments of the distribution of the payoff distribution (called mean-variance utility). This result, however, does not apply to our setting. Indeed, even if we assume the utility function \( v \) has CARA it is clear that the payoffs are not normal even if we assume normality or lognormality of the surgery duration \( t \).

Turning to the problem at hand, the first fact to note is that, by definition, \( n_S \) maximizes the expected profit function \( \pi(n) \). In particular, this means that

\[
\pi(n_R) \leq \pi(n_S)
\]

for any choice of \( n_R \). The next result establishes that if \( n_R > n_S \) then the variance of expected payoffs under \( n_R \) surgeries is greater than for \( n_S \) surgeries.

**Lemma 4.6.4.** The variance of \( \pi(n, t) \), \( \text{Var} \, \pi(n, t) \), is proportional to \( n^2 \).

From this our intuition strongly suggests that \( n_R \leq n_S \), since by performing more than \( n_S \) surgeries one’s expected profit would be lower and variance of payoffs higher (i.e., there is more risk) and thus less attractive to a risk-averse surgeon.

### 4.6.4 Discrete Time Distribution with Two Values

In order to verify analytically the intuitive discussion in the previous subsection we begin by assuming a simple setting where the time distribution has two outcomes, \( t_L \) and \( t_H \), with probabilities \( p_L \) and \( p_H \) respectively. We will see how our intuition leads us to define sufficient conditions for \( n_R \leq n_S \).

By only using \( t_L \) and \( t_H \) values for a surgery duration we are making a rough approximation to the real distribution by picking only two values. We think of the low time \( t_L \) as the duration of routine/easy surgery where there are no complications. On the other hand, \( t_H \) corresponds to a high duration associated with the occurrence of complications during surgery.

We now present some simple conditions on cost coefficients for when \( n_R \leq n_S \). The argument essentially follows our intuition presented in the previous subsection. We begin by assuming that \( n_R > n_S \) and derive conditions whereby this cannot occur. The first step is to show that there exists a number of surgeries \( m \) with the same expected profit as \( n_R \) where \( m \leq n_S \).
Lemma 4.6.5. Let $n_R > n_S$. Then there exists an $m$ with $0 \leq m \leq n_S$ such that $\pi(m) = \pi(n_R)$.

Figure 4.7 provides an illustration of this result. The next step is to demonstrate that the expected utility of performing $m$ surgeries is greater than that of performing $n_R$ surgeries, thus contradicting the definition of $n_R$ (see Eq(4.14)). The following lemma illustrates that when some conditions on the expected profits at $t_L$ and $t_H$ hold, we can indeed establish the contradiction. These conditions imply that the expected utilities of the various outcomes associated with $n_R$ surgeries are “more spread out” than with $m$ surgeries.

Proposition 4.6.6. The expected utility from undertaking $m$ surgeries is greater than the expected utility of undertaking $n_R$ surgeries when the following condition holds:

$$\pi(n_R, t_H) \leq \pi(m, t_L) \leq \pi(m, t_H) \leq \pi(n_R, t_L)$$

(4.16)

The idea of the proof is illustrated in Figure 4.8. The choice of either $n_R$ or $m$ determines one of two lotteries for the surgeon. The first lottery, associated with the choice $n_R$, is as follows: earn profit $\pi(n_R, t_H)$ with probability $p_H$ and earn profit $\pi(n_R, t_L)$ with probability $p_L$. The expected profit of the first lottery is

$$\pi(n_R) = p_H \pi(n_R, t_H) + p_L \pi(n_R, t_L).$$

Similarly the second lottery has outcome $\pi(m, t_L)$ with probability $p_L$ and outcome $\pi(m, t_H)$ with probability $p_H$. Both lotteries have the same expected value of $\pi(n_R) = \pi(m)$. Then,
it can be seen graphically that $E_t[v(n_R, t)] < E_t[v(m, t)]$. Details of the proof can be found in Section 4.8.

A few comments on the previous theorem are in order. First, we give a brief interpretation of inequality Eq(4.16). When surgery duration is short (i.e., $t_L$ is realized) it is intuitive that we would like to do more surgeries, thus motivating the condition $\pi(m, t_L) \leq \pi(n_R, t_L)$. When surgeries are long (i.e., $t_H$) the opposite holds true: the surgeon would favor fewer surgeries, thus motivating the condition $\pi(n_R, t_H) \leq \pi(m, t_H)$. The remaining conditions implied by inequality Eq(4.16) are less straight-forward to motivate. Figure 4.9 gives an illustration of some time values $t_L$ and $t_H$ which satisfy these conditions. The two functions shown in the figure are the profit functions $\pi(m, t)$ and $\pi(n_R, t)$. Since $n_R > m$ by assumption, it follows that $\pi(n_R, t)$ peaks (at $t = d/n_R$) to the left of $\pi(m, t)$ (at $t = d/m$).

From this figure we can also see why some conditions similar to inequality Eq(4.16) are not valid to give our desired result. Indeed, a similar picture to Figure 4.8 could be drawn with conditions

$$\pi(n_R, t_L) \leq \pi(m, t_L) \leq \pi(m, t_H) \leq \pi(n_R, t_H)$$

and it might appear that a similar result would hold. However, it can be readily seen from Figure 4.9 that condition Eq(4.17) does not hold under the natural restriction that $t_L \leq t_H$. 

Figure 4.8: Illustration of $n_R \leq n_S$ with Two $t$ Values
4.6.5 Sufficient Conditions on Cost Coefficients

In the previous subsection we made assumptions on the time distribution in order to find conditions whereby $n_R \leq n_S$. Now we relax these restrictions and allow for a general continuous distribution of time, only restricting the relative sizes of the cost coefficients. We derive the following result.

**Proposition 4.6.7.** If $rn_S > d(o_S + u_S)$ then $n_R \leq n_S$. Thus, in particular if $r > d(o_S + u_S)$ then $n_R \leq n_S$.

This result implies that if the total revenue from performing $n_S$ surgeries is large enough (i.e., greater than $d(o_S + u_S)$) then we will find that $n_R \leq n_S$.

4.7 Conclusion and Future Directions

We have presented a model and analysis for the problem of determining the number of surgeries to schedule in an OR block of fixed length that takes into consideration the competing incentives of hospital and surgeon. By proposing contracts that induce the surgeon to schedule a number of surgeries more aligned with the goals of the hospital, the hope is that this alignment of incentives leads to a reduction in costs (especially the overtime costs) and an overall improvement in the working environment. Savings garnered in this
scheme could be used to open up other OR’s or intensive-care beds and further improve OR throughput.

Depending on how much power the hospital has over surgeons and how much information is available to the hospital, we propose several implementable contracts that hospital might consider in Section 4.5. In this section, we also provide a discussion of the problem in a perspective of a social planner, e.g., a provincial government.

Our analysis is based on some important assumptions. We argue that many of them are quite reasonable and made for tractability to remove complexity in the analysis and demonstrate as simply as possible the incentives involved. Two of the stronger assumptions in our basic setting are risk-neutrality of the surgeon and the fact that all surgeries are identical and independently distributed. In Section 4.6 we provide a framework that relaxes these two assumptions, but introduce another set of restrictive conditions. Although both models are restrictive, the fact that they are different in character but yield similar insights testifies to the robustness of our approach.

We now point out some other possible extensions of our model that may be promising directions for future research. Firstly, we have modeled the interaction between the hospital and surgeon as a single period game. However, the hospital and surgeon have a long-term working relationship and it may add more insight to explore this setting in a repeated game structure. One direction is that the bonus structure may be used as a “carrot” by the hospital to induce cooperation from the surgeon at each stage in a repeated game. Secondly, there is scope to examine more closely how the size of the bonus would change with degrees of risk aversion. Our results on the magnitude of \( n_R \) with respect to \( n_S \) could be a foundation for this study. Thirdly, one can study the asymmetric information case in more detail, in which both surgeon’s and hospital’s cost coefficients are private and not known by the other party.

4.8 Proofs

**Proof of Proposition 4.4.3.** It suffices to show that \( \theta(n) \) is discrete convex. Note that

\[
\pi(n) = (r + u_s \mu)n - (o_S + u_S)\theta(n) - u_S d
\]

is discrete convex when each of the the first two terms are discrete convex. The linear term \((r + u_s \mu)n\) is both discrete convex and concave, and since \( o_S + u_S \geq 0 \) by assumption the second term is discrete concave precisely
when $\theta(n)$ is discrete convex. Thus, it suffices to show that $\theta(n)$ is discrete convex; that is, $\Delta^2 \theta(n) \geq 0$.

To establish this we consider various cases for durations $T(n)$, $T(n+1)$, etc., with respect to the normal day duration $d$. There are three important ranges for $d$: $0 \leq d < T(n)$; $T(n) \leq d < T(n+1)$; $T(n+1) \leq d < T(n+2)$; and $d \geq T(n+2)$. We will compute the expectation in the definition of $\theta(n)$ with these various ranges. The following table contains the necessary information:

<table>
<thead>
<tr>
<th>Range for $d$</th>
<th>$[0, T(n))$</th>
<th>$[T(n), T(n+1))$</th>
<th>$[T(n+1), T(n+2))$</th>
<th>$[T(n+2), \infty)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta O(n)$</td>
<td>$t_{n+1}$</td>
<td>$T(n+1) - d$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\Delta O(n+1)$</td>
<td>$t_{n+2}$</td>
<td>$t_{n+2}$</td>
<td>$T(n+2) - d$</td>
<td>0</td>
</tr>
<tr>
<td>$\Delta^2 O(n)$</td>
<td>$t_{n+2} - t_{n+1}$</td>
<td>$t_{n+2} - (T(n+1) - d)$</td>
<td>$T(n+2) - d$</td>
<td>0</td>
</tr>
</tbody>
</table>

Thus, using conditional expectation we write

$$\Delta^2 \theta(n) = \mathbb{E}[\Delta^2 O(n)|d < T(n)] \Pr(d < T(n))$$

$$+ \mathbb{E}[\Delta^2 O(n)|T(n) \leq d < T(n+1)] \Pr(T(n) \leq d < T(n+1))$$

$$+ \mathbb{E}[\Delta^2 O(n)|T(n+1) \leq d < T(n+2)] \Pr(T(n+1) \leq d < T(n+2))$$

$$+ \mathbb{E}[\Delta^2 O(n)|d \geq T(n+2)] \Pr(d \geq T(n+2))$$

$$\geq \underbrace{\mathbb{E}[t_{n+2} - t_{n+1}|d < T(n)] \Pr(d < T(n))}_{(a)}$$

$$+ \underbrace{\mathbb{E}[t_{n+2} - (T(n+1) - d)|T(n) \leq d < T(n+1)] \Pr(T(n) \leq d < T(n+1))}_{(b)}$$

The first term $(a)$ simplifies as (using linearity of expectations and independence):

$$(a) = (\mathbb{E}[t_{n+2}|d < T(n)] - \mathbb{E}[t_{n+1}|d < T(n)]) \Pr(d < T(n))$$

$$= (\mathbb{E}[t_{n+2}] - \mathbb{E}[t_{n+1}]) \Pr(d < T(n))$$

$$= (\mu - \mu) \Pr(d < T(n)) = 0$$

The second term $(b)$ can be written as:

$$(b) = (\mathbb{E}[t_{n+2}] - \mathbb{E}[t_{n+1} - (d - T(n))|T(n) \leq d, t_{n+1} > d - T(n)]) \Pr(T(n) \leq d < T(n+1))$$

$$= (\mu - \mathbb{E}[t_{n+1} - k|t_{n+1} \geq k, k \geq 0]) \Pr(T(n) \leq d < T(n+1))$$

$$\geq 0$$
where \( k = d - T(n) > 0 \). The last inequality holds that since by NBUE we have \( \mathbb{E}[t_i - k|t_i \geq k] \leq \mathbb{E}[t_i] \). Thus, \( \Delta^2 \theta(n) \geq 0 \).

**Proof of Corollary 4.4.4.** The proof follows from Proposition 4.4.3 and is thus omitted.

**Proof of Theorem 4.4.5.** Since \( \theta(n) \) is discrete convex (shown above), this implies \( \Delta \theta(n) - \Delta \theta(n') \geq 0 \) for \( n \geq n' \). Thus, \( \Delta \theta(n_S) - \Delta \theta(n_H - 1) \geq \frac{t + u_H \mu}{o_S + u_S} - \frac{u_H \mu}{o_H + u_H} > 0 \)

Now, if \( n_S \leq n_H - 1 \) then \( \Delta \theta(n_S) - \Delta \theta(n_H - 1) \leq 0 \) thus we can conclude that \( n_S \geq n_H \).

**Proof of Proposition 4.6.1.** We first establish \((i)\) by considering second order conditions. The first and second derivative of \( C \) (given in Eq(4.8)) with respect to \( n \) are respectively:

\[
C'(n) = o_H \int_{d/n}^{\infty} tf(t)dt - u_H \int_{0}^{d/n} tf(t)dt
\]  \hspace{1cm} \text{(4.18)}

and

\[
C''(n) = \frac{d^2}{n^3} f \left( \frac{d}{n} \right) (u_H + o_H)
\]  \hspace{1cm} \text{(4.19)}

using judicious appeals to Leibniz’s rule for differentiating integrals and some basic housekeeping. Observe \( C''(n) > 0 \) for all \( n \geq 0 \) (and hence \( C \) is (strictly) convex) provided the sum of the cost coefficients \( o_H + u_H \) is positive.

As for \((ii)\), since \( C \) is convex a sufficient condition for optimality is \( C'(n) = 0 \). Thus the equation \( C'(n) = 0 \) characterizes \( n_H \). From Eq(4.18) this yields the equivalent relation:

\[
o_H \int_{d/n}^{\infty} tf(t)dt = u_H \int_{0}^{d/n} tf(t)dt.
\]  \hspace{1cm} \text{(4.20)}

Noting the fact

\[
\mu = \int_{0}^{\infty} tf(t)dt = \int_{0}^{d/n} tf(t)dt + \int_{d/n}^{\infty} tf(t)dt,
\]

we remove the improper integral on the left-hand side of Eq(4.20) by substitution and simplify Eq(4.20) to:

\[
\frac{o_H}{u_H} = \frac{\int_{0}^{d/n} tf(t)dt}{\mu - \int_{0}^{d/n} tf(t)dt}.
\]  \hspace{1cm} \text{(4.21)}
We further simplify this expression further by expanding the integral $\int_0^{d/n} tf(t)dt$ using integration by parts yielding:

$$\int_0^{d/n} tf(t)dt = \frac{d}{n} F\left(\frac{d}{n}\right) - \int_0^{d/n} F(t)dt.$$

Using our notation $G(y) = \int_0^y F(t)dt$ and $\varphi(y) = yF(y) - G(y)$ we yield an equivalent expression for Eq(4.21) as follows:

$$\frac{o_H}{u_H} = \frac{\varphi\left(\frac{d}{n}\right)}{\mu - \varphi\left(\frac{d}{n}\right)}.$$

A simple rearrangement yields the characterization of $n_H$:

$$\varphi\left(\frac{d}{n_H}\right) = \frac{o_H \mu}{o_H + u_H},$$

(4.22)

thus establishing (ii).

Proof of Proposition 4.6.2. The details are similar to the proof of Proposition 4.6.1 so details here are more brief. We establish (i) by considering second order conditions. The first and second derivative of $\pi$ (given in Eq(4.9)) with respect to $n$ are respectively:

$$\pi'(n) = r - o_S \int_0^{\infty} tf(t)dt + u_S \int_0^{d/n} tf(t)dt$$

(4.23)

and

$$\pi''(n) = -\frac{d^2}{n^3} f\left(\frac{d}{n}\right) (u_S + o_S).$$

(4.24)

Observe $\pi''(n) < 0$ for all $n \geq 0$ (and hence $\pi$ is (strictly) concave) provided the sum of the cost coefficients $o_S + u_S$ is positive.

As for (ii), since $\pi$ is concave an optimal solution is achieved at $\pi'(n) = 0$. Thus the equation $\pi'(n) = 0$ characterizes $n_S$. Using similar manipulations as above this yields an equivalent characterization of $n_S$ given by:

$$\varphi\left(\frac{d}{n_S}\right) = \frac{o_S \mu - r}{o_S + u_S},$$

(4.25)

thus establishing (ii).

The following lemma is useful in the proofs of the next two theorems. It reveals a useful property of the function $\varphi$ which plays a common role in the characterizing $n_H$ and $n_S$.

Lemma 4.8.1. The function $\varphi(x) = xF(x) - G(x)$ is an increasing function of $x$ for $x \geq 0$. 

129
Proof. We show that $\varphi'(x) \geq 0$ for $x \geq 0$. Note that:

$$
\varphi'(x) = F(x) + xf(x) - G'(x)
$$

$$
= F(x) + xf(x) - F(x)
$$

$$
= xf(x) \geq 0
$$

where the inequality holds since the pdf $f$ is non-negative and $x \geq 0$. □

Proof of Theorem 4.6.3. Note the characterizations of $n_H$ and $n_S$ given in Eq(4.11) and Eq(4.12) reproduced here for convenience:

$$
\varphi \left( \frac{d}{n_H} \right) = \frac{o_H \mu}{o_H + u_H}
$$

and

$$
\varphi \left( \frac{d}{n_S} \right) = \frac{o_S \mu - r}{o_S + u_S}.
$$

Since $\varphi$ is an increasing function then $\varphi \left( \frac{d}{n} \right)$ is a decreasing function of $n$. Thus to show $n_H \leq n_S$ it is equivalent to show $\varphi \left( \frac{d}{n_H} \right) \geq \varphi \left( \frac{d}{n_S} \right)$. By the above characterization it is in turn equivalent to establish:

$$
\frac{o_H \mu}{o_H + u_H} \geq \frac{o_S \mu - r}{o_S + u_S}.
$$

We obtain the desired result by dividing both sides by $\mu$. □

Proof of Lemma 4.6.4. Recall that $\pi(n,t) = rn - o_S \max\{0, nt - d\} - u_S \max\{0, d - nt\}$. Then $\text{Var} \pi(n,t) = \text{Var} \left( o_S \max\{0, nt - d\} + u_S \max\{0, d - nt\} \right)$. Since we are interested in finding only how $n$ is related to the variance of $\pi(n,t)$ w.l.o.g. we may assume $o_S = u_S = 1$. Also note that $\max\{0, nt - d\} + \max\{0, d - nt\} = |nt - d|$. Therefore we just need to find $\text{Var} |nt - d|$. But since $d$ is a constant we obtain $\text{Var} |nt - d| = \text{Var} |nt| = n^2 \text{Var} |t|$, and since $t > 0$ we obtain $n^2 \text{Var} |t| = n^2 \text{Var} t$. Hence $\text{Var} \pi(n,t)$ is proportional to $n^2$. □

Proof of Lemma 4.6.5. The expected profit function $\pi$ (see Eq(4.9)) is continuous. We know $\pi(0) = -u_S d$ and by inequality Eq(4.15) $\pi(n_R) < \pi(n_S)$ and so by the Intermediate Value Theorem there exists an $m$ with $0 \leq m \leq n_S$ such that $\pi(m) = \pi(n_R)$. □

Proof of Proposition 4.6.6. Condition Eq(4.16) and the fact $v$ is increasing imply

$$
v(\pi(n_R, t_H)) \leq v(\pi(m, t_L)) \leq v(\pi(m, t_H)) \leq v(\pi(n_R, t_L))
$$
as illustrated in Figure 4.8. We know that the point \((\pi(n_R), E_t[v(\pi(n_R, t))])\) lies on the line segment between points \((\pi(n_R, t_H), v(\pi(n_R, t_H)))\) and \((\pi(n_R, t_L), v(\pi(n_R, t_L)))\) and is in fact the convex combination of the points given by:

\[
p_H(\pi(n_R, t_H), v(\pi(n_R, t_H))) + p_L(\pi(n_R, t_L), v(\pi(n_R, t_L))).
\]

By the concavity of \(v\) this line segment lies below the line segment adjoining points \((\pi(m, t_L), v(\pi(m, t_L)))\) and \((\pi(m, t_H), v(\pi(m, t_H)))\) for every profit level on which both line segments are defined, and in particular for expected profit level \(\pi(n_R) = \pi(m)\). It then follows that \(E_t[v(\pi(n_R, t))] \leq E_t[v(\pi(m, t))]\), which can be seen graphically in Figure 4.8. \(\Box\)

**Proof of Proposition 4.6.7.** Recall our assumption that \(v\) is a strictly increasing, twice differentiable and concave utility function, i.e., \(v' > 0\) and \(v'' < 0\). Define \(\eta(n)\) as the expected utility function, i.e., \(\eta(n) = E_t[v(\pi(n, t))]\). Since \(\pi(n, t)\) and \(v\) are (strictly) concave and concavity is preserved by expectation operator \(\eta\) is (strictly) concave. Thus \(n_R \in \arg\max_{n \geq 0} \eta(n)\). We can compute \(n_R\) by finding \(\eta'(n)\) with Leibniz’s rule and solving for \(n\) in

\[
\eta'(n) = \int_0^{d/n} (r + uSt) v'(rn - uSd + uSnt) f(t) dt + \int_{d/n}^\infty (r - oSt) v'(rn + oSd - oSnt) f(t) dt = 0.
\]

We know that \(\pi'(n_S) = 0\) by Proposition 4.6.2. Our strategy is to show \(\eta'(n_S) \leq 0\) which implies \(n_R \leq n_S\). We will start from the expression \(\pi'(n_S)\) and obtain \(\eta'(n_S) \leq 0\).

Recall, by Eq(4.23), \(\pi'(n_S) = \int_0^{d/n_S} (r + uSt) f(t) dt + \int_{d/n_S}^\infty (r - oSt) f(t) dt = 0\).

Since \(r > d(oS + uS)\) and \(n_S \geq 1\), we have \(r/oS > d/n_S\) and can rewrite \(\pi'(n_S)\) so that Eq(4.23) is equivalent to

\[
\pi'(n_S) = \int_0^{d/n_S} (r + uSt) f(t) dt + \int_{d/n_S}^{r/oS} (r - oSt) f(t) dt + \int_{r/oS}^\infty (r - oSt) f(t) dt = 0. \quad (4.26)
\]

Now we multiply each term in Eq(4.26) by \(v'(oSd)\), a nonnegative constant, and rewrite it as

\[
\int_0^{d/n_S} v'(oSd)(r + uSt) f(t) dt + \int_{d/n_S}^{r/oS} v'(oSd)(r - oSt) f(t) dt + \int_{r/oS}^\infty v'(oSd)(r - oSt) f(t) dt = 0. \quad (4.27)
\]

Note that part 1 and part 2 are non-negative, and part 3 is non-positive. Next we will obtain inequalities individually for each one these three parts. First we recall that \(v\) is
strictly concave so we have $v'' < 0$, i.e., $v'$ is strictly decreasing. Also note that we assume $n_S r > d(o_S + u_S)$ and $n_S \geq 1$.

We start with part 1. Since $r > d(o_S + u_S)/n_S$ and $v'$ is decreasing we have $0 \leq v'(r n_S - u_S d + u_S n_st) \leq v'(o_S d)$ for $0 \leq t \leq d/n_S$. Therefore

$$
\int_0^{d/n_S} v'(o_S d)(r + u_S t)f(t)dt \geq \int_0^{d/n_S} v'(r n_S - u_S d + u_S n_st)(r + u_S t)f(t)dt \geq 0. \quad (4.28)
$$

Next we obtain a similar result for part 2. Since $r > d(o_S + u_S)/n_S$ and $v'$ is decreasing we have $0 \leq v'(r n_S + o_S d - o_S n_st) \leq v'(o_S d)$ for $d/n_S \leq t \leq r/o_S$. Hence

$$
\int_{d/n_S}^{r/o_S} v'(o_S d)(r - o_S t)f(t)dt \geq \int_{d/n_S}^{r/o_S} v'(r n_S + o_S d - o_S n_st)(r - o_S t)f(t)dt \geq 0. \quad (4.29)
$$

Finally, we obtain an inequality for part 3. Since $r > d(o_S + u_S)/n_S$ and $v'$ is decreasing we have $0 \leq v'(o_S d) \leq v'(r n_S + o_S d - o_S n_st)$ for $t \geq r/o_S$ we obtain

$$
0 \geq \int_{r/o_S}^{\infty} v'(o_S d)(r - o_S t)f(t)dt \geq \int_{r/o_S}^{\infty} v'(r n_S + o_S d - o_S n_st)(r - o_S t)f(t)dt. \quad (4.30)
$$

When we look at the Eq.(4.28), Eq.(4.29) and Eq.(4.30) we see that non-negative parts, i.e., part 1 and part 2 are getting less positive, and non-positive part, i.e., part 3 gets more negative when we replace $v'(o_S d)$ with the corresponding $v'(.')$’s. We put the Eq.(4.28), Eq.(4.29), Eq.(4.30), Eq.(4.26) and Eq.(4.27) together and obtain

$$
\pi'(n_S) = \begin{cases} 
0 & \\
= \int_0^{d/n_S} (r + u_S t)f(t)dt + \int_{d/n_S}^{r/o_S} (r - o_S t)f(t)dt + \int_{r/o_S}^{\infty} (r - o_S t)f(t)dt \\
\geq \int_0^{d/n_S} v'(r n_S - u_S d + u_S n_st)(r + u_S t)f(t)dt \\
+ \int_{d/n_S}^{r/o_S} v'(r n_S + o_S d - o_S n_st)(r - o_S t)f(t)dt \\
+ \int_{r/o_S}^{\infty} v'(r n_S + o_S d - o_S n_st)(r - o_S t)f(t)dt \\
= \eta'(n_S).
\end{cases}
$$

Hence $\eta'(n_S) \leq 0$ implying that $n_R \leq n_S$ as claimed. \hfill \Box
4.9 Bibliography


5 Advance Multi-Period Quantity Commitment and Appointment Scheduling

We introduce advance multi-period quantity (order or supply) commitment problems with stochastic characteristics (demand or yield) and several real-world applications. There are underage and overage costs if there is a mismatch between committed and realized quantities. Decisions are needed now and they are the order or supply amounts for the next \( n \) periods. The objective is to maximize the total expected profit of \( n \) periods. We establish a link between these advance multi-period quantity commitment problems and the appointment scheduling problem studied in Chapter 2. We show that these problems can be thought of and solved (efficiently) as special cases of the appointment scheduling problem.

5.1 Introduction

We introduce and study advance multi-period quantity (order or supply) commitment problems with random characteristics (demand or yield), explore their relationship with the appointment scheduling problem given in Chapter 2 and provide several real-world applications. All quantity decisions (how much to order or supply in each of the next \( n \) periods) are needed now, i.e., before any realization of demand or yield. We show that these problems can be modeled and solved as special cases of the appointment scheduling problem.

In a supply chain, uncertainty consequences (e.g., due to stochastic demand or random yield) are something that players would like to minimize and, when possible, pass to others. Consider a buyer and a supplier where the buyer can order any amount from the supplier.

---

\(^1\)A version of this chapter will be submitted for publication. Begen M.A. and Queyranne M. Advance Multi-Period Quantity Commitment and Appointment Scheduling.
whenever it is convenient. This may be the case where there are many suppliers and they are competing for buyers. However, when possible a supplier would prefer a contract in which the buyer (who has better information about the demand uncertainty) commits in advance how much to purchase over a certain period of time. In return, the supplier may offer a discount to the buyer to make this choice attractive. These type of agreements are reported in practice, e.g., [2], [8] and [7]. With such an agreement, the challenge for the buyer becomes to determine how much to commit to purchase in advance (e.g., in total for the entire horizon or per period) and how much to order in each period. This problem and its variants (such as finite or infinite horizons, with or without fixed costs, total or individual period commitments) have been well motivated and studied in literature e.g., [2], [8], [5], [4], [3] [7] and [1]. These studies mostly (and naturally) use dynamic programming to determine an optimal policy and in some cases they develop heuristics. Nevertheless, all the previous studies on this topic that we are aware of consider situations where a buyer commits on how much to purchase in advance and decides how much to order in each period consecutively, i.e., the ordering decision for the next period is given after this period’s demand realization.

In our setting, the buyer needs to decide how much to order for all periods at once and now, i.e., before any realization of random demands. There can be some situations where the buyer needs to enter such a contract to secure any orders from a strong supplier. Or alternatively, we think of a producer who is subject to random yield and needs to determine now how much to supply for each of the $n$ periods before any production levels are known. In this case, the producer may be subject to stiff competition and needs to promise customers supply quantities for each of the next $n$ periods before the production horizon starts. If there is any product shortage in a period then the producer will obtain the product by other means, e.g., purchase it from a competitor. Furthermore, the producer has a high inventory holding cost so building inventory in advance to compensate for product shortage may not be profitable but is necessary when there is excess inventory.

We first provide the details on the advance multi-period quantity (order or supply) commitment problems and then show that they have a very strong connection with the appointment scheduling problem given in Chapter 2. Then we use the algorithmic and convex optimization results obtained in Chapter 2 and Chapter 3 (for the appointment scheduling problem) to determine optimal levels of quantity commitments (order or supply) before the first period. To our best knowledge, the problems considered in this chapter have
not been yet studied.

We use the same assumptions and notation as in Chapter 2 and Chapter 3. We provide a description of appointment scheduling problem as well as introduce notation in Section 5.2. The rest of the chapter is organized as follows. In Section 5.3, we introduce a multi-period inventory model for a perishable product with advance order commitments, provide a few real-world examples and show that it has one to one correspondence with the appointment scheduling problem. Therefore, it may be solved efficiently as a special case of the appointment scheduling problem. We will refer to this model as “the inventory model”. Section 5.4 introduces the model for the production problem with advance supply commitments and random yield. In this section, we establish a link between “the production problem” and the appointment scheduling problem. We show that (under a mild condition on cost coefficients) the objective function of this problem is \(L\)-concave\(^2\) (if production quantities are integer) and concave (if production quantities are real). Furthermore, if the yield distributions are independent then the production problem can also be solved efficiently as in the case of appointment scheduling. Finally, we conclude the chapter in Section 5.5.

### 5.2 Description of Appointment Scheduling Problem

This section closely follows from Chapter 2 and Chapter 3. There are \(n+1\) jobs numbered 1, 2, ..., \(n+1\) that need to be sequentially processed (in the order of 1, 2, ..., \(n+1\)) on a single processor. An appointment schedule, i.e., a processing duration allocation \(a_i\) for each job \(i\), is needed before any processing can start. That is, each job is assigned a planned start date, i.e., appointment date \(A_i\) where \(A_1 = 0\) and \(A_i = A_{i-1} + a_{i-1}\) for \(i = 2, 3, ... n+1\). The processing durations are stochastic and we are only given their joint discrete distribution. When a job finishes later than the next job’s appointment date, the system experiences overage cost due to the overtime of the current job and the waiting of the next job. On the other hand, if a job finishes earlier than the next job’s appointment date, the system experiences some cost due to under-utilization, i.e., underage cost. The goal is to find appointment dates, \((A_1, ..., A_n)\), that minimize the total expected cost.

There are \(n\) real jobs. The \((n+1)\)th job is a dummy job with a processing duration of 0. The appointment time for the \((n+1)\)th job is the total time available for the \(n\) real jobs. We

\(^2\)See Definition 5.4.1.
use the dummy job to compute the overage or underage cost of the $n^{th}$ job. We denote the random processing duration of job $i$ by $p_i$ and the random vector$^3$ of processing durations by $\mathbf{p} = (p_1, p_2, ..., p_n, 0)$. Let $\overline{p}_i$ denote the maximum possible value of processing duration $p_i$, respectively. The maximum of these $\overline{p}_i$’s is $\overline{p}_{\text{max}} = \max(\overline{p}_1, ..., \overline{p}_n)$. The \textit{underage cost rate} $u_i$ of job $i$ is the cost (per unit time) incurred when job $i$ is completed at a date $C_i$ before the appointment date $A_i+1$ of the next job $i + 1$. The \textit{overage cost rate} $o_i$ of job $i$ is the unit cost incurred when job $i$ is completed at a date $C_i$ after the appointment date $A_i+1$.

Thus the total cost due to job $i$ completing at date $C_i$ is $u_i(A_i+1 - C_i)^+ + o_i(C_i - A_i+1)^+$ where $(x)^+ = \max(0, x)$ is the positive part of real number $x$. We define $\mathbf{u} = (u_1, u_2, ..., u_n)$ and $\mathbf{o} = (o_1, o_2, ..., o_n)$. We assume, naturally, that all cost coefficients and processing durations are non-negative and bounded. We also assume that processing durations are integer valued.$^4$

Next we introduce our decision variable for the appointment scheduling problem. Let $\mathbf{A} = (A_1, A_2, ..., A_n, A_{n+1})$ (with $A_1 = 0$) be the \textit{appointment vector} where $A_i$ is the appointment date for job $i$. We introduce additional variables which help define and express the objective function. Let $S_i$ be the start date and $C_i$ the completion date of job $i$. Since job 1 starts on-time we have $S_1 = 0$ and $C_1 = p_1$. The other start times and completion times are determined as follows: $S_i = \max\{A_i, C_{i-1}\}$ and $C_i = S_i + p_i$ for $2 \leq i \leq n + 1$. Note that the dates $S_i$ and $C_i$ are random variables which depend on the appointment vector $\mathbf{A}$, and the random duration vector $\mathbf{p}$.

Let $F(\mathbf{A}|\mathbf{p})$ be the total cost of appointment vector $\mathbf{A}$ given processing duration vector $\mathbf{p}$:

$$F(\mathbf{A}|\mathbf{p}) = \sum_{i=1}^{n} (o_i(C_i - A_{i+1})^+ + u_i(A_{i+1} - C_i)^+).$$

The objective to be minimized is the expected total cost $F(\mathbf{A}) = E_\mathbf{p}[F(\mathbf{A}|\mathbf{p})]$ where the expectation is taken with respect to random processing duration vector $\mathbf{p}$.

Our framework can include a given due date $D$ for the end of processing (e.g., end of day for an operating room for the appointment scheduling problem, or a quota set by the supplier for the inventory model) after which overtime is incurred, instead of letting the model choose a planned makespan $A_{n+1}$. We assume $D$ is an integer and that $0 \leq D \leq \sum_{i=1}^{n} \overline{p}_i$. Define

$^3$We write all vectors as row vectors.

$^4$We can restrict ourselves to integer appointment schedules without loss of optimality by Appointment Vector Integrality Theorem 2.5.10 of Chapter 2.
\( \tilde{A} = (A_1, A_2, ..., A_n) \) then the new objective becomes

\[
F^D(\tilde{A}) = \mathbb{E}_p \left[ \sum_{j=1}^{n-1} \left( o_j (C_j - A_{j+1})^+ + u_j (A_{j+1} - C_j)^+ \right) + o_n (C_n - D)^+ + u_n (D - C_n)^+ \right].
\]

We immediately observe that \( F(\tilde{A}, D) = F^D(\tilde{A}) \). We end this section with two definitions that we need later in the chapter. First definition is a mild condition on cost coefficients and is due to Definition 2.6.5 of Chapter 2.

**Definition 5.2.1.** The cost coefficients \((u, o)\) are \(\alpha\)-monotone if there exists reals \(\alpha_i \) (1 \(\leq i \leq n\)) such that 0 \(\leq \alpha_i \leq o_i\) and \(u_i + \alpha_i\) are non-increasing in \(i\), i.e., \(u_i + \alpha_i \geq u_{i+1} + \alpha_{i+1}\) for all \(i = 1, \ldots, n - 1\).

The condition of \(\alpha\)-monotonicity is automatically satisfied if all underage cost coefficients are the identical, i.e., \(u_i = u\) for all \(i\).

Let \(Z\) denote the set of integers and \(1\) is a vector in \(\mathbb{R}^{n+1}\) where each component is 1. The next definition gives the definition of a \(L\)-convex function.

**Definition 5.2.2.** \(f : Z^q \rightarrow \mathbb{R} \cup \{\infty\}\) is \(L\)-convex iff \(f(z) + f(y) \geq f(z \lor y) + f(z \land y) \forall z, \forall y \in Z^q\) and \(\exists r \in \mathbb{R} : f(z + 1) = f(z) + r \forall z \in Z^q\) [9].

### 5.3 A Multi-Period Inventory Model for a Perishable Product with Advance Commitments

Consider a buyer who has to make ordering decisions for the next \(n\) periods at time zero for a perishable product with a stochastic demand. Since the product is perishable, excess (unsold) items at the end of a period cannot be used in the next period and they need to be disposed. On the other hand, unsatisfied demand is backordered. Furthermore, there may be a quota for the total purchases (orders and backorders) such that it is more costly to order beyond this quota. The objective of the buyer is to determine how much to commit now for the next \(n\) periods to maximize his/her expected profits. We think of the profits as revenue—costs where revenue is simply number of products sold times the contribution factor (including unit product cost). In our setting number of products sold is equal to the total demand since we assume backorders and satisfy all the demand. On the other hand, the costs consists of holding (and/or disposal costs) and backorder costs. Revenue will be
a constant after taking the expectation with respect to demand of each period therefore we can think of this problem profit maximization problem as a cost minimization, i.e., minimization of total expected holding (and/or disposal) and backorder costs.

There are many real-world examples for this setting. For example, consider a retailer who is under heavy competition and has a single supplier. Demand for the retailer’s product is stochastic, and the supplier requires the retailer to commit its orders for the next $n$ periods and may set a quota for the total amount of purchases. The retailer has little negotiating power with the supplier, and wants to keep his reputation by satisfying all the demand using backordering when needed. In this setup, we look at the problem in the retailer’s eyes and have to decide on orders for the next $n$ periods.

This inventory model and the appointment scheduling model as defined in Chapter 2 has one to one correspondence in terms of their structure, data, decision variables and objective function. In the inventory model we have $n$ periods whereas in the appointment scheduling model there are $n$ jobs. The random component, $p_i$ in the inventory model is the demand for period $i$ whereas in the appointment scheduling model it is the processing duration of job $i$. The costs are $u$ and $o$; in the appointment scheduling model $u$ is the underage (earliness) and $o$ is the overage (waiting and/or overtime) cost whereas in the inventory model $u$ is the holding (and/or disposal) and $o$ is the backorder cost. The decision variable in the appointment scheduling model is the appointment date $A_i$ of job $i$, or how much time to allocate $a_i$ for job $i$. On the other hand, the decision for the inventory model is how much to order for period $i$, $a_i$, and as given in Section 5.2 we have the relationship $a_i = A_{i+1} - A_i$.

The start time of job $i$, $S_i$, in the appointment scheduling model corresponds to total purchase (orders and backorders) up to period $i$ (not including period $i$), and completion time of job $i$, $C_i$ is the total purchase up to period $i + 1$ (including period $i$). Table 5.1 gives a comparison summary between the appointment scheduling model and the described inventory problem in terms of data and decision variables.

In Figure 5.1 we provide an example. The figure is a graph of periods and order levels for a demand realization (without a quota $D$) showing inventory levels (excess inventory or backorders) for each period. Figure 5.1 shows demands ($p_i$’s), orders ($a_i$’s), backorders (square blocks), excess units (diagonal blocks) for a seven-period instance. The $x$ axis is the cumulative orders ($A_i$’s) and the $y$ axis is time, i.e., periods.

As discussed earlier, the buyer has to decide the order levels $a_1, \ldots, a_n$ for the next $n$
Table 5.1: Comparison of the Appointment Scheduling and Inventory Models

<table>
<thead>
<tr>
<th>appointment scheduling</th>
<th>inventory</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>periods</td>
</tr>
<tr>
<td>i</td>
<td>period</td>
</tr>
<tr>
<td>( p_i )</td>
<td>demand for period ( i )</td>
</tr>
<tr>
<td>( u )</td>
<td>holding and/or disposal cost</td>
</tr>
<tr>
<td>( o )</td>
<td>backorder cost</td>
</tr>
<tr>
<td>( a_i )</td>
<td>order for period ( i )</td>
</tr>
<tr>
<td>( A_i )</td>
<td>cumulative orders up to period ( i )</td>
</tr>
<tr>
<td>( S_i )</td>
<td>total purchase (orders and backorders) up to period ( i )</td>
</tr>
<tr>
<td>( C_i )</td>
<td>total purchase (orders and backorders) up to period ( i + 1 )</td>
</tr>
</tbody>
</table>

periods now such that the total expected cost (holding and/or disposal and backorder) is minimized. We find the cost for period \( i \) first. Recall that total purchases up to period \( i \) is \( S_i \), total orders up to period \( i \) is \( A_i \), and the demand for period \( i \) is \( p_i \). Suppose we had ordered \( a_i \) for period \( i \). Then \( S_i + p_i \) is total purchases up to period \( i + 1 \), and \( A_i + a_i \) is total orders up to period \( i + 1 \). Therefore we pay holding and/or disposal cost \( u \) on \((A_i + a_i - S_i - p_i)^+\) and backorder cost \( o \) on \((S_i + p_i - A_i - a_i)^+\), hence the total cost due to period \( i \) ordering \( a_i \) units is \( u(A_i + a_i - S_i - p_i)^+ + o(S_i + p_i - A_i - a_i)^+\). By definition \( A_{i+1} = A_i + a_i \) and \( C_i = S_i + p_i \), therefore we can represent the cost of period \( i \) as \( u(A_{i+1} - C_i)^+ + o(C_i - A_{i+1})^+\). Then the total expected cost for \( n \) periods of the inventory problem becomes

\[
G(A) = \mathbb{E}_p \left[ \sum_{i=1}^{n} (o(C_i - A_{i+1})^+ + u(A_{i+1} - C_i)^+) \right].
\]  

(5.1)

We see that Eq(5.1) is precisely the definition of \( F(A) \) (given in Section 5.2) when \( u_i = u \) and \( o_i = o \) for all \( i \) \((1 \leq i \leq n)\). Therefore \( G(A) = F(A) \) with \( u_i = u \) and \( o_i = o \) for all \( i \) \((1 \leq i \leq n)\) and hence the inventory model is a special case of the appointment scheduling model. Furthermore, since \( u_i = u \) for all \( i \) \((1 \leq i \leq n)\) \( \alpha \)-monotonicity is automatically satisfied. Therefore, if the demand distributions are independent and integer-valued then \( G(A) \) can be minimized in \( O(n^9 \mathcal{P}_{\max}^2 \log \mathcal{P}_{\max}) \) time by the Theorem 2.7.3 of Chapter 2.

In addition, we see that with real-valued demands \( G \) is convex (by the Corollary 3.3.5 of Chapter 3), and we can use the sampling approach developed in Chapter 3 to obtain
a provable optimal order plan if demand distributions are not known but only a set of independent samples is available. In this case, we do not require independence of demands between one period to another. Last but not least, by our results in Appendix A we can use non-smooth convex optimization methods and a hybrid algorithm\textsuperscript{5} to find an optimal order plan for the buyer.

When there is a limit (or quota) \( D \) set by the supplier on the total number of purchases then we may represent the total expected cost for \( n \) periods of the inventory problem as

\[
G^D(\tilde{A}) = \mathbb{E}_p \left[ \sum_{j=1}^{n-1} \left( o(C_j - A_{j+1})^+ + u(A_{j+1} - C_j)^+ \right) + o_n(C_n - D)^+ + u(D - C_n)^+ \right].
\]

We immediately observe that \( G(\tilde{A}, D) = G^D(\tilde{A}) \) and furthermore, \( G^D(\tilde{A}) = F^D(\tilde{A}) \) (with \( u_{n-1} = u = u \) and \( o_i = o \) for all \( 1 \leq i \leq n - 1 \), and possibly different \( o_n \)). Similar to \( G \), for \( G^D \) \( \alpha \)-monotonicity is satisfied and hence it can be minimized in \( O(n^9 p_{\max}^2 \log p_{\max}) \) time by the Corollary 2.8.7 of Chapter 2 if the demand distributions are independent and integer-valued. Furthermore, like \( G \), with real-valued demands \( G^D \) is convex (by the Corollary 3.3.5 of Chapter 3), and by Proposition 3.4.11 of Chapter 3 we can use the sampling approach

\[\text{Figure 5.1: A Realization of Inventory Levels for 6 Periods}\]
developed in Chapter 3 to obtain a provable optimal order plan if demand distributions are not known but only a set of independent samples is available. In this case, again we do not require independence of demands between periods. Last but not least, by using our results in Appendix A and Chapter 3 (Remark A.3.11 of Appendix A and Proposition 3.4.11 of Chapter 3) we can use non-smooth convex optimization methods and the hybrid algorithm to find an optimal order plan for the buyer.

**Remark 5.3.1.** We may use distinct $u_i$’s and $o_i$’s as long as they are $\alpha$-monotone for the inventory model. However, for simplicity and the fact that it makes sense to have the same backorder and holding (and/or disposal) cost for all periods, we use the same $u$ and $o$ for all periods, except possibly for $o_n$ (to capture the extra cost incurred when there is a quota and it is exceeded). When all the $u_i$’s are the same then $\alpha$-monotonicity is satisfied and the functions $G$ and $GD$ are automatically discretely convex in the case of integer-valued demands, and convex when demand is real-valued.

We provide a real-world example for the inventory model. It comes from the high-tech industry. Consider an Internet company that has a need for high amount of bandwidth daily. The demand for the bandwidth changes from day to day and it is stochastic. The company has to make prior minimum commitments on how much bandwidth to buy for the next 30 days with an Internet service provider at the beginning of each month. Any unused amount is a lost since the company already agreed to buy some minimum quantity of bandwidth daily whereas the company can purchase more if needed in a day. However, there is the quota $D$ that the Internet provider sets for the total purchase of 30 days. If this limit is exceeded then the company must pay a penalty to get any amount over $D$. The company’s objective is to determine how much bandwidth purchase to commit for the next 30 days to minimize the expected unused amount and overuse costs. (We do not consider the cost of the service since the company will pay that amount in any case.) We can think of this company’s objective as $E_p \left[ \sum_{j=1}^{n-1} u(A_{j+1} - C_j)^+ + u(D - C_n)^+ + o(C_n - D)^+ \right]$ where $u$ is the unused amount cost rate (this cost can be thought of as the opportunity cost of unused bandwidth) and $o$ is the overuse cost for exceeding the limit $D$. Note that this function is the same as $G^D(\tilde{A})$ with $o_i = 0$ for $(1 \leq i \leq n - 1)$ and $o_n = o$.

We end this section with another application of appointment scheduling in the context the inventory model. Consider a project manager who is responsible for budget allocation.
decisions for multiple and serial phases of a project. The challenge is that funding requirements of each project phase is stochastic and they need to secured before the project start. If the funding requirement (for a phase) turns out to be more than what was allocated then it costs more (with a rate of $o$) to secure the remaining portion (e.g., the manager needs to pay a higher interest to obtain additional funds). On the other hand, if the allocation (for a phase) is more than the required funds then there is lost of some opportunity cost (with a rate of $u$). Moreover, there may a total quota $D$ for the entire project after which it costs even more to get any funding. The objective is to determine funding allocations for each phase before the project starts such that total expected cost is minimized. We can think of this problem as the same inventory problem where the demand is the funding requirements, order commitments are budget allocations and the same cost structure with $u$ and $o$.

5.4 A Multi-Period Production Model with Random Yield and Advance Commitments

Consider a production manager who is responsible for manufacturing and selling a product which is subject to random yield and high inventory holding cost. In order to secure customer contracts this manager needs to commit in advance how much to supply each period for the next $n$ periods before any production starts. Due to random yield, the production amount is subject to uncertainty however the supply commitments must be met regardless. For example, if the production falls short and the producer cannot satisfy the current period’s supply commitment from its inventory then it needs to provide the missing product amount by other means (e.g., purchase from other manufacturers). On the other hand, if there is more product available than what was committed then the excess amount is placed in inventory and can be used for future periods. However keeping inventory is expensive due the product’s high inventory holding cost.

The manager is interested to maximize the total expected profit after $n$ periods. Revenue consists of number of products sold times the contribution factor $r$ per product (including unit production cost). In our setting, number of products sold is total number of products committed. Costs include product shortage cost $u_i$ per product and inventory holding cost $o_i$ per product for period $i$ ($1 \leq i \leq n$). We think of $u_i$ as the additional cost of obtaining a product (compared to in-house production) when the producer falls short to satisfy the
current period’s commitment. On the other hand, $o_i$ can be thought as the inventory holding cost for periods 1, ..., $n - 1$, and $o_n$ (in addition to holding cost) may include disposal cost of products remaining at the end of $n$ periods.

Similar to the inventory model described in Section 5.3, this production model and the appointment scheduling model given in Chapter 2 have many similarities with respect to their structure, data, decision variables and objective function. In this production model, we have $n$ periods and in the appointment scheduling model there are $n$ jobs. The stochastic component, $p_i$ in the production model is the production level of period $i$ whereas in the appointment scheduling model it is the processing duration of job $i$. The costs are $u_i$ and $o_i$; in the appointment scheduling model $u_i$ is the underage (earliness) and $o_i$ is the overage (waiting and/or overtime) cost of job $i$, on the other hand in the production model $u_i$ is the product shortage and $o_i$ inventory holding (and possibly disposal) cost. The decision variable in the appointment scheduling model is the appointment date $A_i$ of job $i$ whereas the decision for the production model is how much to commit for supply of period $i$, $a_i$, and we have the relationship $a_i = A_{i+1} - A_i$. Completion time of job $i$, $C_i$, in the appointment scheduling model represents total number of products supplied (produced in-house and purchased from outside) until the end of period $i$ before purchases (if any) in period $i$, and start time of job $i$, $S_i$, is the total number of products supplied (produced in-house and purchased from outside) until the end of period $i - 1$ after purchases (if any) in period $i - 1$. Table 5.2 gives a comparison summary between the appointment scheduling model and the described inventory problem in terms of data and decision variables.

We can interpret Figure 5.1 of the inventory model similarly for the production model as well. Now the $x$ axis is cumulative supply commitments and, as before, the $y$ axis is time, i.e., periods. The figure shows supply commitment levels ($a_i$’s), a sample production realization ($p_i$’s), units in inventory (square blocks) and units that are short (diagonal blocks) for each period.

The manager, as mentioned above, has to decide the supply commitment levels $a_1, \ldots, a_n$ now for the next $n$ periods such that the total expected profit is maximized. We first look at revenue for period $i$. The producer sells precisely the committed amount, $a_i$, for each period; if there is shortage then the manager needs to obtain the missing amount from outside to fulfill the supply commitment, and if there is a surplus then the excess goes to inventory. Therefore the revenue for period $i$ as $ra_i$. Next, we look at costs for period $i$. Recall that
Table 5.2: Comparison of the Appointment Scheduling and Production Models

<table>
<thead>
<tr>
<th>appointment scheduling</th>
<th>production</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n )</td>
<td>periods</td>
</tr>
<tr>
<td>( i )</td>
<td>period</td>
</tr>
<tr>
<td>( p_i )</td>
<td>processing duration of job ( i )</td>
</tr>
<tr>
<td>( u_i )</td>
<td>underage cost for job ( i )</td>
</tr>
<tr>
<td>( o_i )</td>
<td>overage cost for job ( i )</td>
</tr>
<tr>
<td>( a_i )</td>
<td>allocated time for job ( i )</td>
</tr>
<tr>
<td>( A_i )</td>
<td>appointment date for job ( i )</td>
</tr>
<tr>
<td>( S_i )</td>
<td>start date for job ( i )</td>
</tr>
<tr>
<td>( C_i )</td>
<td>completion date for job ( i )</td>
</tr>
</tbody>
</table>

\( A_i \) is the cumulative supply commitment up to period \( i \). Suppose the manager promises to supply \( a_i \) for period \( i \). Then \( A_i + a_i \) is the cumulative supply commitment up to period \( i+1 \). Also note that \( C_i \) is the total products supplied (produced in-house and purchased from outside) up to period \( i+1 \) before purchase (if any) in period \( i \). Therefore if \( C_i - A_i - a_i > 0 \) then there is a product surplus and the producer pays \( o_i(C_i - A_i - a_i) \) as inventory holding cost else there is a product shortage and the producer pays \( u_i(A_i + a_i - C_i) \) to purchase the missing products. Hence the profit for period \( i \) will be \( ra_i - (o_i(C_i - A_i+1)^+ + u_i(A_i+1 - C_i)^+) \).

Then the total expected profit for \( n \) periods of the production problem becomes

\[
H(A) = E_p \left[ \sum_{i=1}^{n} (ra_i - o_i(C_i - A_i+1)^+ - u_i(A_i+1 - C_i)^+) \right] \quad (5.2)
\]

\[
= r \sum_{i=1}^{n} a_i - E_p \left[ \sum_{i=1}^{n} (o_i(C_i - A_i+1)^+ + u_i(A_i+1 - C_i)^+) \right] \quad (5.3)
\]

\[
= rA_{n+1} - F(A) \quad (5.4)
\]

where Eq(5.2) is the total expected profit for all \( n \) periods. In Eq(5.3) we take the non-random part out of the expectation and finally we obtain Eq(5.4) by noting \( \sum_{i=1}^{n} a_i = A_{n+1} \) and recognizing \( F(A) \)'s definition as given in Section 5.2. Therefore \( H(A) = rA_{n+1} - F(A) \) and hence the production model has a close connection with the appointment scheduling model. We can think of \( H \) as a special case of \( F \). Before we formalize this relationship with our next result we need a few definitions. First is the definition of \( L \)-concavity, a function \( f \) is \( L \)-concave if \(-f \) is \( L \)-convex. Formal definition is below.
Definition 5.4.1. $f : \mathbb{Z}^q \to \mathbb{R} \cup \{\infty\}$ is $L$-concave iff $f(z) + f(y) \leq f(z \lor y) + f(z \land y) \ \forall z, \forall y \in \mathbb{Z}^q$ and $\exists r \in \mathbb{R} : f(z + 1) = f(z) + r \ \forall z \in \mathbb{Z}^q$ [9].

The next two definitions are on subgradient and subdifferential for a convex function, and supgradient and supdifferential of a concave function.

Definition 5.4.2. A vector $g$ is a subgradient of a convex function $f$ at the point $x$ if $f(y) \geq f(x) + g^T(y - x)$ for all $y$. The subdifferential of $f$ at a point $x$ is the set of all subgradients at the point $x$, i.e., $\partial f(x) = \{g : f(y) \geq f(x) + g^T(y - x)\}$ [6].

Definition 5.4.3. A vector $g$ is a supgradient of a concave function $f$ at the point $x$ if $f(y) \leq f(x) + g^T(y - x)$ for all $y$. The supdifferential at of $f$ a point $x$ is the set of all supgradients at the point $x$, i.e., $\bar{\partial} f(x) = \{g : f(y) \leq f(x) + g^T(y - x)\}$ [6].

Now we are ready for our results on $H$, the objective function of the production model.

Corollary 5.4.4.

1. If production levels are integer-valued and the cost coefficients $(u, o)$ are $\alpha$-monotone then $H$ is $L$-concave. In addition if production level distributions are independent then $H$ can be maximized in $O(n^3p_{\text{max}}^2 \log p_{\text{max}})$ time.

2. If production amount is real-valued and the cost coefficients $(u, o)$ are $\alpha$-monotone then

   - $H$ is concave,
   - if $g$ is a subgradient of $F$ then $r1_{n+1} - g$ is a supgradient of $H$, i.e., if $g \in \partial F$ then $(r1_{n+1} - g) \in \bar{\partial} H(A)$ and
   - the supdifferential of $H$ at $A$ is $\bar{\partial} H(A) = r1_{n+1} - \partial F(A)$.

Proof.

1. By definition $H(A) = rA_{n+1} - F(A)$. The first term $rA_{n+1}$ is linear in $A$ and by the $L$-convexity Theorem 2.6.13 $F(A)$ is $L$-convex when the cost vectors $(u, o)$ are $\alpha$-monotone. Therefore $H(A)$ is $L$-concave.

In the case of independent production level distributions, for a given $A$, we see that the computation of complexity of $H(A)$ is the same as the computation of complexity of $F(A)$ and by Theorem 2.7.3 of Chapter 2 $F$ and $H$ can be optimized in $O(n^3p_{\text{max}}^2 \log p_{\text{max}})$ time.
2. • Note that $F$ is convex by Corollary 3.3.5 of Chapter 3 if $(u, o)$ are $\alpha$-monotone, and $rA_{n+1}$ is linear in $A$ therefore $H(A) = rA_{n+1} - F(A)$ is concave if $(u, o)$ are $\alpha$-monotone.

• If $g \in \partial F$ then $F(B) \geq F(A) + g^T(B - A)$. Since $F = rA_{n+1} - H$ we obtain $rB_{n+1} - H(B) \geq rA_{n+1} - H(A) + g^T(B - A)$. By reorganizing the terms we get $-H(B) \geq -H(A) - (r1_{n+1} - g^T)(B - A)$. Finally we multiply both sides with $-1$ and obtain $H(B) \leq H(A) + (r1_{n+1} - g^T)(B - A)$. Therefore $(r1_{n+1} - g^T)$ is a supgradient of $H$ at $A$, i.e., $(r1_{n+1} - g) \in \partial H(A)$.

• The above results gives If $g \in \partial F$ then $(r1_{n+1} - g) \in \partial H(A)$. This also shows $\partial H(A) \supseteq r1_{n+1} - \partial F(A)$. By using the same arguments above we also see that if $g \in \partial H(A)$ then $(r1_{n+1} - g^T)$ is a subgradient of $F$ showing $\partial H(A) \subseteq r1_{n+1} - \partial F(A)$. Hence we obtain $\partial H(A) = r1_{n+1} - \partial F(A)$.

□

Corollary 5.4.4 allows us to solve the production problem with the tools developed for the appointment scheduling problem in Chapter 2 and Chapter 3. In the case of independent and integer-valued production level distributions we can maximize $H$ in polynomial time. Furthermore, with real-valued production levels $H$ is concave and we obtain a supgradient for $H$ easily once we have a subgradient for $F$. Therefore we can utilize the results in Appendix A and use non-smooth concave optimization methods and the hybrid algorithm to find an optimal production plan for the manager. Last but not least, we characterize $H$’s supdifferential and can use the sampling approach developed in Chapter 3 to obtain a provable near-optimal production plan if production level distributions are not known but only a set of independent samples is available. As discussed earlier, in this case, we do not require independence of production levels between one period to other.

5.5 Conclusion

We introduce two advance multi-period quantity (order or supply) commitment problems with a random demand or yield. The distinct feature of these models with previous ones reported in literature is that all quantity commitment (order and supply amounts) decisions are to be made at once and before the planning interval starts. We show that there is a
close relationship between these problems and the appointment scheduling problem studied in Chapter 2. Therefore, we can solve these type of quantity commitment problems efficiently, e.g., in polynomial time in the case of integer-valued and independent (demand or yield) distributions or by using non-smooth convex optimization methods developed in Appendix A. Furthermore, in the case of unknown demand or yield distributions we can use the sampling approach developed in Chapter 3.
5.6 Bibliography


6 Concluding Remarks

In this thesis, we take a in-depth look at the appointment scheduling problem [2, 1, 7]. In Chapter 2, we study a discrete time version of the appointment scheduling problem and develop a polynomial time algorithm, based on discrete convexity, that, for a given processing sequence, finds an appointment schedule minimizing the total expected cost. To the best of our knowledge this is the first polynomial time algorithm for the appointment scheduling problem. In addition, our framework can handle a given due date for the total processing (e.g., end of day for an operating room) after which overtime is incurred, instead of letting the model choose an end date. We also extend our model and framework to include no-shows (e.g., patient no-shows) and some emergencies (e.g., emergency surgeries).

We believe that our framework is sufficiently generic so that it is portable and applicable to many appointment systems in healthcare and other areas including surgery scheduling, healthcare diagnostic operations (such as CAT scan, MRI) and physician appointments, as well as project scheduling, container vessel and terminal operations, gate and runway scheduling of aircraft in an airport.

After developing our modeling framework and proving that we can find an optimal appointment schedule in polynomial time, we focus on practical implementation issues in Chapter 3 and Appendix A. The objective function of the appointment scheduling problem as a function of continuous appointment vector is non-smooth but in Chapter 3 we show that it is convex, and we characterize its subdifferential. We obtain closed form formulas for the subdifferential as well as for any subgradient. This characterization is very useful as it allows us to develop two very important extensions. In Chapter 3, we relax the perfect information assumption on the probability distributions of processing durations. We develop a sample-based approach to determine the number of independent samples required to obtain a provably near-optimal solution with high confidence. This result has important practical implications, as the true processing duration distributions are often not known and only their past realizations or some samples are available. We believe this is the first sampling
approach developed for the appointment scheduling problem. In Appendix A, we use the subdifferential characterization with independent processing durations to develop a hybrid approach based on both discrete convexity [4] and non-smooth convex optimization [3, 6] combined with a special-purpose rounding algorithm which takes any fractional solution and rounds it to an integer one with the same or improved objective value. We believe the hybrid approach may perform well in practice.

Again motivated by surgery scheduling, in Chapter 4 we look at the problem of determining the number of surgeries for an OR block with a focus on the incentives of the parties involved (hospital and surgeon). In particular, we investigate the commonly observed situation reported in the literature and observed empirically that surgeons over-schedule their allotted OR time, i.e., they schedule too many surgeries for their OR time and cause excessive overtime. We argue that this can be explained by the incentive of surgeons to take advantage of fee-for-service payment structure for surgeries performed combined with the fact surgeons do not bear overtime costs at the hospital level. This creates a cost which is borne by the hospital, which operates the OR and pays surgery support staff. We propose contracts that induce the surgeon to schedule a number of surgeries more aligned with the goals of the hospital and thus reduce overtime. If an OR can be managed in such a way that overtime is decreased then this may translate to immediate and significant cost savings which may be used to increase hospital resources such as regular OR time, recovery and intensive care beds. Depending on how much power the hospital has over surgeons and how much information is available to the hospital, we suggest several contracts that hospital might consider.

There is a connection between the celebrated newsvendor problem and the appointment scheduling problem. If we have only a single job (surgery), i.e., \( n = 1 \), then the appointment scheduling problem becomes the newsvendor problem [8]. In Chapter 5, we introduce a new set of advance multi-period quantity (order or supply) commitment problems with random characteristics (demand or yield), and underage and overage costs if there is a mismatch between committed and realized quantities. We show that these multi-period quantity commitment problems can be modeled and solved as special cases of the appointment scheduling problem. To the best of our knowledge, the problems introduced in Chapter 5 have not been yet studied.

There are exciting future directions and improvement possibilities for this research.
One possibility is to find an optimal sequence and appointment schedule simultaneously, i.e., given the jobs, determine a sequence and a job appointment schedule minimizing the total expected cost. This problem is likely to be hard [5], but it may be possible to develop heuristic algorithms with performance guarantees. Studying some special cases for this problem may shed light on the general case.

In the near future, we are planning to implement the algorithms in Appendix A and develop a computational engine for the appointment scheduling problem. Besides testing and comparing the discrete, non-smooth and hybrid algorithms in computational experiments we plan to put our findings into practice. We are in contact with local healthcare organizations to apply our results with real data and compare the appointment schedules determined by our methods with current practices. Furthermore, we may test performance of various heuristic methods for both appointment scheduling and sequencing problem once the computational engine is built.

One other research avenue that we consider is an extension in which both job arrivals and durations are random, and jobs belong to different priority classes. The goal is to determine a booking policy such that each priority target waiting times are satisfied at minimum cost. Or alternatively, one can find what would be the waiting times for a given budget.

Last but not least, there are many interesting incentive problems in healthcare. For example, in generating an OR block scheduled there is an issue of allocating available OR time between different specialties as well as between surgeons within a specialty. Each surgeon has her/his number of patients waiting for surgery. Furthermore, there some other non-medical factors to consider such as the rank of the surgeon as well as how well the specialty is represented in OR block allocation process. We consider studying this problem and develop a mechanism that will be fair and transparent for everyone with an overall objective to reduce surgical waiting times.
6.1 Bibliography


A Minimizing a Discrete-Convex Function for Appointment Scheduling

We consider the appointment scheduling problem with discrete random durations studied in Chapter 2. Under a simple sufficient condition, the objective of the appointment scheduling problem is discretely convex as a function of the integer appointment vector (Chapter 2), but it is convex but non-smooth when appointment vectors are continuous (Chapter 3). In this chapter, we compute a subgradient of the objective function in polynomial time for any given (real-valued) appointment schedule with independent processing durations. We also extend computation of the expected total cost (in polynomial time) for any (real-valued) appointment vector. Furthermore, we develop a special-purpose integer rounding algorithm that allow us to develop an hybrid approach combining both discrete convexity and non-smooth convex optimization methods. We plan to implement these algorithms and compare different approaches in computational experiments.

A.1 Introduction and Motivation

We consider the appointment scheduling problem with discrete random durations studied in Chapter 2. The goal of appointment scheduling is to determine an optimal planned start schedule, i.e., an optimal appointment schedule for a given sequence of jobs on a single processor such that the expected total underage and overage costs is minimized. In Chapter 2, we showed that the objective function of the appointment scheduling problem is discretely convex (under $\alpha$-monotonicity) and there exists an optimal integer appointment schedule minimizing the objective over integer appointment vectors. These results on the objective function and optimal appointment schedule enabled us to develop a polynomial time algorithm, based on discrete convexity, that, for a given processing sequence, finds an

\footnote{A version of this chapter will be submitted for publication. Begen M.A. and Queyranne M. Minimizing a Discrete-Convex Function for Appointment Scheduling.}
appointment schedule minimizing the total expected cost.

On the other hand, in Chapter 3 we considered the same appointment scheduling problem in Chapter 2 under the assumption that the duration probability distributions are not known and only a set of independent samples is available, e.g., historical data. We showed that, under a simple sufficient condition, the same objective function is convex (as a function of continuous appointment vector) and non-smooth. Under this condition we characterized the subdifferential of the objective function with a closed-form formula. This characterization is useful; it allows us to develop two very important extensions. First, we used it to determine bounds on the number of independent samples required to obtain provably near-optimal solution with high probability in Chapter 3. Second, we use it in this chapter to obtain a subgradient in polynomial time to use subgradient methods (with discrete methods) to optimize the appointment scheduling objective.

In this chapter, we use the subdifferential characterization of Chapter 3 with independent processing durations and compute a subgradient in polynomial time for any given appointment schedule. The reason we are after a fast obtainable subgradient is to use non-smooth convex optimization methods to find an optimal appointment schedule. From Chapter 2 we already have a polynomial time algorithm to minimize the objective and obtain an optimal appointment schedule, however it is not clear at the moment which technique (discrete or non-smooth) will work faster in practice.

Finding a subgradient in polynomial time is not trivial because the subdifferential formulas include exponentially many terms, and some of the probability computations are complicated. In addition to a subgradient, we obtain an easily computable lower bound on the optimal objective value. Furthermore, we extend computation of the expected total cost (in polynomial time) for any (real-valued) appointment vector. These results allow us to use non-smooth convex optimization techniques to find an optimal schedule.

To combine the discrete and non-smooth algorithms, we develop a special-purpose integer rounding method which takes any fractional solution and rounds it to an integer one with the same or improved objective value. This rounding algorithm enable us to develop a hybrid approach combining both discrete convexity and non-smooth convex optimization methods.

In the near future, we are planning to implement our algorithms and compare different approaches in computational experiments.
This chapter is organized as follows\textsuperscript{2}. We start with finding a lower bound on the objective functions $F$ and $F^D$ in Section A.2. In this section, we also extend the computation of $F(A)$ (and hence $F^D(\tilde{A})$) for any real appointment vector $A$. In Section A.3, we find a subgradient of $F$ (and $F^D$) in polynomial time. We first compute probabilities required to obtain a subgradient from subdifferential $\partial F(A)$ and discuss the complexity of this computation. Then we show how to find a subgradient in polynomial time. In Section A.4, we develop the rounding algorithm and discuss how it can be used with the existing discrete and non-smooth algorithms to build a hybrid approach. Finally we conclude the chapter in Section A.5.

\textbf{A.2 Lower Bounds (on the Value) and Computation of $F(A)$ and $F^D(\tilde{A})$ and for any Real $A$}

In this section, we find an easily computable lower bound on the values of $F(A)$ and $F^D(\tilde{A})$. When the underage cost coefficients $u_i$'s ($1 \leq i \leq n$) are identical $F(A)$ has additional interesting properties. In this case, $\alpha$-monotonicity is (automatically) satisfied and hence $F(A)$ is convex by Convexity Proposition 3.3.3 of Chapter 3 (and the same result follows for $F^D(\tilde{A})$ from Corollary 3.3.5 of Chapter 3). Furthermore, a lower bound on the value of $F(A)$ can be easily computed.

A remark is in order here. If underage cost coefficients are not identical for $F(A)$ then define $u \min\{u_1, u_2, ..., u_n\}$ and

$$f(A) = E_p \left[ \sum_{j=1}^{n} (o_j(C_j - A_{j+1})^+ + u(C_j - A_{j+1})^+) \right],$$

eq i., replace $u_i$ with $u$ for all $i$ ($1 \leq i \leq n$), and find a lower bound for $f(A)$ as described in this section. Note that since $f(A) \leq F(A)$ the obtained lower bound of $f(A)$ will be a lower bound for $F(A)$ with non-identical $u_i$'s.

We start by expressing $F(A)$ in a different but an equivalent way that will be essential in finding a lower bound on the value of $F(A)$ (and $F^D(\tilde{A})$). Our first result is a Corollary to Lemma 3.3.1 in Chapter 3.

\textsuperscript{2}Since this chapter is included to the thesis as an appendix, we will omit notation introduction and formal description of appointment problem and refer the reader to Chapter 2 and Chapter 3.
Corollary A.2.1. If \( u_i = u \) for all \( i \) \((1 \leq i \leq n)\) then
\[
F(A) = E_p \left[ \sum_{j=1}^{n} (o_j (C_j - A_{j+1})^+ + u(C_j - A_{j+1})^+) \right] = E_p \left[ \sum_{j=1}^{n} (o_j (C_j - A_{j+1})^+ + u(max\{A_{n+1}, C_n\} - \sum_{k=1}^{n} p_k) \right].
\]

Proof. The proof is an application of Lemma 3.3.1 with specific \( \alpha_i \) \((1 \leq i \leq n)\). Choose \( \alpha_i = 0 \) \((1 \leq i \leq n)\). Then \( \beta_i = \alpha_i \) \((1 \leq i \leq n)\), \( \gamma_i = 0 \) \((1 \leq i < n)\) and \( \gamma_n = u_n = u \). Now the result directly follows from Lemma 3.3.1. \( \square \)

We need the following definition before computing a lower bound on \( F(A) \).

Definition A.2.2. (Single Period) Newsvendor Problem \[8\]
A newsvendor needs to decide the number of units \( Q \) to be purchased before the demand \( Y \) is realized. The newsvendor pays \( c_h \) for each unit remaining unsold and pays \( c_p \) for each unit of unsatisfied demand. The objective of the newsvendor is to choose the \( Q \) minimizing the expected cost. This problem is well studied and it has a closed form solution. Let \( H(Y) \) be the cumulative distribution of \( Y \) and \( Q^* \) the optimum solution then
\[
H(Q^*) = \frac{c_p}{c_h + c_p}.
\]

We may think each job \( j \) \((1 \leq j \leq n + 1)\) as a single period newsvendor problem if we had only that job to process and find its solution as given in Definition A.2.2. We use this idea to obtain a lower bound on \( F(A) \) in our next result.

Proposition A.2.3. Let \( u_i = u \) \((1 \leq i \leq n)\), \( A^* \) be an optimal appointment vector for \( F(A) \) and \( a_j^* \) the (single period) newsvendor solution for job \( j \) \((1 \leq j \leq n + 1)\). Then
\[
\sum_{j=1}^{n} E_p \left[ o_j (p_j - a_j^*)^+ + u(a_j^* - p_j)^+ \right]
\]
is a lower bound for \( F(A^*) \).

Proof. Consider the following optimization problem.

\[
OPT1 \left\{ \begin{array}{l}
\text{min}_A E_p \left[ \sum_{j=1}^{n} (o_j (C_j^P - A_{j+1})^+ + u(max\{A_{n+1}, C_n^P\} - \sum_{k=1}^{n} p_k) \right] \\
\text{subject to} & A_1 = 0, C_1^P = p_1, C_j^P = max(A_j, C_{j-1}^P) + p_j \text{ for } (2 \leq j \leq n), \text{ all } p.
\end{array} \right.
\]

159
OPT1 is the stochastic program for minimizing $F(A)$ hence its optimum will be $F(A^*)$.
We need all the constraints of OPT1 as they are the essential dynamics of our scheduling problem.

We rewrite the constraints $C_j^p = \max(A_j, C_{j-1}^p) + p_j$ as $C_j^p \geq A_j + p_j$ and $C_j^p \geq C_{j-1}^p + p_j$
for all $p$ and obtain OPT2:

\[
\begin{align*}
\text{OPT2} & \quad \min_A E_p \left[ \sum_{j=1}^n o_j(C_j^p - A_{j+1})^+ + u(\max\{A_{n+1}, C_n^p\} - \sum_{k=1}^n p_k) \right] \\
\text{subject to} & \quad A_1 = 0, C_1^p = p_1, C_j^p \geq A_j + p_j \text{ and } C_j^p \geq C_{j-1}^p + p_j \text{ for } (2 \leq j \leq n), \text{ all } p.
\end{align*}
\]

Since objective function coefficients $o_j$ ($1 \leq j \leq n-1$), $o_n$, $u$ are all non-negative and the objective function is a non-decreasing function of $C_j^p$'s OPT1 and OPT2 are equivalent. Since OPT1 and OPT2 are equivalent, $A^*$ is also an optimum appointment vector for OPT2.

Now we relax $C_j^p \geq C_{j-1}^p + p_j$ constraints from OPT2 and the obtain relaxation OPT3:

\[
\begin{align*}
\text{OPT3} & \quad \min_A E_p \left[ \sum_{j=1}^n o_j(C_j^p - A_{j+1})^+ + u(\max\{A_{n+1}, C_n^p\} - \sum_{k=1}^n p_k) \right] \\
\text{subject to} & \quad A_1 = 0, C_1^p = p_1, C_j^p \geq A_j + p_j \text{ for } (2 \leq j \leq n), \text{ all } p.
\end{align*}
\]

Observe that OPT3 is a relaxation of OPT2 and it decomposes into $n$ independent optimization problems, one for each $j$. Furthermore, the $C_j^p \geq A_j + p_j$ constraints will be binding in any optimal solution (since objective function coefficients are all non-negative and the objective function is a non-decreasing function of $C_j^p$'s). Let $a_j = A_{j+1} - A_j$ for $1 \leq i \leq n$ then by using $C_j^p = A_j + p_j$ for $1 \leq i \leq n$ we can rewrite OPT3 as

\[
\min_a E_p \left[ \sum_{j=1}^n \left( o_j(p_j - a_j)^+ + u(\max\{A_{n+1}, A_{n+1} - a_n + p_n\} - \sum_{k=1}^n p_k) \right) \right].
\]

By Corollary A.2.1, OPT3 may be written equivalently as

\[
\min_a E_p \left[ \sum_{k=1}^n \left( o_k(p_k - a_k)^+ + u(a_k - p_k)^+ \right) \right],
\]

and this is nothing but sum of $n$ independent newsvendor problems so we can minimize it by setting $a_i^* = P_i^{-1}(o_i/(o_i + u))$ for $1 \leq i \leq n$ where $P_i^{-1}(.)$ is the inverse cumulative distribution of job duration $i$ ($1 \leq i \leq n$). Therefore the result follows. \hfill \Box
**Remark A.2.4.** Let \( \underline{f}(A) = E_p \left[ \sum_{k=1}^{n} (o_k(p_k - (A_{k+1} - A_k))^+ + u((A_{k+1} - A_k) - p_k))^+ \right] \), then \( \underline{f}(A) \) is not necessarily a lower bound for \( F(A) \). Consider the following example with deterministic processing times. Let \( n = 4, p_1 = 4, p_2 = 6, p_3 = 1, p_4 = 1, A_1 = 0, A_2 = 3, A_3 = 6, A_4 = 9 \) and \( A_5 = 13 \). Then \( \underline{f}(A) = o_1 + 3o_2 + 2u + 3u \) and \( F(A) = o_1 + 4o_2 + 2o_3 + u \). So for \( u = o_2 = o_3, \underline{f}(A) > F(A) \).

However, we find a different lower bound function (as a function of \( A \)) for \( F(A) \) in Lemma 3.5.6 in Chapter 3.

Next we find a lower bound on the value of objective with a due date \( D \), i.e., on the value of \( F^D(\tilde{A}) \). We obtain a lower bound similar to that in Proposition A.2.3 on the value of \( F^D(\tilde{A}) \).

**Corollary A.2.5.** Let \( u_i = u \) (1 \( \leq i \leq n \)), \( \tilde{A}^* \) be an optimal appointment vector for \( F^D(\tilde{A}) \) and \( a_j^* \) the (single period) newsvendor solution for job \( j \) (1 \( \leq j \leq n + 1 \)). Then \( \sum_{j=1}^{n} E_p \left[ o_j(p_j - a_j^*)^+ + u(a_j^* - p_j)^+ \right] \) is a lower bound for \( F^D(\tilde{A}) \).

**Proof.** Consider the following optimization problem:

\[
\min_A \ E_p \left[ \sum_{j=1}^{n} (o_j(C_j^P - A_{j+1})^+) + u(\max\{A_{n+1}, C_n\} - \sum_{k=1}^{n} p_k) \right]
\]

subject to

\[
A_{n+1} = D, \ A_1 = 0, \ C_1^P = p_1, \ C_j^P = \max(A_j, C_{j-1}^P) + p_j \quad \text{for} \ (2 \leq j \leq n), \ \text{all} \ p.
\]

If we relax the \( A_{n+1} = D \) constraint of \( OPT^D \), we immediately obtain the optimization problem for \( F(A) \) (of course with \( u_i = u \)). Therefore \( F(A) \leq F^D(\tilde{A}) \).

On the other hand, \( \sum_{j=1}^{n} E_p \left[ o_j(p_j - a_j^*)^+ + u(a_j^* - p_j)^+ \right] \) is a lower bound for \( F(A) \) by Proposition A.2.3. Hence

\[
\sum_{j=1}^{n} E_p \left[ o_j(p_j - a_j^*)^+ + u(a_j^* - p_j)^+ \right] \leq F(A) \leq F^D(\tilde{A}).
\]

This completes the proof. \( \square \)

We now show how to compute \( F(A) \) (and \( F^D(\tilde{A}) \)) if processing durations are independent and \( A \) is not integer. The following result is due to Theorem 2.7.2 in Chapter 2 and an observation to keep track of previous (potential) fractional points. This extra bookkeeping in the case of non-integer appointment vectors cost a factor of \( n \) in the complexity of \( F(A) \) computation.
Corollary A.2.6. If the processing durations are stochastically independent and $A$ is a real appointment vector then $F(A)$ may be computed in $O(n^3\overline{p}_{\text{max}}^2)$ time.

**Proof.** The first job starts at time zero so $S_1 = A_1 = 0$, and $C_1 = p_1$, i.e., the distribution of $C_1$ is that of $p_1$. Next, we look at the start times $S_i$ ($2 \leq i \leq n$). We have $S_i = \max(A_i, C_{i-1})$ so for all $k$ and,

$$
\text{Prob}\{S_i = k\} = \begin{cases} 
0 & \text{if } k < A_i \\
\text{Prob}\{C_{i-1} \leq k\} & \text{if } k = A_i \\
\text{Prob}\{C_{i-1} = k\} & \text{if } k > A_i.
\end{cases} \tag{A.1}
$$

Note that $S_i$ and $p_i$ are independent because $S_i$ is completely determined by $p_1, p_2, \ldots, p_{i-1}$ and $A_1, A_2, \ldots, A_i$. A remark is in order here. If $A$ is integer then $k = 0, 1, \ldots, n\overline{p}_{\text{max}}$, however when $A$ is not integer then in addition to previous integer values we also need to consider (possible distinct) fractional values arising from non-integer $A_i$ values. $A_1 = 0$ so there will be at most $n\overline{p}_{\text{max}}$ integer values for $k$, from $A_2$ at most $(n-1)\overline{p}_{\text{max}}$, from $A_3$ at most $(n-2)\overline{p}_{\text{max}}$ and so on. Therefore in total we need to consider at most $n^2\overline{p}_{\text{max}}$ distinct values (it was only $n\overline{p}_{\text{max}}$ if $A$ is integer).

Since $C_i = S_i + p_i$, by conditioning on $p_i$ and using independence of $p_i$ and $S_i$, we obtain for all $k$,

$$
\text{Prob}\{C_i = k\} = \text{Prob}\{S_i = k - p_i\} = \sum_{j=0}^{p_i} \text{Prob}\{S_i = k - j\}\text{Prob}\{p_i = j\}, \tag{A.2}
$$

and $\text{Prob}\{C_{i-1} \leq k\} = \text{Prob}\{C_{i-1} = k\} + \text{Prob}\{C_{i-1} \leq k-1\}$. For each $i-1$, $\text{Prob}\{C_{i-1} \leq k\}$ may be computed in $O((i-1)^2\overline{p}_{\text{max}})$ time. Hence $\text{Prob}\{C_i = k\}$ can be computed once we have the distribution of $S_i$. For each job $i$ and value $k$, computing $\text{Prob}\{S_i = k\}$ by Eq(A.1) requires a constant number of operations, and computing $\text{Prob}\{C_i = k\}$ by Eq(A.2) requires $O(p_i + 1)$ operations. Therefore the total number of operations needed for computing the whole start time and completion time distributions for job $i$ is $O(n^2\overline{p}_{\text{max}}^2)$. The distribution of $T_i$ and $E_i$, their expected values $E_p T_i$ and $E_p E_i$ can then be determined in $O(n^2\overline{p}_{\text{max}}^2)$ time. Therefore, the objective value $F(A)$ is obtained in $O(n^3\overline{p}_{\text{max}}^2)$ time. □

When the appointment vector is not integer, the size of possible values (for completion and start times) grows with a factor of $n^2$, as compared to $n$ in the discrete case shown above. However, note that in practice, some components may have the same fractions, and this will speed up the computation.
In this section we compute probabilities

\[ \text{A.3.1 Probability Computations} \]

\[ \text{For easier computation and later purposes, we rewrite } g_k(X, A) \text{ for a given convex hull weight vector } X, \text{ i.e., } g_k(X, A), \text{ } k^{th} \text{ coordinate of a subgradient at the point } A \text{ for a particular } X. \text{ By Eq(3.16) recall that } g_k(X, A) \text{ is given by} \]

\[
\begin{align*}
\sum_{j=k}^{n} \alpha_j & \sum_{S \in P^*(\{j\})} \text{Prob}(I_j = S) X_{kj}^k(S) - \alpha_{k-1} \sum_{S \in P^*([k-1])} \text{Prob}(I_{k-1} = S) \\
+ \sum_{j=k}^{n} \beta_j & \sum_{S \in P^*(\{j\})} \text{Prob}(I_j^\gamma = S) X_{kj}^\gamma(S) - \beta_{k-1} \sum_{S \in P^*([k-1])} \text{Prob}(I_{k-1}^\gamma = S) \\
+ \sum_{j=k}^{n} \gamma_j & \sum_{S \in P^*(\{j\})} \text{Prob}(I_j^\gamma = S) X_{kj}^\gamma(S) + \gamma_{k-1} \sum_{S \in P^*([k-1])} \text{Prob}(I_{k-1}^\gamma = S) \\
+ \sum_{j=k}^{n} \gamma_j & \sum_{S \in P^*(\{j\})} \text{Prob}(I_j^m = S) X_{kj}^m(S \cup \{j+1\}) + \gamma_{k-1}(1 - \sum_{S \in P^*([k-1])} \text{Prob}(I_{k-1}^m = S)).
\end{align*}
\]

To obtain a subgradient of \( F \), as seen in Eq(A.3), we need to compute some probability terms, e.g., \( \text{Prob}(I_{k-1}^m = S) \), choose an appropriate (i.e., feasible) \( X \) vector and need to find a way to deal with exponentially many \( S \in P^*([j]) \) terms.

Our strategy is to compute the probabilities first. Then we discuss the complexity of obtaining a subgradient. Finally we show a way to obtain a subgradient (in fact two subgradients) fast, i.e., in polynomial time.

\[ \text{A.3.1 Probability Computations} \]

In this section we compute probabilities \( \text{Prob}(I_j = S) \) and \( \text{Prob}(I_j^\eta = S) \) for \( S \in P^*([j]) \), \( j \in [n+1] \) and \( \eta \in \{>,=,<\} \). Recall that, by Eq(3.4) and Eq(3.6) we have

\[
I_j = \arg \max_{k \leq j} \{A_k + P_{kj}\}
\]

\[
I_j^\eta = \{k \in I_j : A_k + P_{kj} \eta A_{j+1}\}
\]

For easier computation and later purposes, we rewrite \( I_j^\eta \). Let \( \max I_j = \max\{k : k \in I_j\} \) then

\[
I_j^\eta = \{k \in I_j : A_k + P_{kj} \eta A_{j+1}\} = \{k \in I_j : A_{\max I_j} + P_{\max I_j, j} \eta A_{j+1}\}.
\]

\[ 163 \]
Eq(A.4) follows from the fact that \( A_k + P_{kj} \) is the same quantity for any \( k \in I_j \) and particularly for \( k = \max I_j \). Let \( i_s = \max\{i : i \in S\} \) then by using Eq(A.4) we get

\[
Prob\{I_j^n = S\} = Prob\{I_j = S \text{ and } A_{i_s} + P_{i_s,j} \eta A_{j+1}\}. \tag{A.5}
\]

We first compute \( Prob\{I_j = S\} \). Let \( S \in P^*([j]) \) then

\[
Prob\{I_j = S\} = Prob \left\{ \left( \bigcap_{i \in S} (i \in I_j) \right) \cap \left( \bigcap_{k \in [j] - S} (k \notin I_j) \right) \right\}. \tag{A.6}
\]

Eq(A.6) simply means that for two sets to be equal, they need to agree on each term, i.e., each element of \( S \) should be an element of \( I_j \) and an element which is not in \( S \) should not be in \( I_j \). We now provide an intuitive result which is also crucial in computing \( Prob\{I_j = S\} \).

**Lemma A.3.1.** If \( i \in I_j \) then job \( i \) starts on time, i.e., \( S_i = A_i \).

**Proof.** For \( i = 1 \) the result holds since job 1 always starts on time (i.e., at time \( A_1 = 0 \)).

For \( 2 \leq i \leq n + 1 \) we will show the contrapositive. If job \( i \) does not start on time then job \( i \) is late, i.e., completion date of job \( i - 1 \) is strictly greater than appointment date of job \( i \) (\( C_{i-1} > A_i \)). Therefore job \( i \) cannot be on the critical path of \( C_j \) and hence \( i \notin I_j \). \( \square \)

We give an example to illustrate Eq(A.6) and the application of Lemma A.3.1.

**Example A.3.2.** Let \( S = \{1, 2, 3, 5\} \) then

\[
Prob\{I_5 = \{1, 2, 3, 5\}\} = Prob\{1 \in I_5 \text{ and } 2 \in I_5 \text{ and } 3 \in I_5 \text{ and } 4 \notin I_5 \text{ and } 5 \in I_5\}.
\]

We can visualize the event \( \{I_5 = \{1, 2, 3, 5\}\} \) as depicted in Figure A.1.

We use Lemma A.3.1 in the following arguments to deduce the fact that if \( i \in I_j \) then job \( i \) starts on time.

Let’s examine probability \( Prob\{I_5 = \{1, 2, 3, 5\}\} \) by starting from the end. \( 5 \in I_5 \) tells us that job 5 starts at time \( A_5 \). \( 4 \notin I_5 \) and \( 3 \in S \) imply that job 3 starts on time, and we
must have $p_3 > A_4 - A_3$ (if $p_3 = A_4 - A_3$ then $4 \in I_5$, if $p_3 < A_4 - A_3$ then $A_3 \notin I_5$) and $P_{34} = A_5 - A_3$ since both 3 and 5 $\notin I_5$ (if $P_{34} > A_5 - A_3$ then 5 $\notin I_5$, if $P_{34} < A_5 - A_3$ then 3 $\notin I_5$). Similarly, 2 $\in I_5$ gives us that job 2 starts on time and $p_2 = A_3 - A_2$. Finally, 1 $\in I_5$ implies that $p_1 = A_2 - A_1$ (note that $A_1 = 0$), otherwise either 1 $\notin I_5$ (if $p_1 < A_2 - A_1$) or 2 $\notin I_5$ (if $p_1 > A_2 - A_1$). Therefore,

$$Prob\{I_5 = \{1, 2, 3, 5\}\}$$

$$= Prob\{1 \in I_5 \text{ and } 2 \in I_5 \text{ and } 3 \in I_5 \text{ and } 4 \notin I_5 \text{ and } 5 \in I_5\}$$

$$= Prob\{1 \in I_5 \text{ and } 2 \in I_5 \text{ and } p_3 > A_4 - A_3 \text{ and } P_{34} = A_5 - A_3\}$$

$$= Prob\{1 \in I_5 \text{ and } p_2 = A_3 - A_2 \text{ and } p_3 > A_4 - A_3 \text{ and } P_{34} = A_5 - A_3\}$$

$$= Prob\{p_1 = A_2 - A_1 \text{ and } p_2 = A_3 - A_2 \text{ and } p_3 > A_4 - A_3 \text{ and } P_{34} = A_5 - A_3\}$$

$$= Prob\{p_1 = A_2 - A_1\} \cdot Prob\{p_2 = A_3 - A_2\} \cdot Prob\{p_3 > A_4 - A_3\} \cdot Prob\{P_{34} = A_5 - A_3\}.$$ 

The last equality follows from the independence of job duration distributions. Furthermore, the convolutions such as $P_{34}$ and their corresponding probabilities such as $Prob\{p_3 > A_4 - A_3 \text{ and } P_{34} = A_5 - A_3\}$ can be computed efficiently as our job duration distributions are discrete and independent. We will compute these probabilities with the RCA (Recursive Computation Algorithm) developed later in the section.

We provide another example to show the other possibilities that may occur in $Prob\{I_j = S\}$ computation.

**Example A.3.3.** Let $S = \{3, 5, 6\}$. Then

$$Prob\{I_S = \{3, 5, 6\}\}$$

$$= Prob\{1 \notin I_S \text{ and } 2 \notin I_S \text{ and } 3 \in I_S \text{ and } 4 \notin I_S \text{ and } 5 \in I_S \text{ and } 6 \in I_S \text{ and } 7 \notin I_S \text{ and } 8 \notin I_S\}$$

Event $\{I_S = \{3, 5, 6\}\}$ may be visualized as shown in Figure A.2.

As in Example A.3.2, we use Lemma A.3.1 to deduce the fact that if $i \in I_j$ then job $i$ starts on time. We compute $Prob\{I_S = \{3, 5, 6\}\}$ by starting from the end and by considering blocks of jobs between successive elements of $S$. 

Figure A.2: Event $\{I_S = \{3, 5, 6\}\}$ Visualization
6 \in I_8 \text{ gives us that job 6 starts on time}. 7 \not\in I_8 \text{ implies } A_6 + p_6 > A_7 \text{ (if } A_6 + p_6 = A_7 \text{ then } 7 \in I_8, \text{ if } A_6 + p_6 < A_7 \text{ then } 6 \not\in I_8). \text{ Similarly, } 8 \not\in I_8 \text{ implies } A_6 + p_{67} > A_8.

5 \in I_8 \text{ means that job 5 starts on time and hence } A_5 + p_5 = A_6 \text{ (if } A_5 + p_5 > A_6 \text{ then } 6 \not\in I_8, \text{ if } A_5 + p_5 < A_6 \text{ then } 5 \not\in I_8).\n
3 \in I_8 \text{ implies that job 3 starts on time}. 4 \not\in I_8 \text{ tells us that } A_3 + p_3 > A_4 \text{ (if } A_3 + p_3 = A_4 \text{ then } 4 \in I_8, \text{ if } A_3 + p_3 < A_4 \text{ then } 3 \not\in I_8). \text{ Furthermore, } 3 \in I_8 \text{ also implies that } A_3 + p_{34} = A_5 \text{ (if } A_3 + p_{34} < A_5 \text{ then } 3 \not\in I_8, \text{ if } A_3 + p_{34} > A_5 \text{ then } 5 \not\in I_8).

The remaining jobs, jobs 1 and 2, are not in $I_8$. This implies that completion time of job 2 $C_2$ is strictly less than $A_3$ (if $C_2 > A_3$ then $3 \not\in I_8$, if $C_2 = A_3$ then $1 \in I_8$ or $2 \in I_8$ or both $1, 2 \in I_8$). If we collect together the arguments above, we obtain:

\[
\text{Prob}\{I_8 = \{3, 5, 6\}\} = \text{Prob}\{1 \not\in I_8 \text{ and } 2 \not\in I_8 \text{ and } 3 \in I_8 \text{ and } 4 \not\in I_8 \text{ and } 5 \in I_8 \text{ and } 6 \in I_8 \text{ and } 7 \not\in I_8 \text{ and } 8 \not\in I_8\} = \text{Prob}\{1 \not\in I_8 \text{ and } 2 \not\in I_8 \text{ and } 3 \in I_8 \text{ and } 4 \not\in I_8 \text{ and } 5 \in I_8 \text{ and } A_6 + p_6 > A_7 \text{ and } A_6 + p_{67} > A_8\}
\]

As in Example A.3.2, the last equality follows from the independence of job duration distributions.

We next obtain a formula for computing $\text{Prob}\{I_j = S\}$ for any $S \in \mathcal{P}^*([j])$. The nice thing about this probability is that it breaks down into blocks of independent events with the elements of $S$ as seen in Examples A.3.2 and A.3.3. We now prove this result.

**Lemma A.3.4.** Write any subset $S \in \mathcal{P}^*([j])$ as an ordered list: $S = \{i_1, \ldots, i_s\}$ where
\(i_l < i_{l+1}\) for \(l = 1, 2, \ldots, s-1\) and \(s = |S|\). Then

\[
\text{Prob}\{I_j = S\} = \text{Prob}\{C_{i_{l-1}} < A_{i_l}\} \\
\quad \cdot \prod_{l=1}^{s-1} \text{Prob} \left\{ \left( \bigcap_{i_l \leq k < i_{l+1} - 1} P_{i_l k} > A_{k+1} - A_{i_l} \right) \cap \left( P_{i_{l+1} k} = A_{i_{l+1}} - A_{i_l} \right) \right\} \\
\quad \cdot \text{Prob} \left\{ \bigcap_{i_s \leq k < j} P_{i_s k} > A_{k+1} - A_{i_s} \right\}
\]

where \(\text{Prob}\{C_0 < A_1\} = 1\).

**Proof.** As in Examples A.3.2 and A.3.3, we consider blocks of jobs between successive elements of \(S\). We start our derivation from the largest element of \(S\) (i.e., \(i_s\)) and use Lemma A.3.1 (viz, if \(i \in I_j\) then job \(i\) starts on time, i.e., it starts at \(A_i\)).

The largest element of \(I_j\) is \(i_s\) hence \(i_s\) is the last job, before \(j+1\), which starts on time (indeed, otherwise there must exist a job \(i_s < k \leq j\) either with \(C_{k-1} < A_k\) implying that \(A_s + P_{s j} < A_k + P_{k j}\) and \(i_s \not\in I_j\), or with \(C_{k-1} = A_k\) implying that \(A_s + P_{s j} = A_k + P_{k j}\) and \(i_s < k \in I_j\)). Therefore all jobs after \(i_s\) must be late, i.e., event \(\bigcap_{i_s \leq k < j} A_{i_s} + P_{i_s k} > A_{k+1}\) holds.

Next, we consider \(i_l\). Since \(i_l, i_{l+1} \in I_j\) both of them are on time and we must have \(A_{i_l} + P_{i_l, i_{l+1} - 1} = A_{i_{l+1}}\) (indeed, if \(A_{i_l} + P_{i_l, i_{l+1} - 1} < A_{i_{l+1}}\) then \(i_l \not\in I_j\), if \(A_{i_l} + P_{i_l, i_{l+1} - 1} > A_{i_{l+1}}\) then \(i_l \not\in I_j\)). Furthermore, all jobs between \(i_l\) and \(i_{l+1}\) must be late, i.e., \(A_{i_l} + P_{i_l k} > A_{k+1}\) for \(i_l \leq k < i_{l+1} - 1\) (if \(A_{i_l} + P_{i_l k} = A_{k+1}\) then \(k+1 \in I_j\), if \(A_{i_l} + P_{i_l k} < A_{k+1}\) then \(i_l \not\in I_j\)). Therefore, the following event must hold:

\[
\left\{ \prod_{l=1}^{s-1} \left( \bigcap_{i_l \leq k < i_{l+1} - 1} A_{i_l} + P_{i_l k} > A_{k+1} \right) \cap \left( A_{i_l} + P_{i_l i_{l+1} - 1} = A_{i_{l+1}} \right) \right\} \cap \left\{ \bigcap_{i_s \leq k < j} A_{i_s} + P_{i_s k} > A_{k+1} \right\}
\]

Finally, we consider the jobs before \(i_1\). Since \(i_1 \in I_j\), \(i_1\) starts on time, i.e., \(S_{i_1} = A_{i_1}\). Furthermore, completion time of job \(i_1 - 1\) must be strictly less than \(A_{i_1}\), i.e., \(C_{i_1 - 1} < A_{i_1}\) (indeed, if \(C_{i_1 - 1} = A_{i_1}\) then there exist at least one job \(k\) such that \(k \leq i_1 - 1\) and \(k \in I_j\), if \(C_{i_1 - 1} > A_{i_1}\) then \(i_1 \not\in I_j\)). Therefore the following event must hold:

\[
\left\{ \left\{ \left. C_{i_1 - 1} < A_{i_1} \right| \text{part 1} \right\} \cap \left( \prod_{l=1}^{s-1} \left( \bigcap_{i_l \leq k < i_{l+1} - 1} A_{i_l} + P_{i_l k} > A_{k+1} \right) \cap \left( A_{i_l} + P_{i_l i_{l+1} - 1} = A_{i_{l+1}} \right) \right\} \right\} \cap \left\{ \bigcap_{i_s \leq k < j} A_{i_s} + P_{i_s k} > A_{k+1} \right\} \right\}
\]
This proves that \( \{I_j = S\} \subseteq \{\text{part } 1 \cap \text{part } 2 \cap \text{part } 3\} \).

Conversely, assume that outcome \( p \) is such that the event \( \{\text{part } 1 \cap \text{part } 2 \cap \text{part } 3\} \) holds.

\( \text{part } 1 \text{ implies } m \notin I_j \) for \( 1 \leq m < i_1 \) since \( A_m + P_{m,i_1-1} \leq C_{i_1-1} < A_{i_1} \) (i.e., job \( m \) cannot be on the critical path of \( I_j \)).

\( \text{part } 3 \text{ implies } m \notin I_j \) for \( i_s < m \leq j \) since \( A_{i_s} + P_{i_s,m} > A_{m+1} \) (i.e., job \( m \) cannot be on the critical path of \( I_j \)), and \( A_{i_s} + P_{i_s,j-1} > A_j \) implies \( i_s \in I_j \).

\( A_{i_l} + P_{i_l,i_{l+1}-1} = A_{i_{l+1}} \) (\( 1 \leq l \leq s - 1 \)) of \( \text{part } 2 \) implies that either \( i_l, i_{l+1} \in I_j \) or \( i_l, i_{l+1} \notin I_j \) (i.e., they are either on the critical path of \( I_j \) or not). But as shown above \( i_s \in I_j \), therefore \( i_1, i_2, ..., i_s \in I_j \). The only remaining thing to show is that \( m \notin I_j \) for \( i_l < m < i_{l+1} \) and \( 1 \leq l \leq s - 1 \). But this follows from \( \big( \cap_{i_1 \leq k < i_{l+1} - 1} A_{i_l} + P_{i_l,k} > A_{k+1} \big) \) of \( \text{part } 2 \) and the fact that \( i_l \in I_j \).

Therefore \( \{I_j = S\} = \{\text{part } 1 \cap \text{part } 2 \cap \text{part } 3\} \) and

\[
\begin{align*}
\text{Prob}\{I_j = S\} &= \text{Prob}\left\{ \left. \bigcap_{i=1}^{s-1} \left( \bigcap_{i_l \leq k < i_{l+1} - 1} A_{i_l} + P_{i_l,k} > A_{k+1} \right) \cap \left( A_{i_l} + P_{i_l,i_{l+1} - 1} = A_{i_{l+1}} \right) \right\} \right. \\
&\quad \cap \left\{ \bigcap_{i_s \leq k < j} A_{i_s} + P_{i_s,k} > A_{k+1} \right\} \right. \\
&\quad \text{part } 2 \\
&\quad \text{part } 3 \\
&\quad (A.7)
\end{align*}
\]

We now carefully look at \( \text{part } 1 \), \( \text{part } 2 \) and \( \text{part } 3 \). In terms of processing times (i.e., only \( p_i \)'s), \( \text{part } 1 \) is a function of \( p_k \) for \( 1 \leq k < i_1 \), \( \text{part } 2 \) is a function of \( p_k \) for \( i_1 \leq k < i_s \), and \( \text{part } 3 \) is a function of \( p_k \) for \( i_s \leq k < i_j \). Therefore \( \text{part } 1 \), \( \text{part } 2 \) and \( \text{part } 3 \) are independent of each other. Furthermore, we may break \( \text{part } 2 \) into \( s - 1 \) independent smaller parts (i.e., one for each \( l \) in \( 1 \leq l \leq s - 1 \)) since each term

\[
\left\{ \left( \bigcap_{i_l \leq k < i_{l+1} - 1} A_{i_l} + P_{i_l,k} > A_{k+1} \right) \cap \left( A_{i_l} + P_{i_l,i_{l+1} - 1} = A_{i_{l+1}} \right) \right\}
\]

is (in terms of processing times) a function of only \( p_k \) for \( i_l \leq k \leq i_{l+1} - 1 \). Then by
independence we obtain

\[
\Pr \{ I_j = S \} = \Pr \{ C_{i_1 - 1} < A_{i_1} \} \cdot \prod_{l=1}^{s-1} \Pr \left( \left( \bigcap_{i_l \leq k < i_{l+1} - 1} A_{i_k} + P_{i_k} > A_{k+1} \right) \cap \left( A_{i_k} + P_{i_k, i_{l+1} - 1} = A_{i_{l+1}} \right) \right) \cdot \Pr \left( \bigcap_{i_s \leq k < j} A_{i_k} + P_{i_k} > A_{k+1} \right)
\]

This completes the proof.

\[\square\]

**Remark A.3.5.** Note that the probability of a set which is an intersection of sets over an empty set of indices is 1. Therefore, if \( S = \{1\} \) then by Lemma A.3.4,

\[
\Pr \{ I_j = \{1\} \} = \Pr \left\{ \bigcap_{i_s \leq k < j} P_{i_k} > A_{k+1} - A_{i_k} \right\}
\]

and if \( S = \{j\} \) then by Lemma A.3.4,

\[
\Pr \{ I_j = \{j\} \} = \Pr \{ C_{j-1} < A_j \}.
\]

Probability \( \Pr \{ C_{i_1 - 1} < A_{i_1} \} \) can be directly obtained from the distribution of completion times as we compute the entire distribution for completion times in the expected cost computations.

The remaining probabilities can be computed efficiently in a recursive manner. Probabilities that we need (for current and later purposes) are in the form of

\[
\Pr^{\xi, \eta}(l, m) = \Pr \left\{ \left( \bigcap_{l \leq v < m-1} P_{l,v} \xi A_{v+1} - A_l \right) \cap \left( P_{l,m-1} \eta A_m - A_l \right) \right\}
\]  

(A.8)

for all \( 1 \leq l < m \leq n + 1 \) where \( \xi \in \{>, \geq\} \) and \( \eta \in \{>, =, <\} \).

We now give a recursive algorithm \( RCA \) (Recursive Computation Algorithm) to compute \( \Pr^{\xi, \eta}(l, m) \). For \( 1 \leq i < k < m \) and \( (k - i + 1)\overline{p}_{\text{max}} \geq t \) \( \xi (A_k - A_i) \) define

\[
J_{i,k}^{\xi}(t) = \Pr \{ P_{ik} = t \cap \left( \bigcap_{l \leq v \leq k-1} P_{l,v} \xi A_{v+1} - A_l \right) \}
\]

(A.9)

\[
J_{i,i}^{\xi}(t) = \Pr \{ P_{ii} = t \} = \Pr \{ p_i = t \}.
\]

(A.10)

If we can compute these probabilities then

\[
\Pr^{\xi, \eta}(l, m) = \sum_t \{ J_{l,m-1}^{\xi}(t) : t \eta A_m - A_l \}
\]
Recall that $\bar{p}_{\text{max}}$ is the maximum possible value for any job distribution. Initial conditions for the recursion is given by Eq(A.10). We now provide the recursion for computing $J_{l,k}^\xi(t)$'s.

$$
J_{l,k}^\xi(t) = \text{Prob}\left\{ (P_{l,k} = t) \cap \left( \bigcap_{i \leq v \leq k-1} P_{t,v} \xi A_{v+1} - A_i \right) \right\}
$$  \hspace{1cm} (A.11)

$$
= \sum_{u \in [\bar{p}_{\text{max}}] \cap [t]} \text{Prob}\left\{ (p_k = u) \cap (P_{l,k-1} = (t-u)) \cap \left( \bigcap_{i \leq v \leq k-1} P_{t,v} \xi A_{v+1} - A_i \right) \right\}
$$  \hspace{1cm} (A.12)

$$
= \sum_{u \in [\bar{p}_{\text{max}}] \cap [t]} \text{Prob}\{p_k = u\} \text{Prob}\left\{ (P_{l,k-1} = (t-u)) \cap \left( \bigcap_{i \leq v \leq k-2} P_{t,v} \xi A_{v+1} - A_i \right) \right\}
$$  \hspace{1cm} (A.13)

$$
= \sum_{(t-u) \in [\bar{p}_{\text{max}}] \cap [t]} \text{Prob}\{p_k = u\} \text{Prob}\left\{ (P_{t,k-1} = (t-u)) \cap \left( \bigcap_{i \leq v \leq k-2} P_{t,v} \xi A_{v+1} - A_i \right) \right\}
$$  \hspace{1cm} (A.14)

$$
= \sum_{u} \left\{ \text{Prob}\{p_k = u\} \ J_{l,k-1}^\xi(t-u) : (t-u) \xi A_k - A_i \right. \left. \text{ and } u \in [\bar{p}_{\text{max}}] \cap [t] \right\}
$$  \hspace{1cm} (A.15)

Eq(A.11) is just the definition of $J_{l,k}^\xi(t)$ given by Eq(A.9). Then we write $P_{l,k}$ as the convolution of $p_k$ and $P_{l,k-1}$ in Eq(A.12). By recognizing the fact that $p_k$ and $P_{l,k-1}$ are independent, and $p_k$ does not appear in the remaining terms, we obtain Eq(A.13). Next, in Eq(A.14) we place the condition $P_{l,k-1} \xi A_k - A_i$ in the sum to obtain $J_{l,k-1}^\xi(t)$ and therefore the recursion as shown in Eq(A.15).

We now take a close look to the $\text{Prob}^{\eta}(l,m)$ computation.

$$
J_{l,t+1}^\xi(t) = \text{Prob}\{p_l \xi A_{l+1} - A_l \text{ and } P_{l,t+1} = t\}
$$

$$
J_{l,t+2}^\xi(t) = \text{Prob}\{p_l \xi A_{l+1} - A_l \text{ and } P_{l,t+1} \xi A_{l+2} - A_l \text{ and } P_{l,t+2} = t\}
$$

$$
= \ldots
$$

$$
J_{l,m-1}^\xi(t) = \text{Prob}\{p_l \xi A_{l+1} - A_l \text{ and } \ldots \text{ and } P_{l,m-2} \xi A_{m-1} - A_l \text{ and } P_{l,m-1} = t\}
$$

$$
\text{Prob}^{\eta}(l,m) = \text{Prob}\{p_l \xi A_{l+1} - A_l \text{ and } \ldots \text{ and } P_{l,m-2} \xi A_{m-1} - A_l \text{ and } P_{l,m-1} \eta A_m - A_l\}
$$

$$
= \sum_{\eta A_m - A_l} J_{l,m-1}^\xi(t)
$$  \hspace{1cm} (A.16)

We can now give the formula for $\text{Prob}\{I_j = S\}$. Write $S \in \mathcal{P}^*(\{j\})$ as an ordered list: $S = \{i_1, \ldots, i_s\}$ where $i_l < i_{l+1}$ for $l = 1, 2, \ldots, s-1$ and $s = |S|$. Then by Lemma A.3.4 and Eq(A.16) we get:

$$
\text{Prob}\{I_j = S\} = \text{Prob}\{C_{i_1-1} < A_{i_1}\} \left( \prod_{l=1}^{s-1} \text{Prob}^{\eta_{(l,i_{l+1})}}(i_l, i_{l+1}) \right) \text{Prob}^{\eta_j}(i_s, j)
$$  \hspace{1cm} (A.17)

To compute $\text{Prob}\{I_j^\eta = S\}$ for $\eta \in \{>, =, <\}$, we need a Corollary.
Corollary A.3.6.

\[ \text{Prob}\{I^\eta_j = S\} = \text{Prob}\{C_{i_1-1} < A_{i_1}\} \left( \prod_{i=1}^{s-1} \text{Prob}^{>,-}(i_i, i_{i+1}) \right) \text{Prob}^{>,-}(i_s, j + 1). \]

**Proof.** By Eq(A.7) of Lemma A.3.4 we obtain

\[ \text{Prob}\{I^\eta_j = S\} = \text{Prob}\{C_{i_1-1} < A_{i_1}\} \]

\[ \times \prod_{i=1}^{s-1} \left\{ \left( \bigcap_{i_j \leq k < i_{i+1}-1} A_{i_j} + P_{i_{i_k}} > A_{k+1} \right) \cap \left( A_{i_j} + P_{i_{i_{i_k+1}} - 1} = A_{i_{i+1}} \right) \right\} \]

\[ \times \left\{ \left( \bigcap_{i_s \leq k < j} A_{i_s} + P_{i_{s_k}} > A_{k+1} \right) \cap \left( A_{i_s} + P_{i_{s_j} \eta} A_{j+1} \right) \right\}. \]

As in Lemma A.3.4, by independence we obtain

\[ \text{Prob}\{I^\eta_j = S\} = \text{Prob}\{C_{i_1-1} < A_{i_1}\} \]

\[ \times \prod_{i=1}^{s-1} \text{Prob}^{>,-}(i_i, i_{i+1}) \text{Prob}^{>,-}(i_s, j + 1). \quad (A.18) \]

This completes the proof. \(\square\)

### A.3.2 Complexity of Subgradient Computations

We first look at the complexity of the required probability computations. We see that the complexity of \(\text{Prob}\{I_j = S\}, \text{Prob}\{I^>_j = S\} \) and \(\text{Prob}\{I^=_j = S\} \) are the same.

For \(\text{Prob}\{I_j = S\}\), we need to compute and multiply probabilities given in Eq(A.17), namely \(\text{Prob}\{C_{i_1-1} < A_{i_1}\}, \prod_{i=1}^{s-1} \text{Prob}^{>,-}(i_i, i_{i+1})\) and \(\text{Prob}^{>,-}(i_s, j)\). Since \(\prod_{i=1}^{s-1} \text{Prob}^{>,-}(i_i, i_{i+1})\) is the product of \(|S| - 1\) \(\text{Prob}^{>,-}(\ldots)\) terms, the complexity of \(\text{Prob}\{I_j = S\}\) is the same as the complexity of \(\text{Prob}\{C_{i_1-1} < A_{i_1}\}\)(\(|S|\text{Prob}^{>,-}(i, j))\).

We start with \(\text{Prob}\{C_{i_1-1} < A_{i_1}\}\). The complexity of this probability computation is \(O((n\overline{\eta}_{\text{max}})^2)\) when \(\mathbf{A}\) is integer and \(O(n(n\overline{\eta}_{\text{max}})^2)\) otherwise. These directly follow from
expected cost computations. Recall that \( h = n \bar{p}_{\max} \), so we can represent the worst case as \( O(nh^2) \).

Next, we look at the complexity of \( \text{Prob}^{\succ \succ}(i, j) \). We need to compute \( \text{Prob}^{\succ \succ}(i, j) \) for every \( i \) (\( i < j \)), so we have a factor of \( n \). Furthermore, in the worst case we need to compute \( J^2_{1:n} \). For this convolution, we may have as many as \( n \bar{p}_{\max} \) values and \( O(\bar{p}_{\max}) \) operations for each value. Finally, \( |S| \) may be at most \( n \) so all together the complexity of \( |S| \text{Prob}^{\succ \succ}(i, j) \) becomes \( O(nn \bar{p}_{\max} \bar{p}_{\max}) \). Since \( h = n \bar{p}_{\max} \) it becomes \( O(nh^2) \).

Hence for a given \( S \in \mathcal{P}^*([j]) \), the complexity of \( \text{Prob}\{I_j = S\} \) becomes \( O(nh^2) \) for a particular \( j \), and \( O(n^2h^2) \) for all jobs. However, due to \( S \in \mathcal{P}^*([j]) \) (in both \( \partial F(A) \) and \( \text{Prob}\{I_j = S\} \)), we have an additional factor of \( O(2^n) \) in the complexity of subgradient computations.

### A.3.3 Obtaining a Subgradient Fast (in Polynomial Time)

The preceding characterizations of \( \partial F(A) \) may not be efficient for convex analysis methods to find an optimal appointment schedule. This is due to the \( O(2^n) \) factor in obtaining \( \partial F(A) \). Instead of fully characterizing \( \partial F(A) \), we may obtain a subgradient \( g \in \partial F(A) \) quickly. We will do so by not computing the convex hulls (in \( \partial L_j(A), \partial T_j(A) \) and \( \partial M_j(A) \)) but by choosing a particular (smallest (or largest) index) element for that convex hull (i.e., set a particular \( X \) variable to 1 and all others to zero in every convex combination).

Recall that the \( O(2^n) \) factor comes from the fact that \( S \in \mathcal{P}^*([j]) \) where \( [j] = \{1, 2, \ldots, j\} \). We are now after just one subgradient, but not the whole subdifferential, so instead of considering all possible (non-empty) subsets of \( [j] \) (i.e., \( \mathcal{P}^*([j]) \)) for \( S \) and computing the corresponding convex hull, we just choose the vector corresponding to the smallest (or the largest) element of each \( S \). In other words, when we choose the smallest element of each \( S \), we eliminate variables \( X_{1:j}^{(i)}(S) \) by constraining \( X_{1:j}^{(i)}(S) = 1 \) if \( i = \min S = \min\{t : t \in S\} \) and \( X_{1:j}^{(i)}(S) = 0 \) otherwise. Similarly, when we choose the largest element of each \( S \), we eliminate variables \( X_{1:j}^{(i)}(S) \) by constraining \( X_{1:j}^{(i)}(S) = 1 \) if \( i = \max S = \max\{t : t \in S\} \) and \( X_{1:j}^{(i)}(S) = 0 \) otherwise. We illustrate this idea with an example.

**Example A.3.7.** Let \( F(A) = L_3(A) \) then \( \partial F(A) = \partial L_3(A) \). We will find two subgradi-
ents, $g'(A), g''(A) \in \partial L_3(A)$. Recall that $[3] = \{1, 2, 3\}$ and by Eq(3.10) of Chapter 3

$$
\partial L_3(A) = \left\{ \sum_{S \in P^*([3])} \text{Prob}\{ I_3 = S \} \sum_{k \in S} (1_k - 1_4) X^L_{k3}(S) : \sum_{k \in S} X^L_{k3}(S) = 1 \forall S \in P^*([3]) \right\}
$$

then we obtain $g'(A)$ and $g''(A)$ as

$$
g'(A) = \text{Prob}\{1 \in I_3\}(1_1 - 1_4) + \text{Prob}\{2 \in I_3, 1 \notin I_3\}(1_2 - 1_4)
+ \text{Prob}\{3 \in I_3, 2 \notin I_3, 1 \notin I_3\}(1_3 - 1_4)
= \sum_{i=1}^{3} \text{Prob}\{i = \min I_3\}(1_i - 1_4) \text{ and }
$$

$$
g''(A) = \text{Prob}\{3 \in I_3\}(1_3 - 1_4) + \text{Prob}\{2 \in I_3, 3 \notin I_3\}(1_2 - 1_4)
+ \text{Prob}\{1 \in I_3, 2 \notin I_3, 3 \notin I_3\}(1_1 - 1_4)
= \sum_{i=1}^{3} \text{Prob}\{i = \max I_3\}(1_i - 1_4).
$$

The nice thing about $g'(A)$ and $g''(A)$ is that the probabilities appearing in the equations above may be computed efficiently by RCA. Recall that $I_3 = \arg \max_{k \leq 3} \{A_k + P_k3\}$. For example,

$$
\text{Prob}\{1 \in I_3\} = \text{Prob}\{A_1 + p_1 \geq A_2 \text{ and } A_1 + p_1 + p_2 \geq A_3\}. \quad (A.19)
$$

$$
\text{Prob}\{2 \in I_3 \text{ and } 1 \notin I_3\} = \text{Prob}\{A_2 + p_2 \geq A_3 \text{ and } A_1 + p_1 < A_2\} \quad (A.20)
= \text{Prob}\{A_2 + p_2 \geq A_3\} \text{Prob}\{A_1 + p_1 < A_2\} \quad (A.21)
= \text{Prob}\{A_2 + p_2 \geq A_3\} \text{Prob}\{C_1 < A_2\}.
$$

$$
\text{Prob}\{3 \in I_3 \text{ and } 2 \notin I_3 \text{ and } 1 \notin I_3\} = \text{Prob}\{A_2 + p_2 < A_3 \text{ and } A_1 + p_1 + p_2 < A_3\}
= \text{Prob}\{C_2 < A_3\}. \quad (A.22)
$$

In this paragraph’s arguments we use the Critical Path Lemma 2.4.1 of Chapter 2 for $C_3$. Eq(A.19) follows from the fact that $1 \in I_3$ if and only if $A_1$ is on the critical path. In Eq(A.20), $\{2 \in I_3 \text{ and } 1 \notin I_3\}$ implies that $A_2$ is on the critical path, but $A_1$ is not on
the critical path, and this can only happen if and only if \( A_2 + p_2 \geq A_3 \) and \( A_1 + p_1 < A_2 \). Eq(A.21) is just the result of \( p_1 \) and \( p_2 \)'s independence. \( \{3 \in I_3 \text{ and } 2 \notin I_3 \text{ and } 1 \notin I_3 \} \) if and only if \( A_3 \) is on the critical path, and \( A_2 \) and \( A_1 \) are not on the critical path. This can only happen if and only if \( C_2 < A_3 \), and Eq(A.22) gives this result.

In general, for \( \forall j \in [n + 1] \) and \( i \leq j \) we need to compute probabilities

\[
\text{Prob}\{i = \min I_j\} = \text{Prob}\{1 \notin I_j \text{ and } 2 \notin I_j \text{ and } \ldots \text{ and } i - 1 \notin I_j \text{ and } i \in I_j\},
\]
\[
\text{Prob}\{i = \min I_j^\geq\} = \text{Prob}\{1 \notin I_j^\geq \text{ and } 2 \notin I_j^\geq \text{ and } \ldots \text{ and } i - 1 \notin I_j^\geq \text{ and } i \in I_j^\geq\}
\]

for \( g'(A) \), and for \( g''(A) \) we need probabilities

\[
\text{Prob}\{i = \max I_j\} = \text{Prob}\{i \in I_j \text{ and } i + 1 \notin I_j \text{ and } \ldots \text{ and } j - 1 \notin I_j \text{ and } j \notin I_j\},
\]
\[
\text{Prob}\{i = \max I_j^\geq\} = \text{Prob}\{i \in I_j^\geq \text{ and } i + 1 \notin I_j^\geq \text{ and } \ldots \text{ and } j - 1 \notin I_j^\geq \text{ and } j \notin I_j^\geq\}.
\]

Recall that by Eq(A.5) we have \( \text{Prob}\{I_j^\geq = S\} = \text{Prob}\{I_j = S \text{ and } A_{i_s} + P_{i_s,j} > A_{j+1}\} \)
where \( i_s = \max\{i : i \in S\} \). Therefore,

\[
\text{Prob}\{i = \max I_j^\geq\} = \text{Prob}\{i \in I_j^\geq \text{ and } i + 1 \notin I_j^\geq \text{ and } \ldots \text{ and } j \notin I_j^\geq\}
\]
\[
= \text{Prob}\{i \in I_j \text{ and } i + 1 \notin I_j \text{ and } \ldots \text{ and } j \notin I_j \text{ and } A_i + P_{ij} > A_{j+1}\}\) (A.23)

Let \( i_1 = \min\{i : i \in S\} \). Since \( A_{i_1} + P_{i_1,j} = A_{i_1} + P_{i_1,j} \) we can rewrite \( \text{Prob}\{I_j^\geq = S\} \) as

\[
\text{Prob}\{I_j^\geq = S\} = \text{Prob}\{I_j = S \text{ and } A_{i_1} + P_{i_1,j} > A_{j+1}\}
\]
\[
= \text{Prob}\{I_j = S \text{ and } A_{i_1} + P_{i_1,j} > A_{j+1}\}
\]

therefore,

\[
\text{Prob}\{i = \min I_j^\geq\} = \text{Prob}\{1 \notin I_j^\geq \text{ and } \ldots \text{ and } i - 1 \notin I_j^\geq \text{ and } i \in I_j^\geq\}
\]
\[
= \text{Prob}\{1 \notin I_j \text{ and } \ldots \text{ and } i - 1 \notin I_j \text{ and } i \in I_j \text{ and } A_i + P_{ij} > A_{j+1}\}\) (A.24)

We next compute \( \text{Prob}\{i = \min I_j\}, \text{Prob}\{i = \max I_j\}, \text{Prob}\{i = \min I_j^\geq\} \) and
\( \text{Prob}\{i = \max I_j^\geq\} \). We start with the computation of \( \text{Prob}\{i = \min I_j\}\).

**Lemma A.3.8.**

\[
\text{Prob}\{i = \min I_j\} = \text{Prob}\{C_{i-1} < A_i\} \text{Prob}\{A_i + P_{it} \geq A_{t+1} \forall t = i, i + 1, \ldots, j - 1\}
\]
\[
= \text{Prob}\{C_{i-1} < A_i\} \text{Prob}^{\geq\geq}(i,j)
\]
\[
= \text{Prob}\{C_{i-1} < A_i\} \sum_{m \geq A_j - A_i} J_{i,j-1}(m).
\]

174
\textbf{Proof.} By definition \( \{i = \min I_j\} = \{1 \not\in I_j \text{ and } \ldots \text{ and } i - 1 \not\in I_j \text{ and } i \in I_j\} \), i.e., this is the event that \( A_1, A_2, \ldots, A_{i-1} \) are not on the critical path of \( C_j \) but \( A_i \) is. Then,

\[
\{i = \min I_j\} = \{1 \not\in I_j \text{ and } \ldots \text{ and } i - 1 \not\in I_j \text{ and } i \in I_j\} \\
\iff \{ \max_{k \leq i-1} A_k + P_{k,i-1} < A_i \text{ and } A_i + P_{it} \geq A_{t+1} \forall t = i, i + 1, \ldots, j - 1\} \\
\iff \{C_{i-1} < A_i \text{ and } A_i + P_{it} \geq A_{t+1} \forall t = i, i + 1, \ldots, j - 1\}.
\]

Now, since \( C_{i-1} \) is a function of only \( p_1, p_2, \ldots, p_{i-1} \) (in terms of processing times) and \( P_{it} \) is the sum of \( p_i + p_{i+1} + \ldots + p_t \) (\( i \leq t \leq j - 1 \)), events \( \{C_{i-1} < A_i\} \) and \( \{A_i + P_{it} \geq A_{t+1} \forall t = i, i + 1, \ldots, j - 1\} \) are independent. Therefore,

\[
\Prob\{i = \min I_j\} = \Prob\{C_{i-1} < A_i\} \Prob\{A_i + P_{it} \geq A_{t+1} \forall t = i, i + 1, \ldots, j - 1\}
\]

but by Eq(A.8) and Eq(A.16) we have,

\[
\Prob\{A_i + P_{it} \geq A_{t+1} \forall t = i, i + 1, \ldots, j - 1\} = \Prob^{\geq\geq}(i, j) = \sum_{m \geq A_j - A_i} J^\geq_{i,j-1}(m).
\]

Therefore the result follows. \hfill \Box

Similarly to Lemma A.3.8, we have another Lemma for \( \Prob\{i = \max I_j\} \).

\textbf{Lemma A.3.9.}

\[
\Prob\{i = \max I_j\} = \Prob\{C_{i-1} \leq A_i\} \Prob\{A_i + P_{it} > A_{t+1} \forall t = i, i + 1, \ldots, j - 1\} \\
= \Prob\{C_{i-1} \leq A_i\} \Prob^{>\geq}(i, j) \\
= \Prob\{C_{i-1} \leq A_i\} \sum_{m > A_j - A_i} J^{>\geq}_{i,j-1}(m).
\]

\textbf{Proof.} By definition \( \{i = \max I_j\} = \{i \in I_j \text{ and } i + 1 \not\in I_j \text{ and } \ldots \text{ and } j \not\in I_j\} \), i.e., this is the event that \( A_{i+1}, A_{i+2}, \ldots, A_j \) are not on the critical path of \( C_j \) but \( A_i \) is. Then,

\[
\{i = \max I_j\} = \{i \in I_j \text{ and } i + 1 \not\in I_j \text{ and } \ldots \text{ and } j - 1 \not\in I_j \text{ and } j \not\in I_j\} \\
\iff \{ \max_{k \leq i-1} A_k + P_{k,i-1} \leq A_i \text{ and } A_i + P_{it} > A_{t+1} \forall t = i, i + 1, \ldots, j - 1\} \\
\iff \{C_{i-1} \leq A_i \text{ and } A_i + P_{it} > A_{t+1} \forall t = i, i + 1, \ldots, j - 1\}.
\]
Now, since \( C_{i-1} \) is a function of only \( p_1, p_2, ..., p_{i-1} \) (in terms of processing times) and \( P_{it} \) is the sum of \( p_i + p_{i+1} + ... + p_t \) \( (i \leq t \leq j - 1) \), events \( \{ C_{i-1} \leq A_i \} \) and \( \{ A_i + P_{it} > A_{t+1} \} \) \( i, i+1, ..., j-1 \) are independent. Therefore,

\[
\text{Prob}\{i = \text{max } I_j\} = \text{Prob}\{C_{i-1} \leq A_i\} \cdot \text{Prob}\{A_i + P_{it} > A_{t+1} \forall t = i, i+1, ..., j-1\}
\]

but by Eq(A.8) and Eq(A.16) we have,

\[
\text{Prob}\{A_i + P_{it} > A_{t+1} \forall t = i, i+1, ..., j-1\} = \text{Prob}^{>\gamma}(i, j) = \sum_{m > A_j - A_i} J_{i,j-1}^\gamma(m).
\]

Therefore the result follows. \( \square \)

Then by Lemmata A.3.8 and A.3.9, and Eq(A.24) and Eq(A.23) it follows that

\[
\text{Prob}\{i = \min I_j\} = \text{Prob}\{C_{i-1} < A_i\} \cdot \text{Prob}^{>\gamma}(i, j + 1) = \sum_{m > A_j - A_i} J_{i,j}^\gamma(m) \text{ and }
\]

\[
\text{Prob}\{i = \max I_j\} = \text{Prob}\{C_{i-1} \leq A_i\} \cdot \text{Prob}^{>\gamma}(i, j + 1) = \sum_{m > A_j - A_i} J_{i,j}^\gamma(m).
\]

As mentioned before, completion time distributions are already available to us from expected cost computations and \( J_{\gamma}(\cdot) \) is computed efficiently by the recursive algorithm RCA. Therefore we can compute all the required probabilities for finding \( g' \) and \( g'' \).

We now go back to computation of \( g'(A) \) and \( g''(A) \). We compute \( g'(A) \) first. We will find \( g'(A) \) by computing contributions \( g^L_j(A) \), \( g^T_j(A) \) and \( g^M_j(A) \) of \( \partial L_j(A) \), \( \partial T_j(A) \) and \( \partial M_j(A) \) to \( g'(A) \in \partial F(A) \) respectively, and obtain \( g'(A) \) by Rule 1 (Eq(3.3) of Chapter 3) as

\[
g'(A) = \sum_{j=1}^{n}(\alpha_j g_j^L(A) + \beta_j g_j^T(A) + \gamma_j g_j^M(A)). \tag{A.25}
\]

We start with \( g_j^L(A) \), contribution of \( L_j(A) \) to \( g'(A) \). Recall that by Eq(3.10) of Chapter 3,

\[
\partial L_j(A) = \left\{ \sum_{S \in P^*([j])} \text{Prob}\{I_j = S\} \sum_{k \in S} (1_k - 1_{j+1}) X^L_{kj}(S) : \sum_{k \in S} X^L_{kj}(S) = 1 \forall S \in P^*([j]) \right. \\
C \left. \sum_{k \in S} X^L_{kj}(S) \geq 0 \forall S \in P^*([j]) \forall k \in S \}
\]

176
then by choosing the smallest index for each $S$ in every convex combination we obtain:

$$g^L_j(A) = \sum_{i=1}^j \left[ \text{Prob}\{1 \not\in I_j \text{ and } 2 \not\in I_j \text{ and } ... \text{ and } i-1 \not\in I_j \text{ and } i \in I_j \} (1_i - 1_{j+1}) \right]$$

$$= \sum_{i=1}^j \left[ \text{Prob}\{i = \min I_j\} (1_i - 1_{j+1}) \right]. \quad (A.26)$$

Next, we obtain $g^T_j(A)$. $\partial T_j(A)$ is given by Eq. (3.11) of Chapter 3 as

$$\partial T_j(A) = \left\{ \sum_{S \in P^*(\{j\})} \left[ \text{Prob}\{I_j^\geq = S\} \sum_{k \in S} (1_k - 1_{j+1}) X_{k,j}^{T}(S) \right. \right.$$

$$+ \text{Prob}\{I_j^\leq = S\} \sum_{k \in S} (1_k - 1_{j+1}) X_{k,j}^{T}(S) \right] :$$

$$\sum_{k \in S} X_{k,j}^{T}(S) = 1 \quad \forall S \in P^*(\{j\})$$

$$\sum_{k \in S} X_{k,j}^{T}(S) \leq 1 \quad \forall S \in P^*(\{j\})$$

$$X_{k,j}^{T}(S), X_{k,j}^{T}(S) \geq 0 \quad \forall S \in P^*(\{j\}) \quad \forall k \in S \right).$$

We will eliminate all the terms in the second line of $\partial T_j(A)$ by assigning 0 to all $X_{k,j}^T(S)$ variables. We may do so due to the $\sum_{k \in S} X_{k,j}^{T}(S) \leq 1$ inequality. Then by choosing the smallest index for each $S$ in every convex combination for the remaining of the terms we obtain

$$g^T_j(A) = \sum_{i=1}^j \left[ \text{Prob}\{1 \not\in I_j^\geq \text{ and } 2 \not\in I_j^\geq \text{ and } ... \text{ and } i-1 \not\in I_j^\geq \text{ and } i \in I_j^\geq \} (1_i - 1_{j+1}) \right]$$

$$= \sum_{i=1}^j \left[ \text{Prob}\{i = \min I_j^\geq \} (1_i - 1_{j+1}) \right]. \quad (A.27)$$
Finally, we obtain \( g_j^M(A) \). Recall that by Eq(3.12) of Chapter 3,

\[
\partial M_j(A) = \left\{ \sum_{S \in \mathcal{P}^*(\{j\})} \left( \text{Prob}\{I_j^\geq = S\} \sum_{k \in S} (1_k - 1_{j+1})X_{kj}^M(S) \right.ight.
\]
\[
+ \text{Prob}\{I_j^\geq = S\} \sum_{k \in S \cup \{j + 1\}} (1_k)X_{kj}^M(S \cup \{j + 1\}) \right)\]
\[
+ (1 - \sum_{S \in \mathcal{P}^*(\{j\})} \text{Prob}\{I_j^\geq = S\})1_{j+1} : \]
\[
\sum_{k \in S} X_{kj}^M(S) = 1 \ \forall S \in \mathcal{P}^*(\{j\})
\]
\[
\sum_{k \in S \cup \{j + 1\}} X_{kj}^M(S \cup \{j + 1\}) = 1 \ \forall S \in \mathcal{P}^*(\{j\}) \ \forall k \in S
\]
\[
X_{kj}^M(S) \geq 0 \ \forall S \in \mathcal{P}^*(\{j\}) \ \forall k \in S
\]
\[
X_{kj}^M(S \cup \{j + 1\}) \geq 0 \ \forall S \in \mathcal{P}^*(\{j\}) \ \forall k \in S \cup \{j + 1\} \right\}.
\]

Here we choose the smallest index for each \( S \) in the first convex combination (i.e., with \( X_{kj}^M \) terms) and choose \( j + 1 \) (note that \( j + 1 \) is always in \( S \cup \{j + 1\} \)) for each \( S \) in the second convex combination (i.e., with \( X_{kj}^M \) terms). Then,

\[
g_j^M(A) = \sum_{i=1}^{j} \text{Prob}\{1 \not\in I_j^\geq \ \text{and} \ \ldots \ \text{and} \ i - 1 \not\in I_j^\geq \ \text{and} \ i \in I_j^\geq \}(1_i - 1_{j+1})] + 1_{j+1}
\]
\[
= \sum_{i=1}^{j} [\text{Prob}\{i = \min I_j^\geq \}(1_i - 1_{j+1})] + 1_{j+1}. \quad (A.28)
\]

Next we obtain \( g'(A) \) by using Eq(A.25) and collecting \( g_j^L(A) \), \( g_j^T(A) \) and \( g_j^M(A) \) terms together.

\[
g'(A) = \sum_{j=1}^{n} (\alpha_j g_j^L(A) + \beta_j g_j^T(A) + \gamma_j g_j^M(A))
\]
\[
= \sum_{j=1}^{n} \left( \alpha_j \sum_{i=1}^{j} [\text{Prob}\{i = \min I_j^\geq \}(1_i - 1_{j+1})]\right.
\]
\[
+ \beta_j \sum_{i=1}^{j} [\text{Prob}\{i = \min I_j^\geq \}(1_i - 1_{j+1})] + \gamma_j \left[ \sum_{i=1}^{j} [\text{Prob}\{i = \min I_j^\geq \}(1_i - 1_{j+1})] + 1_{j+1} \right]. \quad (A.29)
\]

We can write \( g'(A) \) component by component. We derive the formula for \( g'_k(A) \), the
The $k^{th}$ component of $g'(\mathbf{A})$, directly from Eq(A.29).

$$g'_k(\mathbf{A}) = -\alpha_{k-1} \sum_{i=1}^{k-1} \text{Prob}(i = \min I_{k-1}) + \sum_{j=k}^{n} \alpha_j \text{Prob}(k = \min I_j)$$

$$-\beta_{k-1} \sum_{i=1}^{k-1} \text{Prob}(i = \min I_{k-1}^c) + \sum_{j=k}^{n} \beta_j \text{Prob}(k = \min I_j^c)$$

$$-\gamma_{k-1} \left( \sum_{i=1}^{k-1} \text{Prob}(i = \min I_{k-1}^c) - 1 \right) + \sum_{j=k}^{n} \gamma_j \text{Prob}(k = \min I_j^c)$$

$$= -\alpha_{k-1} \sum_{i=1}^{k-1} \text{Prob}(i = \min I_{k-1}) + \sum_{j=k}^{n} \alpha_j \text{Prob}(k = \min I_j) + \gamma_{k-1}$$

$$-(\beta_{k-1} + \gamma_{k-1}) \sum_{i=1}^{k-1} \text{Prob}(i = \min I_{k-1}^c) + \sum_{j=k}^{n} (\beta_j + \gamma_j) \text{Prob}(k = \min I_j^c). \quad \text{(A.30)}$$

By using Eq(A.25), Eq(A.26), Eq(A.27), Eq(A.28), Eq(A.29) and Eq(A.30) we can easily obtain $g''(\mathbf{A})$ and $g''_k(\mathbf{A})$ as below.

$$g''(\mathbf{A}) = \sum_{j=1}^{n} \left( \alpha_j \sum_{i=1}^{j} \left[ \text{Prob}(i = \max I_j)(1 - I_{j+1}^c) \right] \right.$$  

$$\left. + \beta_j \sum_{i=1}^{j} \left[ \text{Prob}(i = \max I_j^c)(1 - I_{j+1}) \right] \right.$$  

$$\left. + \gamma_j \left( \sum_{i=1}^{j} \left[ \text{Prob}(i = \max I_j^c)(1 - I_{j+1}^c) \right] + I_{j+1} \right) \right) \quad \text{.} \quad \text{(A.31)}$$

$$g''_k(\mathbf{A}) = -\alpha_{k-1} \sum_{i=1}^{k-1} \text{Prob}(i = \max I_{k-1}) + \sum_{j=k}^{n} \alpha_j \text{Prob}(k = \max I_j)$$

$$-\beta_{k-1} \sum_{i=1}^{k-1} \text{Prob}(i = \max I_{k-1}^c) + \sum_{j=k}^{n} \beta_j \text{Prob}(k = \max I_j^c)$$

$$-\gamma_{k-1} \left( \sum_{i=1}^{k-1} \text{Prob}(i = \max I_{k-1}^c) - 1 \right) + \sum_{j=k}^{n} \gamma_j \text{Prob}(k = \max I_j^c)$$

$$= -\alpha_{k-1} \sum_{i=1}^{k-1} \text{Prob}(i = \max I_{k-1}) + \sum_{j=k}^{n} \alpha_j \text{Prob}(k = \max I_j) + \gamma_{k-1}$$

$$-(\beta_{k-1} + \gamma_{k-1}) \sum_{i=1}^{k-1} \text{Prob}(i = \max I_{k-1}^c) + \sum_{j=k}^{n} (\beta_j + \gamma_j) \text{Prob}(k = \max I_j^c). \quad \text{(A.32)}$$

Once we know the probabilities $\text{Prob}(i = \max I_j)$, $\text{Prob}(i = \min I_j)$, $\text{Prob}(i = \max I_j^c)$ and $\text{Prob}(i = \min I_j^c)$ for all $j$ then we can find $g'(\mathbf{A})$ and $g''(\mathbf{A})$ in $O(n^2)$. And these probabilities can be computed in $O(nh^2)$ by the recursive algorithm RCA. Therefore we can obtain a subgradient of $F$ (in fact two) in polynomial time, i.e., $O(nh^2)$. (We have
implemented a preliminary program to compute subgradients $g'(A)$ and $g''(A)$ for any appointment vector $A$ (real or integer)).

**Remark A.3.10.** \[ \sum_{k=1}^{n+1} g'_k = \sum_{k=1}^{n+1} g''_k = \sum_{k=2}^{n+1} \gamma_{k-1} = \gamma_1 = u_1 + \alpha_1. \] This follows from equations Eq(A.29) and Eq(A.31), due to $(1_i - 1_{j+1})$ vectors all terms but $\sum_{j=1}^{n} \gamma_j 1_{j+1}$ disappear. This is useful in implementations since it provides an easy test to check subgradient computations.

Another observation is that the norms of the subgradients depend on $\alpha$. Therefore, one may want to choose $\alpha$ (among possible $\alpha$’s) such that subgradient norm is minimized.

**Remark A.3.11.** By Proposition 3.4.11 of Chapter 3 we can easily extend our results for $F_D$ and obtain a subgradient for $F_D$. We just need to find a subgradient for $\partial F(\tilde{A}, D)$, i.e., last component, $A_{j+1}$, will be set to $D$ in $F$.

### A.4 Algorithms

The objective function of the appointment scheduling problem has useful and interesting properties that allow us to minimize it efficiently. We can divide these properties into two main groups, depending on whether we work with integer or non-integer appointment vectors.

In the case of integer appointment vectors, we can limit our search of optimal appointment schedule to only integer appointment vectors without loss of optimality by Appointment Vector Integrality Theorem 2.5.10. Furthermore, under a mild condition on cost coefficients, i.e., $\alpha$-monotonicity (Definition 2.6.5), we can find an optimal appointment schedule with discrete algorithms using polynomial time and number of expected cost evaluations (Theorem 2.7.1). In the case of independent processing durations we can minimize $F$ in $O(nP_{\text{max}}^2 \log P_{\text{max}})$ time as shown in Theorem 2.7.3. Similar results hold for $F_D$ (Section 2.8). The results above use algorithms based on L-convexity and submodular set function minimization (e.g., see Section 10.3.2 of Murota [7], [6], [5], [9]). Besides these (discrete) algorithms, in [2, 3] authors propose to use minimum-norm-point algorithm [11] for submodular set function minimization to reduce the complexity of existing discrete algorithms. Computational results reported in [3] shows that the proposed algorithm (using minimum-norm-point algorithm) may perform better than existing polynomial algorithms.
On the other hand, if we work with non-integer appointment vectors then the objective is convex by Proposition 3.3.3, under a mild condition on cost coefficients ($\alpha$-monotonicity). Furthermore, as shown in this chapter, we can obtain a subgradient (in fact two subgradients) of the objective in $O(nh^2)$ time, which is the same complexity of computing the objective at a non-integer point. Therefore we can use non-smooth convex optimization algorithms (e.g., see [4], [10], [1]) to find an optimal appointment vector efficiently.

Both approaches, discrete and non-smooth convex optimization, have their advantages and disadvantages. For example, discrete methods have polynomial complexity with guaranteed optimality but they may be slow and more difficult to implement. (Although [3]'s proposed minimum-norm-point algorithm has a potential to be fast). On the other hand, non-smooth convex optimization methods may have a fast start (can take larger steps to reach near by optimum vector) and can be easier to implement however finding an optimal integer solution may be a challenge. Furthermore, finding a good solution can again be slow. It is not clear at this point which methods will work faster and which implementation will be easier in practice.

A third approach besides using only discrete or non-smooth convex optimization methods, is to combine these two approaches and develop a hybrid algorithm. The idea is to start with non-smooth methods to get close to an optimum vector quickly and when the improvement with non-smooth methods loses its steam turn to discrete methods. However, non-smooth methods work with non-integer appointment vectors whereas discrete algorithms work with integer appointment schedules. Therefore, to combine these two methods in a meaningful way we should be able to pass the current solution from one to another without worsening the objective. To achieve this we develop a rounding algorithm which takes any fractional solution (e.g., of a non-smooth optimization method) and rounds it to an integer one with the same or improved objective value (for a discrete algorithm). With this rounding algorithm we can combine both non-smooth and discrete methods and develop an hybrid algorithm for our appointment scheduling problem. Next we provide the details of the rounding algorithm.

The rounding algorithm is based on the Appointment Vector Integrality Theorem and its supporting Lemmata, see Section 2.5 for the details on these results. We use the same notation as in Section 2.5. Recall that for a given non-integer appointment vector $\mathbf{A}$ there exists a positive scalar $\Delta$ and we construct two new appointment schedules $\mathbf{A}'$ and $\mathbf{A}''$. The
important idea behind the rounding algorithm is that there exists an integer appointment schedule by the Appointment Vector Integrality Theorem 2.5.10 and the objective function changes linearly between $A'$ and $A''$, i.e., either $\min\{F(A'), F(A'')\} < F(A)$ or $F(A') = F(A'') = F(A)$. In other words, at each iteration (unless we’ve found an optimum) we strictly improve the objective and we run the algorithm until we obtain an integer $A$.

We now describe the rounding algorithm. The algorithm starts with an appointment vector $A$ and computes $F(A)$. If $A$ is integer then the algorithm stops, else it finds the $\Delta$ as given above. Then by using the $\Delta$ the algorithm constructs appointment schedule $A'$ and computes its cost, i.e., $F(A')$. If $F(A') < F(A)$ then it sets $A$ to $A'$ and goes to start, else generates $A''$ and sets $A$ to $A''$ and goes to start. In algorithm 1 we present the rounding algorithm.

**Algorithm 1** Rounding Algorithm

start with a given (non-integer) $A$

compute $F(A)$

while $A$ is not integer do

find $\Delta$

generate $A'$
compute $F(A')$

if $F(A') < F(A)$ then

$A \leftarrow A'$

$F(A) \leftarrow F(A')$

else

generate $A''$
compute $F(A'')$

$A \leftarrow A''$

$F(A) \leftarrow F(A'')$

end if

end while
A.5 Conclusion and Future Work

In this chapter, we use the subdifferential characterization of the objective function for appointment scheduling problem with independent processing durations and compute a subgradient in polynomial time for any given appointment schedule. Finding a subgradient in polynomial time is not trivial because the subdifferential characterization include exponentially many terms, and some of the probability computations are complicated. We also obtain an easily computable lower bound on the optimal objective value. Furthermore, we extend computation of the expected total cost (in polynomial time) for any (real-valued) appointment vector. These results allow us to use non-smooth convex optimization techniques to find an optimal schedule. Previously, we already showed that there exists a polynomial time algorithm to find an optimal appointment schedule however it is not clear at the moment which technique (discrete or non-smooth) will work faster in practice.

Besides the discrete convexity and and non-smooth convex optimization approaches, we also develop an hybrid method in which we combine both approaches with a special-purpose integer rounding method which takes any fractional solution and rounds it to an integer one with the same or improved objective value. We believe this hybrid approach may perform well in practice.

In the near future, we are planning to implement all these algorithms and methods and develop a computational engine for appointment scheduling problem. Besides, testing and comparing the discrete, non-smooth and hybrid algorithms in computational experiments we plan to test performance of various heuristic methods for both appointment scheduling and sequencing problem, and apply it to real-world appointment scheduling problem in healthcare. (A preliminary version of the rounding algorithm and subgradient computations have been implemented.)
A.6 Bibliography


