# Data-Driven Appointment Scheduling

by

Tugce Gurek

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering - Industrial Engineering and Operations Research

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Philip M Kaminsky, Chair
Professor Rhonda Righter
Professor Anil Aswani
Professor Haiyan Huang

Fall 2019

Data-Driven Appointment Scheduling

## Abstract

Data Driven Appointment Scheduling

by

Tugce Gurek

Doctor of Philosophy in Engineering - Industrial Engineering and Operations Research

University of California, Berkeley

Professor Philip Kaminsky, Chair

Advances in electronic medical records and healthcare databases enable researchers to easily acquire and analyze large amounts of data, and to build data-driven models to improve the system performance. Surgical departments, in particular, utilize a variety of expensive resources, so efficient appointment scheduling and sequencing decisions that minimize patient-surgeon waiting time and the surgeon-operating room idle time substantially reduce costs. We aim to improve the way that schedules are generated by incorporating both dynamically updated data sets, and the opinions of surgeons.

Our research focuses on appointment scheduling of stochastic tasks on a single server where the task durations are challenging to estimate. The task types are known prior to the appointment date but the task duration data is initially limited so that the estimates need to be continuously updated. Appointment scheduling involves both sequencing the tasks and setting the start time of those tasks. Our goal is to develop a data driven appointment scheduling algorithm for sequencing and scheduling tasks. Our research is motivated by a project we have completed with University of California, San Francisco (UCSF) on surgical scheduling where the tasks are the surgical procedures and the server is the operating room.

In Chapter 1 we introduce the appointment scheduling problem with a motivating example of surgical appointment scheduling. We run some simulations to show patient-surgeon waiting time (tardiness) and the surgeon-operating room idle time (earliness) can be reduced by changing the sequence of the procedures and the start times of the procedures. We go over the appointment scheduling literature with various objective functions. We analyze the objective of minimizing expected earliness and tardiness and bound the performance of the commonly used sequencing heuristic based on the standard deviation of procedure duration.

In Chapter 2 we focus on data-driven appointment scheduling. Without making any distributional assumptions we use the empirical distributions of the procedures while computing the objective function which is the expectation of weighted earliness and tardiness. We study the continuity and the convexity of the objective function and the conditions under which there is an integral optimal solution. We briefly go over the methods to optimize the objective function and also constrain the search space containing the minimizer. We develop sequencing heuristics tailored for this problem. Lastly, we consider data selection algorithms

when there are categorical features such as name of the surgeon or surgeon estimates about how long the next procedure might take.

To my parents Ayse Gul and Atilla Kubilay Gurek, and my family.

# Contents

# List of Figures

# List of Tables

## Acknowledgments

First and foremost, I would like to express my sincere gratitude and appreciation to my advisor, Professor Philip M. Kaminsky for his guidance, optimism and for believing in me. I really appreciate him making time to discuss our research, to go further and overcome challenges. His dedication to his work and enthusiasm about his research have inspired me all the time. I am grateful to Professor Rhonda Righter, Professor Anil Aswani, and Professor Haiyan Huang for their academic guidance and being on my qualifying exam and dissertation committee. I would like to thank my committee members for their valuable feedback during my graduate studies.

I would like to thank the IEOR the department staff, especially Keith McAleer, Anayancy Paz, Sonia Chahal, Rebecca R. Pauling, Yeri Caesar-Kaptoech, Diana Salazar and Heather Iwata. I appreciate their timely help.

I am grateful to Kamil Nar and Orhan Ocal who have given me an unconditional support and valuable advice through my PhD journey. I am incredibly proud of being in the research group with Heejung Kim, Shiman Ding, Arman Jabbari, Stewart Liu, Yang Wang and Dan Bu. I am also happy that our paths crossed with Kevin Li, Quico Spaen, Rebecca Sarto, Birce Tezel, Auyon Siddiq, Jiaying Shi, Mo Zhou, Jiung Lee, Erik Bertelli, Guang Yang, Amber Richter, Cheng Lyu, Angel Yang, Wen Gao, Jared Bauman, Carlos Deck, Brent Eldridge, Alfonso Lobos, Salar Fattahi, Mark Velednitsky, and Renyuan Xu. I am thankful to Sevi Baltaoglu for being a wonderful companion, Aydan Inak and Burak Onal for our weekly motivating phone calls, Dilara Semerci and Regis Frey for being a part of Berkeley team.

I appreciate my family's unconditional support from Turkey. I thank my parents for taking time out of their busy schedules to visit me frequently. My cousins, uncles, aunts, even my grandmother take long-haul flight just to spend time with me. I am grateful that I am part of this loving, caring family: Aysel, Cemal, Atilla, Ayse Gul, Bahadir, Faliha, Dorukan, Bengihan, Cenk, Yasemin, Selin, Alp Gurek, Aynur, Metin Mete, Idil Topcuoglu, Nezahat, Ahmet, Cagatay, Muge, Cagan Kanal, Gozde, Evren, Eren Ege, Melissa Kayakiran. Last but not least, I would like to thank my husband Mehmet Mustafa Yilmaz for making my life filled with so much happiness and love.

# Chapter 1

# Appointment Scheduling

## 1.1 Introduction

This research focuses on appointment scheduling of stochastic tasks on a single server where the task processing durations are challenging to estimate. Appointment scheduling involves both sequencing tasks, and setting estimated start times of those tasks. Tasks types are known prior to the appointment date, but task duration data is initially limited so duration estimates are continuously updated. Our goal is to develop a data-driven approach to sequencing and scheduling these tasks that also integrates expert knowledge into the scheduling decision.

Our work is motivated by a project we completed with UCSF on surgical scheduling, where the tasks are surgeries and the server is the operating room. Advances in electronic medical records and healthcare databases enable researchers to easily acquire and analyze large amounts of data, and to build data-driven models to improve the system performance. Surgical departments, in particular, utilize a variety of expensive resources, so efficient appointment scheduling and sequencing decisions that minimize patient-surgeon waiting time and the surgeon-operating room idle time substantially reduce costs. Our eventual goal is to improve the way schedules are generated by incorporating both dynamically updated data sets, and the opinions of surgeons.

In Chapter 1 we introduce the appointment scheduling problem with a motivating example of surgical appointment scheduling. We run some simulations to show patient-surgeon waiting time (tardiness) and the surgeon-operating room idle time (earliness) can be reduced by changing the sequence of the procedures and the start times of the procedures. We go over the appointment scheduling literature with various objective functions. We analyze the objective of minimizing expected earliness and tardiness and bound the performance of the commonly used sequencing heuristic based on the standard deviation of procedure duration.

In Chapter 2 we focus on data-driven appointment scheduling. Without making any distributional assumptions we use the empirical distributions of the procedures while computing the objective function which is the expectation of weighted earliness and tardiness. We study the continuity and the convexity of the objective function and the conditions under which

there is an integral optimal solution. We briefly go over the methods to optimize the objective function and also constrain the search space containing the minimizer. We develop sequencing heuristics tailored for this problem. Lastly we talk about data selection algorithms if there are categorical features such as name of the surgeon or surgeon estimates about how long the next procedure might take.

### 1.1.1 Motivation

Caring Wisely™ is a program run by University of California, San Francisco (UCSF) Center for Healthcare Value. This program is designed to fund interventions that can reduce costs, improve value and enable innovation. We collaborated with Lindsay Hampson, MD, Max Meng, MD and their team on Operating Wisely: Operating Room Teamwork in Improving and Measuring Efficiency (ORTIME), a Caring Wisely™ project.

The ORTIME project focused on increasing operating room (OR) efficiency without increasing the preoperative, intraoperative, and postoperative complication rates. OR efficiency is defined in terms of the percentage of on-time cases, patient-surgeon waiting time, surgeon-operating room idle time and non-operative OR time. Improving OR efficiency requires improving prediction of procedure times (schedules) while:

1. Increasing the percentage of on-time cases,
2. Reducing patient-surgeon waiting time, surgeon-operating room idle time, and
3. Minimizing non-operative OR time (e.g. turnover time).

Operating rooms account for a substantial amount of revenue and hospital expenses. Non-operative OR time does not generate revenue, so the goal is to minimize or eliminate it if possible. Turning over a room is a non-operative OR time which requires surgeons, anesthesiologists, nurses and other staff to work together. OR turnover time is the time between when a patient leaves an OR after a procedure and the time the next patient arrives in the OR for the next procedure. Decreasing turnover time increases OR efficiency.

Data about current operations has been collected for several years. We analyzed this data to determine the current scheduling efficiency, and used this data combined with a simulation to assess the impact of alternative sequencing rules and appointment scheduling approaches.

### 1.1.2 Analysis of Existing Data

The baseline data we obtained from UCSF Medical Center at Mt. Zion shows that 8.6% of cases are completed before the scheduled end time and only 13.5% are completed within 15 minutes of the scheduled end time. This data consists of the information about 14593 different cases (note that confidential data was encoded to hide confidential information). The data includes:

- Timeline of the procedures (Surgery date, Time patient enters and exits OR, Time procedure starts and ends, Scheduled start time and end time)
- Procedure type and code

Actual duration of the procedure (time spent in the room), which is the difference between the time patient enters the OR and exits, can be directly obtained using the baseline data. Unfortunately the data doesn't have the actual turnover time after procedure. So we approximated the turnover time as (upper bound on the actual turnover time):

$$Proc\_Duration[i] = Time\_Exit\_OR[i] - Time\_Enter\_OR[i]$$

$$Turnover[i] = \begin{cases} Time\_Enter\_OR[i+1] - Time\_Exit\_OR[i] \\ \quad \text{if} \quad Sched\_Start\_Time[i+1] = Sched\_End\_Time[i] \\ NA \\ \quad \text{otherwise} \end{cases}$$

If a procedure ends later than the scheduled start time of the next procedure, this results in patient-surgeon waiting time for the next procedure, whereas if the procedure ends before the start time of the next procedure, the surgeon and the OR stay idle. We define tardiness as patient-surgeon waiting time and earliness as surgeon-OR idle time.For each procedure we calculate the earliness and tardiness. Unfortunately average total earliness and tardiness is more than an hour.(See 'baseline' column of Figure 1.1) Also, the average approximate turnover time is around 45 minutes, so the current scheduling approach seems to be quite inefficient. It is worth mentioning that the all things being equal, the hospital administration prefers procedures to end early rather than be tardy, but almost 42% of the procedures are early whereas 58% are tardy.

**How are the schedules generated so far?** Each procedure time is estimated by taking the average of the last three procedures of the same kind done by the same surgeon, if there is enough data. By default the scheduler adds 30 minutes to the predicted procedure time to reflect the turnover time. The sequence of the procedures is randomly generated.

We conducted a simulation analysis to test if OR efficiency could be increased by changing the sequence of procedures or by changing the time scheduled for turnovers.

### 1.1.3 Simulation Study of Alternative Sequencing Rules

We ran simulations to observe the effect of changing the sequence and increasing the time scheduled for turnover. In literature the most commonly used heuristic is sequencing the procedures in increasing standard deviation of durations. It is reasonable to schedule a procedure with higher variability later to minimize its potential impact on the upcoming procedures. The simulations we run are:

**Step 1:** For each type of procedure we calculated the sample standard deviation.

**Step 2:** Without changing the room or day for each procedure we sequenced the surgeries in order of increasing standard deviation.

| | Baseline | Simulation 1 | Simulation 2 |
|---|---|---|---|
| Percentage of the operations finished early | 41.77% | 41.57% | 49.41% |
| Mean earliness | 22.25 minutes | 20.48 minutes | 26.81 minutes |
| Standard deviation of earliness | 45.66 | 42.21 | 47.68 |
| Expected earliness given an operation is early | 53.25 minutes | 49.25 minutes | 54.26 minutes |
| Percentage of the operations finished late | 57.47% | 57.61% | 49.77% |
| Mean tardiness | 41.78 minutes | 40.04 minutes | 32.32 minutes |
| Standard deviation of earliness | 63.69 | 62.59 | 57.49 |
| Expected tardiness given an operation is tardy | 72.7 minutes | 69.49 minutes | 64.93 minutes |

Figure 1.1: Results of the simulation runs

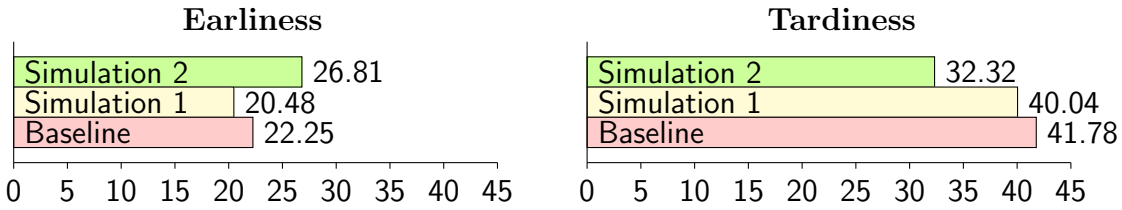



Figure 1.2: Average earliness and tardiness

**Step 3:** We generated random turnover times from an empirical distribution of the estimated turnover time data.
**Step 4:** We keep the scheduled OR and turnover times for each procedure in the baseline data. In other words we use the previous estimates of time allowances of each procedure and turnover calculated by the hospital's system.

Our new approach leads to a statistically significant smaller total tardiness and earliness compared to the actual. However, our estimate of the average turnover time is 45.23 minutes, but turnover was scheduled for 30 minutes, so we next tested an alternative step 4:
**Step 4:** The scheduled turnover time is increased to 45 minutes. The scheduled OR each procedure also is not changed.

This approach performed even better. Figures 1.1 and 1.2 show the results of our simulation runs compared to baseline data. As mentioned before, the decrease in tardiness is considered more valuable than the decrease in earliness. In other words hospital administration is content as long as the increase in earliness is less than the decrease in tardiness.

These simulation results both suggest that there is room for improvement, and motivate us to design more efficient approaches for appointment scheduling so that patient-surgeon waiting time and surgeon-operating room idle time and turnover times (non-operative OR times) can be be minimized. In other words our overall goal in this research is to develop more effective and efficient appointment scheduling procedures. In Section 2, we introduce our notation, and review the relevant literature. In Section 3, we formulate the problem, and present our preliminary results. We introduce an initial heuristic for the data-driven appointment scheduling problem, and bound the performance of that heuristic. In Section 4, we review our research agenda.

## 1.2 Literature Review

Most appointment scheduling literature focuses on minimizing expected earliness and tardiness but there are also alternative objectives studied. In this section we review appointment scheduling literature with different objectives. Our plan to explore dynamically updating estimates of task times is closely related to approaches used for the so-called data-driven newsvendor model, so we also review that literature.

### 1.2.1 Appointment Scheduling: Minimizing Expected Earliness and Tardiness

Extensive surveys on research in appointment scheduling are provided by Cayirli and Veral (2003), Gupta (2007) and Gupta and Denton (2008). Cayirli and Veral (2003) primarily focuses on outpatient scheduling and presents various problem formulations, different performance measures used while designing appointment scheduling, possible appointment scheduling designs and the analysis methodologies. Gupta and Denton (2008) gives a detailed description of three health care appointment scheduling environments: primary care specialty clinic, and elective surgeries. The appointment system in each environment is classified by the mapped arrival process (time when the appointment decision is made: as soon as the patient arrives or later), service time (random or deterministic), existence of patient provider preference and the objective (cost minimization or revenue maximization). Gupta (2007) describes three common surgical suites' operations management problems, models and some solution approaches. The first common problem is elective surgery capacity allocation: how much surgeon, service block time is needed to maximize the hospital's total contribution. The second problem, elective surgery booking control, arises from the post anesthesia care unit (PACU), intensive care unit (ICU) and bed capacity constraints and aims to allocate the patients (demand) to bend while satisfying the bed-demand. Elective surgery sequencing problem is the most relevant to our research, and involves searching for the sequence of surgeries with the minimum expected cost after finding the time allowance of each surgery in a particular sequence (i.e. scheduling the surgery start times for any sequence).

Gupta and Denton (2008) consider three types of service process design scenarios: constant, diagnosis dependent and random service times. A variety of papers considers random service times which can be diagnosis independent (identical distributed) or diagnosis dependent: Weiss (1990), Bailey (1952), Welch and Bailey (1952), Denton and Gupta (2003), Denton et al. (2007), Kaandorp and Koole (2007), Erdogan et al. (2011), Begen and Queyranne (2011), Begen et al. (2012) , Ge et al. (2013), Kong et al. (2013) and Mak et al. (2014).

Weiss (1990), Denton et al. (2007) show that the problem of scheduling only two procedures is similar to the famous 'Newsvendor Problem'. The cost of waiting time (tardy) is analogous to underage cost whereas cost of being early is analogous to the overage cost.

Denton and Gupta (2003) consider a single server system at which the sequence of the customer arrival is fixed and minimize the expected cost associated with the server idle time (earliness of each procedure), customer waiting time (tardiness of each procedure) and the session length (tardiness, overtime) over the probability distribution of job duration. They formulate a two-stage stochastic linear program (2-SLP) resulting in a convex minimization problem and adapt the standard L-shaped algorithm (Van Slyke and Wets (1969)) for stochastic programming to determine the upper and lower bounds on the optimal solution. Using the recursive definition of waiting time and idle time, the problem can be written as a 2-SLP. If the support of the procedures with finite first moments is not finite (the distribution of the actual duration is continuous), the support can be partitioned (for simplicity rectangular partition) so that $k$ sets of procedure realizations are created as scenarios with associated probabilities $p_k$. Denton and Gupta (2003) propose a sequential bounding algorithm which refines the partition at each step with a stopping condition based on absolute difference between upper and lower bound on objective. The partition at each step depends on the step number. The L-shaped algorithm (Van Slyke and Wets (1969)) follows:

**Step 1:** Set the index $v$=0

**Step 2:** $v = v + 1$. Solve the discrete problem above defined by partition $v$ using standard L-shaped method.

**Step 3:** If $f^{UB} - f_v \leq \epsilon$, then Stop. Otherwise refine the current partition and go to step 2.

The upper bound on the convex function can be obtained by applying aggregation bounds to this problem.

Denton et al. (2007) study the effects of appointment sequencing on the patient waiting time, OR idle time and session overtime assuming a discrete finite set of scenarios. Scenarios were generated by sampling with replacement from the historical data. The model is similar to Denton and Gupta (2003) but the assumption of the sequence being predetermined is relaxed. This updated two-stage stochastic mixed-integer program gives the optimal sequence and the optimal schedule and is combinatorial in nature, presumably $\mathcal{NP}$-hard. They propose three heuristic rules for approximating the optimal solution: sequence surgeries in order of increasing mean, variance and coefficient of variation of service duration. After sequencing the surgeries, the schedule is determined by solving the deterministic equivalent of the two-stage recourse problem. Numerical experiments show that the sequencing rule based on variation of service duration dominates the other two sequencing rules.

Mak et al. (2014) evaluate the performance of the heuristic *ordering by increasing variance (OV)* by simulating 1000 procedures from 3 different distributions (normal, gamma, lognormal). They compared the objective value found using the model proposed in Denton et al. (2007) without relaxing the sequence assumption with the the objective value calculated using the same model with relaxing the assumption and instead using the heuristic *increasing standard deviation (OV)*. OV sequences may or may not be optimal if the time allowances are calculated with respect to the true probability distribution but the numerical studies done by Mak et al. (2014) show OV has close-to-optimal performance.

Denton et al. (2007) also propose an interchange heuristic which is a local search starting with an initial feasible sequence. At each iteration the interchange heuristic searches randomly generated pairwise interchanges which improves the current solution:

**Step 1:** Find a feasible sequence, define $f^{UB}$, counter=1

**Step 2:** Use the L-shaped algorithm to obtain the solution $f_v$.

**Step 3:** If $f_v > f_{UB}$ then increase the counter and generate a new sequence and go to Step 2.

**Step 4:** If $f_v = f_{v-1}$ then $f_{UB} = f_v$. If counter has reached the maximum number of interchanges, stop. Otherwise generate new feasible sequence and return to Step 2.

According to the Bailey-Welch rule (Welch and Bailey (1952)), two appointments are scheduled to start at the beginning of the session and the remaining appointments are given at an interval equal to the mean service time. Kaandorp and Koole (2007) allow no-shows and use a local search method and the results from queuing theory to attempt to minimize the patient waiting time, physician idle time and overtime. They assume that service times are exponentially distributed with a common parameter and the operational time in a day is split into intervals of equal length, so that appointments can be spaced by discrete intervals. The operational time during a day is split into $T$ intervals with the same length. $n$ patients should be scheduled within these $T$ intervals. The decision variable is a vector $x = (x_1, \ldots, x_T)$, the number of patients scheduled at the start of an interval. There are $\binom{n+T-1}{n}$ possible schedules. Instead of trying all possible schedules to find the lowest objective value, Kaandorp and Koole (2007) propose a local search algorithm starting with a feasible solution and try to improve it. They define $T$ vectors ($U = (u_1, \ldots, u_T)$) to moves an appointment of a patient either to the former or the next interval (e.g. $x + u_1$ moves the patient scheduled in the first interval to the last).

$$\begin{Bmatrix} u_1 \\ u_2 \\ \vdots \\ u_T \end{Bmatrix} = \begin{Bmatrix} (-1, 0, \ldots, 0, 1) \\ (1, -1, 0, \ldots, 0) \\ \vdots \\ (0, \ldots, 0, 1, -1) \end{Bmatrix}$$

The local search algorithm is:

**Step 1:** Start with a schedule x
**Step 2:** For all $V \subseteq U$ s.t. $y = x + \sum_{u \in V} u \geq 0$
compute the objective function's value, if it is less than the former value, then $x := y$ and go back to Step 2
**Step 3:** $x$ is the local optimal (minimal) solution

The optimality of the algorithm can be proven by showing the objective is multimodular. Numerical results show that the interarrival times should be scheduled shorter at the beginning and towards the end of the day (dome-shaped). In other words interarrival times first increase then decrease. Under some conditions the optimal rule is close to the Bailey-Welch rule.

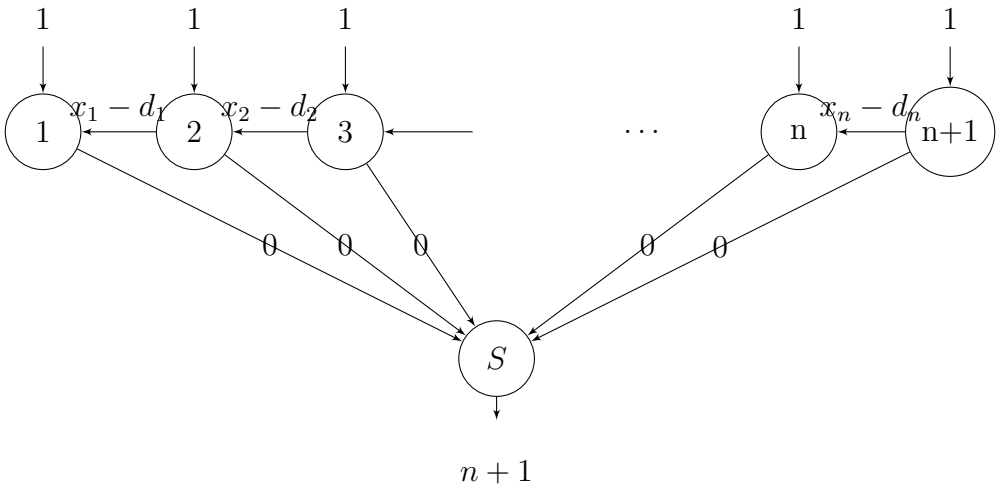
Begen and Queyranne (2011) assume processing times are given by a joint discrete distribution. For a given sequence they prove the existence of an optimal integer appointment vector minimizing the expected cost. Under the assumption of cost vectors being $\alpha-monotone$, the total cost function is L-convex so that the expected cost can be minimized in polynomial time. The same conclusion can be reached if the objective function is penalized for any deviation from due date. In Begen et al. (2012), the authors go one step further and develop a sampling-based approach to determine the number of independent samples to get a near-optimal solution with high probability. The underlying joint discrete distribution for job durations is unknown however the historical data of surgery durations is available. The sample size bound is a polynomial in the number of jobs and does not depend on the underlying distribution. Begen et al. (2012) show that the number of samples required to achieve $(1+\epsilon)$ multiplicative error bound with probability $(1-\gamma)$ is $O(n^6(1/\epsilon^4 \ln(n/\gamma)))$. Ge et al. (2013) extend those results to the case with piecewise linear cost functions: cost function is L-convex, under some conditions the problem can be solved in polynomial time and the bound on the number of samples required to have near-optimal solution is proven.

Although the existing literature primarily assumes that the underlying service distribution is known, fitting distributions requires a large amount of data. Hence in some research in appointment scheduling, distributional information is considered to be limited. Kong et al. (2013) assume that the mean, the covariance estimates and nonnegative support of the service durations are known. Their objective is to minimize the maximum expectation of the weighted sum of patient waiting times and the overtime (minimax approach). The idle time is also another performance measure which has to be minimized but it can be ignored, because adding the expected idle time to the objective only increases the objective by a constant and increases the weight corresponding to the overtime by 1. The objective is to determine the time allowances while minimizing the worst case expected value of the cost of the total weighted waiting time and the over-time among all distributions of the procedure durations with moments $\mu$ and covariance matrix $\Sigma$. The objective function can be represented as follows using the notation $c_t$ and$c_o$ being the cost of waiting time and overtime respectively, $T_j$ being the waiting time of procedure $j$, $x$ being the vector of random procedure durations and $d$ being the length of the time slot for $j$th procedure:

$$\min_{d} \max_{x \sim (\mu, \Sigma)} E\left[\sum_{j} c_t T_j + c_o T_n\right] = \min_{d} \max_{x \sim (\mu, \Sigma)} E\left[f(d, x)\right]$$

$f(d, x)$ can be represented as a network flow problem on a directed acyclic graph with costs given above the edges in Figure 1.3. They formulate the appointment scheduling problem

Figure 1.3: Network representation of the appointment scheduling problem given the sequence




as copositive programming which is not necessarily polynomial time solvable. An efficient semidefinite relaxation technique is developed to obtain near-optimal solutions.

Mak et al. (2014) assume only moments information of service duration is known, formulate the minimax appointment scheduling problem as a tractable conic programs. The first model they address is a mean-variance model which is formulated as second order cone program. Later they show sequencing jobs by increasing order of variance is optimal for their minimax model. Given the job sequence, the minimax appointment scheduling problem can be modeled as linear program. Exploiting this theorem, closed form expression for the optimal objective value of the mean-support model is provided. Sequencing jobs by increasing width of support is optimal under some assumptions.

Most literature assumes the underlying parametric distribution of procedure times are know. This assumption might introduce inaccuracy thus erroneous conclusions. There are few recent papers (such as Begen et al. (2012)) bounding the number of independent samples of the historical data to have a near optimal solution for a given sequence. We aim to develop a data-driven appointment scheduling algorithm determining the sequence then, estimating the start times and study the performance bounds of the most commonly used heuristics for sequencing the procedures.

### 1.2.2 Alternative Objective Functions of Appointment Scheduling

We also intend to consider alternative objectives in our research. Although the bulk of appointment scheduling literature considers minimizing the expected cost associated with earliness, tardiness and overtime, this risk neutral objective ignores the variability of the process, which in many cases seems to be a significant concern. Unfortunately risk averse appointment scheduling models have not been well studied.

#### Minimizing Value-at-Risk

We first intend to consider Value-at-Risk (VaR) which is defined as the threshold such that the probability that the objective exceeds the threshold is at most $(1-\alpha)$, given a random variable $X$ and confidence level $\alpha$:

$$VaR_X(\alpha) = \inf\{x : F_X(x) \geq \alpha\} \quad \text{for} \quad \alpha \in (0,1)$$

VaR is a non-convex and discontinuous function of the confidence level $\alpha$ for discrete distributions. VaR doesn't satisfy the axiom of subadditivity. (Artzner et al., 1999) Rockafellar and Uryasev (2000) introduce Conditional-Value-at-Risk (CVar) as an alternative to VaR.

#### Minimizing Conditional-Value-at-Risk

Given a random variable $X$ and confidence level $\alpha$, CVaR is defined as:

$$CVaR_X(\alpha) = E[X|X \geq VaR_X(\alpha)] = \frac{1}{1-\alpha}\int_{VaR_X(\alpha)}^{\infty} x dF_X(x) \quad \text{for} \quad \alpha \in (0,1)$$

CVaR controls the observations exceeding VaR whereas VaR does not. For any random variable $X$ with finite expected value and CVaR,the CVaR of the random variable is always greater than or equal to its expected value. This means that minimizing the CVaR tends to decrease its expectation.

Rockafellar and Uryasev (2000) prove that CVaR is a coherent risk measure having the following properties: transition-equivariant, subadditivity, positively homogeneous, convex, monotonic with respect to stochastic dominance of order 1, and monotonic with respect to monotonic dominance of order 2. Rockafellar and Uryasev (2000) show how to incorporate CVaR into an optimization framework as follow

$$CVaR_X(\alpha) = \min_{\eta \in R}\left\{\eta + \frac{1}{1-\alpha}E[(X-\eta)^+]\right\} \tag{1.1}$$

where $(a)^+ = max\{a, 0\}$.

In our appointment scheduling setting, we want to minimize the CVaR of total cost of earliness and tardiness. Rockafellar and Uryasev (2002) proposes the function (1.1) in convex if the objective function is convex.

Sarin et al. (2014) choose only total weighted tardiness as their performance measure and use CVaR as a criterion for stochastic scheduling. They formulate a scenario-based-mixed-integer program (MIP) to minimize CVaR for the total weighted tardiness assuming the procedure times to be the only random elements in this problem. Job precedence and completion time of jobs are decision variables. They present a specialized integer L-shaped algorithm and provide an alternative dynamic programming based heuristic procedure for large sized problems. They extend their model to the setting with identical parallel machines.

Jiang et al. (2015) consider a distributionally robust (DR) single server appointment scheduling problem given a fixed sequence of appointments with random no-shows and service durations. To understand the DR approach, first consider a classical stochastic program as an example: $\min_D E_x[c(D,x)]$ where $D$ represents a decision vector and $x$ represents a random vector. The probability distribution of $x$ is assumed unknown but a confidence set containing the actual distribution is known. A DR variant of the stochastic program minimizes the worst-case expected cost.

$$\min_{D} \max_{f_x \in \bullet F} E_x[c(D,x)]$$

Jiang et al. (2015) assume only the support and first moments are given and build two DR models incorporating the worst-case expected cost and the worst-case CVaR of earliness, tardiness and overtime as the objective or constraints.

We plan to design an algorithm focusing on different objectives of appointment scheduling.

### 1.2.3 Data-Driven Newsvendor Problems

As we will discuss in detail in subsequent sections, appointment scheduling (that is, estimating start times of jobs) is closely related to the Newsvendor model. Because our intention is to use dynamically updated data to improve appointment scheduling, we are inspired by approaches in the literature that use data to directly estimate newsvendor order quantities.

Liyanage and Shanthikumar (2005) introduce operational statistics, where the demand distribution function belongs to parametric distribution family and study the Newsvendor problem. They illustrate operational statistics for exponentially distributed demand. Instead of estimating the parameter, the optimal order quantity is estimated using the historical data directly such that the a priori expected profit is maximized. The goal is to find an operational statistic of the data and set it equal to the order quantity so that the objective is maximized. Chu et al. (2008) use Bayesian analysis to find the optimal operational statistic.

Bertsimas and Thiele (2005) combine historical data and an optimization framework to study the Newsvendor problem. They aim to provide robust solutions that perform well under most demand scenarios. They define a trimming factor ($\alpha$) which defines the percentage of the data points to be removed so that the objective is optimized over the remaining ones. In the Newsvendor setting the objective is optimized over $N_\alpha \lfloor (1-\alpha)N + \alpha \rfloor$ worst cases, where $N$ is the number of total data points. The optimum order quantity corresponds to $\lceil \frac{c_u}{c_u+c_o} N_\alpha \rceil$'th smallest data point, where $c_u$ and $c_o$ are the underage cost and overage cost respectively. If costs are equal to each other and the trimming factor is 0, then the ordering

quantity would be equal to the median. They also consider different cost structures (holding cost, recourse and fixed ordering cost) and formulate LP's or MIP's to maximize the objective while selecting the worst-case data points.

Wang et al. (2016) introduce a distributionally robust optimization model called 'likelihood robust optimization' (LRO) for the cases where the distribution of the input is unknown but there is enough historical data. They aim to achieve an empirical likelihood of at least $\exp(\gamma)$ among the distributions where the observed data is the support. They formulate the problem of optimizing the expected value of the objective over the worst case distribution of the data points which is chosen among the empirical distributions achieving a certain level of likelihood. Wang et al. (2016) study the Newsvendor problem and formulate a single convex optimization problem minimizing the maximum expected cost among the distributions which achieve a predetermined likelihood.

Our ultimate goal is to combine ideas from classical appointment scheduling with approaches introduced for the data-driven Newsvendor problem.

## 1.3 Model and Preliminary Results

Appointment scheduling of outpatient surgical services with stochastic procedure times minimizing the expected cost of waiting time and the idle time (deviation from the schedule), given the set of procedures need to be scheduled, involves:

1. Appointment Sequencing: Determining the sequence in which procedures are performed
2. Scheduling Start Time: Estimating the start time of each procedure

In our subsequent discussion we typically use tardiness to describe patient-surgeon waiting time and earliness instead of surgeon-OR idle time.

### 1.3.1 Notation

Our models assume that the set of tasks or procedures to be scheduled is known. The actual start time of a task depends on the prior tasks scheduled on the server (or in the same room, in the OR setting). If a task ends later than the scheduled start time of the next task, this results in waiting time for the next task (that is, for the patient and the surgeon), whereas if the task ends before the start time of the next task, the server (the OR) stays idle. We determine the estimates of the starting time of tasks while minimizing the penalty (cost) associated with the earliness and the tardiness. In the future, we will consider overtime cost in the objective because overtime is costly to the hospital.

There are multiple types of tasks which might have different distributions. The order in which they are processed depends on the scheduler. For instance if there are two types of tasks A and B and the processing sequence is A-B, the random task duration of A and B are denoted by $x_1$ and $x_2$ respectively. To ease exposition, WLOG the subscript depends on the

order rather than depending on the task type. Although we assume that the distribution of the task times are unkonwn, sometimes for analysis we use their distribution functions. The relevant variables and parameters are defined as follows:

| | |
|---|---|
| $S_j$ | Scheduled start (appointment) time of $j^{th}$ task |
| $D_j$ | Scheduled end time (due date) of $j^{th}$ task |
| $x_j$ | Random duration of $j^{th}$ task with density $f_j$ and cdf $F_j$ |
| $N_j$ | Number of observations of $j^{th}$ task |
| $\mathbf{x_j} = \{x_j^1, \ldots, x_j^{N_j}\}$ | Previous observations of duration of $j^{th}$ task |
| $C_j$ | Actual end (completion) time of $j^{th}$ task with density $f_{C_j}$ and cdf $F_{C_j}$ |
| $c_e$ | Penalty per unit time of earliness ($c_0$ or $\alpha$) |
| $c_t$ | Penalty per unit time of tardiness ($c_u$or $\beta$) |
| $c_o$ | Penalty per unit time of overtime ( $\gamma$) |
| $E_j$ | Earliness of $j^{th}$ task |
| $T_j$ | Tardiness of $j^{th}$ task |
| $n$ | Number of tasks to be scheduled per day per room |

Index $j$ refers to the order of the task. Earliness and tardiness are defined as follows:

$$E_j = \max(0, D_j - C_j) \tag{1.2}$$
$$T_j = \max(0, C_j - D_j) \tag{1.3}$$

Ultimately, we explore data-driven models in this setting using empirical distributions. In preparation, we explore models with known distributions with the objective of minimizing the expected earliness and tardiness.

### 1.3.2 Minimizing Expected Earliness and Tardiness with Known Distributions

We first consider the case where there are only two procedures to be scheduled. Later, we extend these results to the case where there are more than two procedures.

#### Two Procedures Model

Weiss (1990) initially considers the case with two procedures ($n = 2$): For a given sequence of procedures the scheduler estimates the start times in order to minimize the expected cost of the earliness and the tardiness. Note that much of the scheduling literature focuses on end times rather than start times. Since without loss of generality, the first procedure can be assumed to start at time 0, and since the scheduled start time of second procedure is equal to the scheduled end time of the first procedure, we can equivalently focus on the end time of the first procedure, and effectively ignore the second procedure. The total expected cost is:

$$E[\text{cost}] = E\left[\sum_j c_e E_j + \sum_j c_t T_j\right]$$

$$
\begin{aligned}
&= c_e \int_0^{D_1} (D_1 - x_1) f_1(x_1) dx_1 + c_t \int_{D_1}^{\infty} (x_1 - D_1) f_1(x_1) dx_1 \\
&= (c_e + c_t) \int_0^{D_1} D_1 f_1(x_1) dx_1 - (c_e + c_t) \int_0^{D_1} x_1 f_1(x_1) dx_1 + c_t E[x_1] - c_t D_1
\end{aligned}
$$

The expression above is similar to the expected cost in "Newsvendor" problem. The expected cost is convex with respect to end time, $D_1$ which can be minimized by finding the value of $D_1$ such that:

$$
\frac{c_t}{c_e + c_t} = F_1(D_1) = \int_0^{D_1} f_1(x_1) dx_1 \tag{1.4}
$$

Equation (1.4) above is used to determine the scheduled end times of the first procedure. In order to find out if the given sequence is optimal, Weiss (1990) examines the expected cost function after plugging the equation (1.4) in. The resulting expected cost is given by:

$$
\begin{aligned}
E[\text{cost}] =& c_e \int_0^{D_1} (D_1 - x_1) f_1(x_1) dx_1 + c_t \int_{D_1}^{\infty} (x_1 - D_1) f_1(x_1) dx_1 \\
=& c_e \int_0^{D_1} D_1 f_1(x_1) dx_1 - c_t \int_{D_1}^{\infty} D_1 f_1(x_1) dx_1 - c_e \int_0^{D_1} x_1 f_1(x_1) dx_1 \\
&+ c_t \int_{D_1}^{\infty} x_1 f_1(x_1) dx_1 \\
=& c_e D_1 F(D_1) - c_t D_1 \left(1 - F(D_1)\right) - c_e \int_0^{D_1} x_1 f_1(x_1) dx_1 + c_t \int_{D_1}^{\infty} x_1 f_1(x_1) dx_1 \\
=& c_e D_1 \frac{c_t}{c_e + c_t} - c_t D_1 \frac{c_e}{c_e + c_t} - c_e \int_0^{D_1} x_1 f_1(x_1) dx_1 + c_t \int_0^{\infty} x_1 f_1(x_1) dx_1 \\
&- c_t \int_0^{D_1} x_1 f_1(x_1) dx_1 \\
=& -(c_e + c_t) \int_0^{D_1} x_1 f_1(x_1) dx_1 + c_t E[x_1] \\
=& -(c_e + c_t) E[x_1 | x_1 < D_1] \frac{c_t}{c_e + c_t} + c_t E[x_1] \\
=& c_t (E[x_1] - E[x_1 | x_1 < D_1]) \\
=& c_t (E[C_1] - E[C_1 | C_1 < D_1])
\end{aligned} \tag{1.5}
$$

Scheduling the procedure with the smaller value of (1.5) first, minimizes the expected cost. Weiss (1990) shows that ordering by (1.5) is the same as ordering by variance for both uniform and exponential distributions.

Denton et al. (2007) extend Weiss's original sequencing argument and proposes it is optimal to sequence procedures according to the convex ordering if it exists. If $x_1 \leq_{cx} x_2$

which means $E[\phi(x_1)] \leq E[\phi(x_1)]$ for all convex $\phi$, then the sequence $\{1, 2\}$ is optimal.

$$\begin{aligned} E[\text{cost}_{\{1,2\}}] &= E\left[c_e E_1 + c_t T_1\right] \\ &= c_e E[\max(0, D_1 - C_1)] + c_t E[\max(0, C_1 - D_1)] \\ &\leq c_e E[\max(0, D_2 - C_2)] + c_t E[\max(0, C_2 - D_2)] \\ &= E[\text{cost}_{\{1,2\}}] \end{aligned}$$

It is important to note that convex ordering requires the expected procedure times to be the same. ($E[x_1] = E[x_2]$)

### $n$ Procedures Model

Now consider the case with more than two procedures ($n > 2$) to be scheduled: Given the sequence the scheduler follows a similar procedure to calculate the start times for all $n$ procedures. Again without loss of generality, the first procedure can be assumed to start at time 0. And the start time of other procedures is equal to the end time of its previous procedure. The following analysis gives the end time of procedures. Since it is not necessary, calculating the end time of the last procedure is ignored. The objective is :

$$\begin{aligned} E[\text{cost}] &= E\left[\sum_j c_e E_j + \sum_j c_t T_j\right] \\ &= \sum_j E\left[c_e \max(0, D_j - C_j) + c_t \max(0, C_j - D_j)\right] \end{aligned}$$

We need to note that the objective is similar the sum of expected costs of $n$ "Newsvendor" problems. The objective is still convex because the sum of convex functions is convex. Note that the estimate of $j^{th}$ end time depends on the distribution of the $j^{th}$ completion time which is the convolution of distributions of procedure times for the previous procedures scheduled and the current procedure.

- *Early start is allowed:* If the $j^{th}$ procedure can start as soon as the previous one, $(j-1)^{th}$, is finished, the completion time $C_j$ is given by the distribution of $C_j = C_{j-1} + x_j = \sum_{i=1}^{j} x_i$. The corresponding scheduled end time can be computed as follows:

$$\frac{c_t}{c_e + c_t} = F_{C_j}(D_j) = \int_0^{D_j} f_{C_j}(C_j) dC_j \tag{1.6}$$

- *Early start is not allowed:* The scheduler first needs to determine the distribution of the completion times, conditional on the scheduled end time of the previous procedures. The convolution of $C_j = \max(C_{j-1}, D_{j-1}) + x_j$ is used to calculate $D_j$ in the following way:

$$\frac{c_t}{c_e + c_t} = F_{C_j}(D_j) = \int_0^{D_j} f_{C_j}(C_j | D_1, \ldots, D_{j-1}) dC_j \tag{1.7}$$

Ordering procedures by (1.5) is not always optimal for the more than 2 procedure case. Weiss (1990) considers an example with two discrete distributions and shows that the optimal order of the first two procedures might change, if there is a third procedure to be scheduled. The objective of the case where there are more than 2 procedures:

$$\begin{aligned} E[\text{cost}] &= E\left[\sum_j c_e E_j + \sum_j c_t T_j\right] \\ &= \sum_j E\left[c_e \max(0, D_j - C_j) + c_t \max(0, C_j - D_j)\right] \\ &= \sum_j c_t (E[C_j] - E[C_j | C_j < D_j]) \end{aligned} \tag{1.8}$$

Ordering procedures by (1.5) ignores the fact that completion time is the convolution of the distributions of previous procedure times. The sequence minimizing each term in the sum (1.8) doesn't necessarily correspond the ordering procedures by (1.5).

For the case of $n > 2$, there is no such result to sequence the procedures as showed for the case of $n = 2$. Motivated by the insight from the case of $n = 2$, Denton et al. (2007) propose some easy-to-implement heuristics for sequencing procedures:

1. Sequence procedures in order of increasing mean of the duration
2. Sequence procedures in order of increasing standard deviation of duration
3. Sequence procedures in order of increasing coefficient of variation of duration

While minimizing expected earliness and tardiness cost, Denton and Gupta (2003), Denton et al. (2007), Kaandorp and Koole (2007) try to minimize another OR performance measure: expected overtime of the day. The expected overtime of the day is equal to the expected tardiness of the last procedure scheduled on that day. So the cost associated with the expected tardiness of the last procedure is expected to be higher relative to the cost associated with the expected tardiness of the previous procedures. The new objective becomes:

$$E[\text{cost}] = E\left[\sum_j c_e E_j + \sum_j c_t T_j + c_o T_n\right]$$

Since appointment sequencing problem is combinatorial optimization problem, as a result many studies propose heuristics or consider simulation. Sequencing the procedures in order of increasing standard deviation of duration is the most commonly used heuristic.

### 1.3.3 Minimizing Expected Earliness and Tardiness using the Empirical Distribution

Previous work has assumed distributions are known, but there is often not enough data to accurately estimate underlying distributions. In this section, without assuming any parametric

distribution, the empirical distribution function associated with the each procedure is directly used. We apply the n-procedures formulation defined above in the empirical setting where $\mathbf{x_j} = \{x_j^1, \ldots, x_j^{N_j}\}$ represents the previous observations of duration of the procedure in the $j$th order in the sequence determined by the heuristic. These observations in the data are ranked in increasing order, so that $x_j^{(1)} \leq \cdots \leq x_j^{(N_j)}$. $\mathbf{C_j} = \{C_j^1, \ldots, C_j^{N_1 \cdots N_j}\}$ represents (support) possible values of the completion time of the procedure in $j$th order, which is calculated using the data. $\mathbf{x_j}$ and $\mathbf{C_j}$ are multisets that do not necessarily have unique entries so that they allow multiple instances of elements unlike sets. It is explicit that $\mathbf{x_1} = \mathbf{C_1}$.

Given these assumptions, we apply the most common heuristic from the literature for sequencing, increasing order of standard deviations. Note that for the trivial case of two procedures after estimating start times using the equation (1.4), the sequence given by the heuristic is optimal for some distributions. (e.g. normal, uniform and exponential distributions comply.)

For instance take the case where $c_t = c_e$ and there are only two independent procedures which are exponentially distributed with parameter $\lambda_1$ and $\lambda_2$. The mean and the median of the random variable distributed exponentially with the parameter of $\lambda$ are $\frac{1}{\lambda}$ and $\frac{\ln(2)}{\lambda}$ respectively. It is shown in the previous section that setting $D_1$ equal the median of the first procedure's duration ($D_1 = F_{C_1}^{-1}(\frac{c_t}{c_e+c_t}) = F_{C_1}^{-1}(0.5)$) minimizes the the objective. The resulting objective is:

$$\begin{aligned}
E[|C_1 - D_1|] &= \int_0^{D_1} (D_1 - x_1) f_1(x_1) dx_1 + \int_{D_1}^{\infty} (x_1 - D_1) f_1(x_1) dx_1 \\
&= \int_0^{\ln(2)/\lambda} (\ln(2)/\lambda - x_1) \lambda e^{-\lambda x_1} dx_1 + \int_{\ln(2)/\lambda}^{\infty} (x_1 - \ln(2)/\lambda) \lambda e^{-\lambda x_1} dx_1 \\
&= \frac{(\lambda x - \ln(2) + 1) e^{-\lambda x}}{\lambda} \Big]_0^{\ln(2)/\lambda} - \frac{(\lambda x - \ln(2) + 1) e^{-\lambda x}}{\lambda} \Big]_{\ln(2)/\lambda}^{\infty} \\
&= \frac{\ln(2)}{\lambda}
\end{aligned}$$

Since the resulting objective is a positive multiple of the standard deviation, sequencing procedures in increasing order of standard deviation gives the optimal order if the underlying distribution is exponential. This sequencing heuristic is not necessarily optimal in general.

Our SD-MAD algorithm using the empirical distributions follows:

**Step 1:** Sequence the procedures using the heuristic : 'Sequence procedures in order of increasing standard deviation of duration'

**Step 2:** Estimate start times

Step 2 of the SD-MAD Algorithm to estimate the start times:

- If there are only two procedures to be scheduled, assume WLOG that the first procedure is scheduled to start at time 0:

1. Arrange the observed durations of the first procedure in the sequence proposed by the heuristic in increasing order.

$$x_1^{(1)} \leq x_1^{(2)} \leq \cdots \leq x_1^{(\lceil N_1 \cdot c_t/(c_e+c_t)\rceil)} \leq \cdots \leq x_1^{(N_1)}$$

2. Set the scheduled end time of the first procedure which is also the scheduled start time of the second procedure equal to $\frac{c_t}{(c_t+c_e)}$-quantile of first procedure's duration (also the completion time).

$$D_1 = x_1^{(\lceil N_1 \cdot c_t/(c_e+c_t)\rceil)}$$

- If there are more than two procedures to be scheduled, schedule the first two as it explained above. To schedule the $j^{\text{th}}$ procedure ($j \in 2, \ldots, n-1$):

1. – *If early start is allowed:* Calculate all possible completion times, $\mathbf{C_j}$, recursively by summing all the elements of $\mathbf{x_j} = \{x_j^1, \ldots, x_j^{N_j}\}$ with all the elements of $\mathbf{C_{j-1}} = \{C_{j-1}^1, \ldots, C_{j-1}^{N_1 \cdots N_{j-1}}\}$, starting with the condition $\mathbf{x_1} = \mathbf{C_1}$.

$$C_j = \{x_j^k + C_{j-1}^\ell : k \in \{1, \ldots, N_j\}, \ell \in \{1, \ldots, M_{j-1}\}\},$$

where $M_j = \prod_{i=1}^{j} N_i$

– *Else if early start is not allowed:* Calculate all possible completion times, $\mathbf{C_j}$, recursively by summing all the elements of $\mathbf{x_j} = \{x_j^1, \ldots, x_j^{N_j}\}$ with all the elements of $\mathbf{C_{j-1}} = \{\max\{D_{j-1}, C_{j-1}^1\}, \ldots, \max\{D_{j-1}, C_{j-1}^{N_1 \cdots N_{j-1}}\}\}$ starting with the condition $\mathbf{x_1} = \mathbf{C_1}$.

$$C_j = \{x_j^k + \max\{D_{j-1}, C_{j-1}^\ell\} : k \in \{1, \ldots, N_j\}, \ell \in \{1, \ldots, M_{j-1}\}\},$$

where $M_j = \prod_{i=1}^{j} N_i$

There will be $N_1 \cdots N_j$ possibilities and arrange those numbers in increasing order.

$$C_j^{(1)} \leq C_j^{(2)} \leq \cdots \leq C_j^{(N_1 \cdots N_j)}$$

2. Set the scheduled end time of the task which is also equal to the scheduled start time of the next task equal to $\frac{c_t}{(c_t+c_e)}$-quantile of $j$th procedure's completion time.

$$D_1 = C_j^{(\lceil N_1 \cdots N_j \cdot c_t/(c_e+c_t)\rceil)} = C_j^{(\lceil M_j \dot{c}_t/(c_e+c_t)\rceil)}$$

### 1.3.4 Performance Bound of the Sequencing Heuristic Based on the Standard Deviation of Procedure Duration

In this section, we explore the worst case performance of the SD-MAD heuristic described above. Specifically, if $z^*$ is the optimal objective function value for an instance of the problem, and $z^h$ is the objective function value resulting from applying the algorithm in the previous section.

We show the worst-case bound on $z^h/z^*$. First we find the bound for the case where there are only two procedures and same earliness and tardiness penalties. Later we extend the results to general case.

#### Two Procedures with $c_e = c_t = 1$

We assume that there are only two procedures to be scheduled, and the penalties for earliness and tardiness are same. The objective is:

$$z^* = \min_{D_1} \quad E[|C_1 - D_1|]$$

We use SD-MAD to determine the sequence and the start times of procedures. Since the empirical distribution function puts mass $\frac{1}{N}$ at each data point, $x_1^i$, the resulting objective function value is:

$$\begin{aligned} E[|C_1 - D_1|] &= E[|x_1 - D_1|] \\ &= \frac{1}{N_1}\sum_{i=1}^{N_1} |x_1^i - D_1| \end{aligned}$$

**Lemma 1.** *For a given sequence, Step 2 of the SD-MAD algorithm determines the optimal start times of the procedures: If there are only two procedures and the penalties of earliness and tardiness are same, the scheduled start time of the first procedure is 0 and the scheduled start time of the second procedure is the median of the first procedure's duration.*

*Proof.* Let $x_1^{(1)} \leq x_1^{(2)} \leq \cdots \leq x_1^{(N_1)}$ and for any $D_1 \leq x_1^{(1)}$, as $D_1$ increases each term in the objective's summation decreases. For any $D_1$ greater than $x_1^{(1)}$ and less than $x_1^{(2)}$ as $D_1$ increases, only the first term of the summation increases while each of the rest $N_1 - 1$ terms decreases at the same amount. This means the total decrease in the objective is greater than the increase. $D_1$ can be increased up to $x_1^{(\lceil N_1/2 \rceil)}$ so that the number of terms having a decrease is still greater than the number of terms having an increase. Increasing $D_1$ above the median only increases the objective.

This lemma also can be interpreted as the mean absolute deviation $MAD(D) = E[|x_1 - D|]$ is minimized by the median of the random variable $x_1$. □

Now we focus on Step 1 of SD-MAD to measure how $z^h$ can be different from $z^*$ if we sequence the procedure with lower standard deviation first. Since there are only two

procedures, there are only 2! possible orderings. $z_p$ represents the objective function value of the sequence where the procedure of type $p$ scheduled first.

**Lemma 2.** *The lower bound on the objective function of a given sequence where the procedure type p is scheduled first:*

$$z_p \geq \frac{\sigma_p}{\sqrt{N_p}}$$

*where $N_p$ is the number of observations of procedure type p and $\sigma_p$ is the sample standard deviation of procedure duration of type p.*

*Proof.* We have the set, $\mathbf{x_p} = \{x_p^1, \ldots, x_p^{N_p}\}$, of observed durations of the procedure type $p$ (from the data) and its median and its mean are defined as $m_p$ and $\mu_p$ respectively.

$$\begin{aligned}
(z_p)^2 &= \left(\frac{1}{N_p}\sum_{i=1}^{N_p}|x_p^i - m_p|\right)^2 \\
&\geq \frac{1}{N_p^2}\sum_{i=1}^{N_p}|x_p^i - m_p|^2 \qquad (1.9)\\
&\geq \frac{1}{N_p^2}\sum_{i=1}^{N_p}|x_p^i - \mu_p|^2 \qquad (1.10)\\
&= \frac{1}{N_p^2}\sum_{i=1}^{N_p}(x_p^i - \mu_p)^2 \\
&= \frac{1}{N_p}\sigma_p^2
\end{aligned}$$

(1.9) follows from the multinomial theorem, powers of the sums with positive entries (i.e. $a_i > 0$):

$$\left(\sum_i a_i\right)^2 = \sum_i\sum_j a_i a_j = \sum_i a_i^2 + \sum_i\sum_{j\neq i} a_i a_j \geq \sum_i a_i^2$$

(1.10) follows from the fact that the sum of squared deviations is minimized when the deviations are calculated around the sample mean. □

**Lemma 3.** *The upper bound on the objective function of a given sequence where the procedure type p is scheduled first is:*

$$z_p \leq \sigma_p$$

*where $\sigma_p$ is the sample standard deviation of procedure duration of type p.*

*Proof.* Using the same notation in the previous lemma:

$$\begin{aligned} z_p &= \frac{1}{N_p}\sum_{i=1}^{N_p}|x_p^i - m_p| \\ &\leq \frac{1}{N_p}\sum_{i=1}^{N_p}|x_p^i - \mu_p| \qquad (1.11) \\ &= \frac{1}{N_p}\sqrt{\left(\sum_{i=1}^{N_p}|x_p^i - \mu_p|\right)^2} \\ &= \frac{1}{N_p}\sqrt{\left(\sum_{i=1}^{N_p}\mathbf{1}\cdot|x_p^i - \mu_p|\right)^2} \\ &\leq \frac{1}{N_p}\sqrt{N_p\sum_{i=1}^{N_p}(x_p^i - \mu_p)^2} \qquad (1.12) \\ &= \sigma_p \end{aligned}$$

(1.11) follows from the lemma 1, the mean absolute deviations is minimized by the median of the sample, and (1.12) follows from the Cauchy-Schwarz inequality. □

The optimal objective function $z^*$ is less than or equal to the objective values of any given sequence so that the upper bounds for both sequences found via Lemma 3 are also an upper bounds on $z^*$. Also, $z^*$ should be greater than the smallest lower bound calculated for both orderings using the Lemma 2. Thus:

$$\min_p\left\{\frac{\sigma_p}{\sqrt{N_p}}\right\} \leq z^* \leq \min_p \sigma_p$$

The SD-MAD algorithm schedules the procedure with smallest standard deviation first, so,

$$\frac{\min_p \sigma_p}{\sqrt{N_p}} \leq z^h \leq \min_p \sigma_p$$

We can conclude that $\frac{z^h}{z^*} \leq \frac{\min_p \sigma_p}{\min_p\left\{\frac{\sigma_p}{\sqrt{N_p}}\right\}}$. In other words the ratio grows as the square root of number of observations of the procedure with relatively larger sample set grows. If the standard deviation of both procedures are equal to each other the bound will be $\sqrt{\max_p\{N_p\}}$, which is also the maximum value the ratio can have.

We would like to prove that this is the tightest bound achievable. In the next section we formulate a mathematical programming model to find a class of instances maximizing the expected total tardiness and earliness and another class of instances minimizing the expected total tardiness and earliness.

**Performance Bound of the Heuristic**

We consider a setting in which we have procedures with equal standard deviation. Among all such sets of procedures we would like to find a class of examples maximizing and another class of examples minimizing the objective. We formulate a mathematical programming model where the decision variables, $\mathbf{z} = \{z_1, \ldots, z_N\}$, are the sample data points, $m$ denotes their median and $\mu$ is their mean.

$$\begin{array}{lll} \mathbf{P}_{\min} & \min & \dfrac{1}{N}\sum_{i=1}^{N}|z_i - m| \\ & \text{s.t.} & \dfrac{\sum_{i=1}^{N} z_i^2}{N} - \left(\dfrac{\sum_{i=1}^{N} z_i}{N}\right)^2 = C^2 \\ & & zi \geq 0 \quad \forall i \end{array} \qquad \begin{array}{lll} \mathbf{P}_{\max} & \max & \dfrac{1}{N}\sum_{i=1}^{N}|z_i - m| \\ & \text{s.t.} & \dfrac{\sum_{i=1}^{N} z_i^2}{N} - \left(\dfrac{\sum_{i=1}^{N} z_i}{N}\right)^2 = C^2 \\ & & zi \geq 0 \quad \forall i \end{array} \tag{1.13}$$

In order to visualize the model we set the number of data points ($N$) equal to 5 ($\mathbf{z} = \{z_1, \ldots, z_5\}$) and rewrite the model with an assumption of $0 \leq z_1 \leq \cdots \leq z_5$ which means $z_3$ is the median ($m$). The resulting mathematical models are:

$$\begin{array}{lll} \min & \dfrac{1}{5}(z_4 + z_5 - z_1 - z_2) & \\ \text{s.t.} & \dfrac{\sum_{i=1}^{5} z_i^2}{5} - \left(\dfrac{\sum_{i=1}^{5} z_i}{5}\right)^2 = C^2 & \\ & z_1 \geq 0 & \\ & z_2 - z_1 \geq 0 & \\ & z_3 - z_2 \geq 0 & \\ & z_4 - z_3 \geq 0 & \\ & z_5 - z_4 \geq 0 & \end{array} \quad \text{AND} \quad \begin{array}{lll} \max & \dfrac{1}{5}(z_4 + z_5 - z_1 - z_2) & \\ \text{s.t.} & \dfrac{\sum_{i=1}^{5} z_i^2}{5} - \left(\dfrac{\sum_{i=1}^{5} z_i}{5}\right)^2 = C^2 & \\ & z_1 \geq 0 & \\ & z_2 - z_1 \geq 0 & \\ & z_3 - z_2 \geq 0 & \\ & z_4 - z_3 \geq 0 & \\ & z_5 - z_4 \geq 0 & \end{array}$$

**Theorem 1.** *Given $a \in \mathbb{R}_{\geq 0}$ there exist $b \in \mathbb{R}_{\geq 0}$ such that the optimal procedure times maximizing the objective is in the following form: $(z_1 = a, \ldots, z_k = a, z_{k+1} = b, \ldots, z_N = b)$ if $N$ is even, $(z_1 = a, \ldots, z_k = a, z_{k+1} = \frac{a+b}{2}, z_{k+2} = b, \ldots, z_N = b)$ if $N$ is odd, where $k = \lfloor N/2 \rfloor$ and the variance constraint (first constraint in (1.13)) is met.*

**Theorem 2.** *Given $c \in \mathbb{R}_{\geq 0}$ there exist $d \in \mathbb{R}_{\geq 0}$ such that the optimal procedure times minimizing the objective is in form of: $(z_1 = c, \ldots, z_{N-1} = c, z_N = d)$ and the variance constraint (first constraint in (1.13)) is met.*

To prove both theorems above, we simplify and redefine the model. It is important to point out that among those $N$ data points we only need to know which one is the median:

Half of the points would be greater than or equal to that point whereas the rest are less than or equal to that. For simplicity, WLOG we assume the median is equal to 0, half of the data points ($w$) would be non-negative and the rest ($y$) would be non-positive. (We shift all the points so that the median will be 0, but this process does not change the variance.) The objective can be rewritten as the following, where $k = \lfloor N/2 \rfloor$, $N$ is the number of data points:

$$\ell(w, y) = \sum_{i=1}^{k} w_i - \sum_{i=1}^{k} y_i$$

The variance of the data points can be calculated as:

$$G^2(w, y) = \frac{\sum_{i=1}^{k} w_i^2 + \sum_{i=1}^{k} y_i^2}{N} - \left( \frac{\sum_{i=1}^{k} w_i + \sum_{i=1}^{k} y_i}{N} \right)^2$$

Since the variance is equal to a constant, ($C^2$ (first constraint in (1.13))), without loss of generality that constant is set equal to 1. The mathematical model becomes:

$$\mathbf{P1}_{\min} \quad \begin{aligned} \min \quad & \ell(w, y) \\ \text{s.t.} \quad & G^2(w, y) = 1 \\ & w_i \geq 0 \quad \forall i \in [1, \ldots, k] \\ & y_i \leq 0 \quad \forall i \in [1, \ldots, k] \end{aligned} \qquad \text{AND} \qquad \mathbf{P1}_{\max} \quad \begin{aligned} \max \quad & \ell(w, y) \\ \text{s.t.} \quad & G^2(w, y) = 1 \\ & w_i \geq 0 \quad \forall i \in [1, \ldots, k] \\ & y_i \leq 0 \quad \forall i \in [1, \ldots, k] \end{aligned}$$

The procedure times are of course non-negative, but the optimal solution to **P1** consists of both positive and negative numbers. With a proper scaling the optimal solution to the model above can be made non-negative. The optimal solution of **P** is equal to the optimal solution of **P1** multiplied by $C$ (standard deviation of the data points).

**P1** is not convex, but if we swap the objective and the variance constraint, the resulting model will be convex. $\mathbf{P1}_{\min}$ corresponds to $\mathbf{P2}_{\max}$ whereas $\mathbf{P1}_{\max}$ corresponds to $\mathbf{P2}_{\min}$.

$$\mathbf{P2}_{\max} \quad \begin{aligned} \max \quad & G^2(w, y) \\ \text{s.t.} \quad & \ell(w, y) = 1 \\ & w_i \geq 0 \quad \forall i \in [1, \ldots, k] \\ & y_i \leq 0 \quad \forall i \in [1, \ldots, k] \end{aligned} \qquad \text{AND} \qquad \mathbf{P2}_{\min} \quad \begin{aligned} \min \quad & G^2(w, y) \\ \text{s.t.} \quad & \ell(w, y) = 1 \\ & w_i \geq 0 \quad \forall i \in [1, \ldots, k] \\ & y_i \leq 0 \quad \forall i \in [1, \ldots, k] \end{aligned}$$

**Lemma 4.** *After scaling the variables in the optimal solution of P2, the optimal solution of P1 will be obtained.*

*Proof.* Lets $(w_{P1}, y_{P1})$ and $(w_{P2}, y_{P2})$ be the minimizers of **P1** and maximizers of **P2** respectively, and define two strictly positive variables $\ell_1$ and $G_2$:

$$\ell_1 = \ell(w_{P1}, y_{P1}) \qquad G_2^2 = G^2(w_{P2}, y_{P2})$$

$\ell_1$ is not necessarily equal to 1, so $(w_{P1}, y_{P1})$ may not be a feasible solution of **P2**. After dividing each variable $(w_{P1}, y_{P1})$ by $\ell_1$, the resulting set is a feasible solution to **P2**

$$\ell(w_{P1}/\ell_1, y_{P1}/\ell_1) = \sum_{i=1}^{k} \frac{w_{P1}^i}{\ell_1} - \sum_{i=1}^{k} \frac{y_{P1}^i}{\ell_1} = \frac{\ell(w_{P1}, y_{P1})}{\ell_1} = 1$$

Since $(w_{P2}, y_{P2})$ are the maximizers of **P2**

$$G_2^2 = G^2(w_{P2}, y_{P2}) \geq G^2(w_{P1}/\ell_1, y_{P1}/\ell_1) = \frac{G^2(w_{P1}, y_{P1})}{\ell_1^2} = \frac{1}{\ell_1^2}$$

$G_2^2$ is not necessarily equal to 1, so $(w_{P2}, y_{P2})$ may not be a feasible solution of P1. After dividing each variable $(w_{P2}, y_{P2})$ by $G_2$, the resulting set is a feasible solution to **P1**

$$\begin{aligned} &G^2(w_{P2}/G_2, y_{P2}/G_2) \\ &\quad = \frac{\sum_{i=1}^{k}(w_{P_2}^i/G_2)^2 + \sum_{i=1}^{k}(y_{P_2}^i/G_2)^2}{N} - \left(\frac{\sum_{i=1}^{k} w_{P_2}^i/G_2 + \sum_{i=1}^{k} y_{P_2}^i/G_2}{N}\right)^2 \\ &\quad = \frac{G^2(w_{P2}, y_{P2})}{G_2^2} = 1 \end{aligned}$$

Since $(w_{P1}, y_{P1})$ are the minimizers of **P1**

$$\ell_1 = \ell(w_{P1}, y_{P1}) \leq \ell(w_{P2}/G_2, y_{P2}/G_2) = \frac{\ell(w_{P2}, y_{P2})}{G_2} = \frac{1}{G_2}$$

$G_2$ is the standard deviation and there are at least two distinct numbers in the data set so it is strictly positive, $\ell_1$ is by definition strictly positive. Since $G_2^2 \geq \frac{1}{\ell_1^2}$ and $\ell_1 \leq \frac{1}{G_2}$ hold, $\ell_1 = \frac{1}{G_2}$. By scaling the optimal solutions of one model we may obtain the optimal solution of the other.
We follow the same steps to prove that after scaling the minimiziers of **P2**, the maximizers of **P1** can be computed. □

**Lemma 5.** $\left(\sqrt{\frac{N}{N-1}}, \ldots, \sqrt{\frac{N}{N-1}}, 0, -\sqrt{\frac{N}{N-1}}, \ldots, -\sqrt{\frac{N}{N-1}}\right)$ *is the maximizer of* ***P1****, if* $N$ *is odd. If* $N$ *is even,* $(1, \ldots, 1, -1, \ldots, -1)$ *is the maximizer of* ***P1****.*

*Proof.* Solving **P2** is easier relative to solving **P1**, since the objective function of **P2** is convex and the constraints are linear. Since **P2** is convex, any local minimum is a global minimum. If the proposed point satisfies the KKT conditions, then it is the global minimum.

**KKT Conditions of P2$_{\min}$:**

1.

$$\sum_{i=1}^{k} w_i - \sum_{i=1}^{k} y_i = 1$$

$$
\begin{aligned}
-w_i &\leq 0 \qquad \forall i \in \{1,\dots,k\} \\
y_i &\leq 0 \qquad \forall i \in \{1,\dots,k\}
\end{aligned}
$$

2.

$$
\begin{bmatrix}
2w_1 - 2\left(\frac{\sum_{i=1}^{k} w_i + \sum_{i=1}^{k} y_i}{N}\right) \\
\vdots \\
2w_k - 2\left(\frac{\sum_{i=1}^{k} w_i + \sum_{i=1}^{k} y_i}{N}\right) \\
2y_1 - 2\left(\frac{\sum_{i=1}^{k} w_i + \sum_{i=1}^{k} y_i}{N}\right) \\
\vdots \\
2y_k - 2\left(\frac{\sum_{i=1}^{k} w_i + \sum_{i=1}^{k} y_i}{N}\right)
\end{bmatrix}
+ \mu
\begin{bmatrix} 1 \\ \vdots \\ 1 \\ -1 \\ \vdots \\ -1 \end{bmatrix}
+
\begin{bmatrix} -\lambda_1 \\ \vdots \\ -\lambda_k \\ \gamma_1 \\ \vdots \\ \gamma_k \end{bmatrix}
= 0
$$

3.

$$
\begin{aligned}
\lambda_i &\geq 0 \qquad \forall i \in \{1,\dots,k\} \\
\gamma_i &\geq 0 \qquad \forall i \in \{1,\dots,k\}
\end{aligned}
$$

4.

$$
\begin{aligned}
\lambda_i w_i &= 0 \qquad \forall i \in \{1,\dots,k\} \\
\gamma_i y_i &= 0 \qquad \forall i \in \{1,\dots,k\}
\end{aligned}
$$

If $N$ is even, $(w_1 = 1/N, \dots, w_k = 1/N, y_1 = -1/N, \dots, y_k = -1/N)$ satisfies the KKT conditions, if $N$ is odd, $(w_1 = 1/(N-1), \dots, w_k = 1/(N-1), w_{k+1} = 0, y_1 = -1/(N-1), \dots, y_k = -1/(N-1))$ satisfies the KKT conditions.

If $N$ is even, $(w_1 = 1/N, \dots, w_k = 1/N, y_1 = -1/N, \dots, y_k = -1/N)$ is the minimizer of **P2**. After the proper scaling, $(1, \dots, 1, -1, \dots, -1)$ is the maximizer of **P1**.
If $N$ is odd, $(w_1 = 1/(N-1), \dots, w_k = 1/(N-1), w_{k+1} = 0, y_1 = -1/(N-1), \dots, y_k = -1/(N-1))$ is the minimizer of **P2**. $\left(\sqrt{\frac{N}{N-1}}, \dots, \sqrt{\frac{N}{N-1}}, 0, -\sqrt{\frac{N}{N-1}}, \dots, -\sqrt{\frac{N}{N-1}}\right)$ is the maximizer of **P1** after the proper scaling.

□

**Lemma 6.** $\left(\frac{N}{\sqrt{N-1}}, 0, \dots, 0\right)$ *is the minimizer of* ***P1****.*

*Proof.* Thanks to the convexity of the model, the global maximum is at the extreme points (basic feasible solution) of the feasible set. The extreme points of **P2** are in the form of (0,...,0,1,0,...,0).

$(w_1 = 1, w_2 = 0, \ldots, y_k = 0)$ is one of the maximizers of **P2**. This set is scaled to satisfy the equality constraint in P1. The scaled set, $\left(\frac{N}{\sqrt{N-1}}, 0, \ldots, 0\right)$, is the minimizer of **P1**. □

It is important to point out:

- The minimum objective value of **P1** for a fixed number of data points $N$ is equal to $\frac{N}{\sqrt{N-1}}$.
- The maximum objective value of **P1** is $\sqrt{N(N-1)}$ if $N$ is odd, $N$ if $N$ is even.

Theorem 1 and Theorem 2 follow from Lemma 6 and 5 respectively. Because the optimal solution of **P1** could be scaled so that their variance would be equal to $C$ in **P**, and a constant can be added to each point so that all data points will be positive. This linear transformation allows us to find the optimal solution of **P**.

**Theorem 3.** *The growth rate of the ratio of maximum objective function value to minimum objective function value using scheduling rule SD-MAD grows is* $\mathcal{O}(\sqrt{N})$.

*Proof.* The optimal objective values of **P** and **P1** only differs by a scalar ($C_p$ depending on the standard deviation of procedure type $p$). Hence we bound the ratio of **P** instead. There are only two possible orderings. The optimal objective function $z^*$ is less than or equal to the upper bounds of objective values of any given sequence and also greater than the smallest lower bound calculated.

$$\min_p \left\{ \frac{N_p}{\sqrt{N_p - 1}} \right\} \leq z^*_{P1} \leq \min_p N_p$$

If the variances of both procedures are same, SD-MAD would be indifferent between possible orderings. So

$$\min_p \left\{ \frac{N_p}{\sqrt{N_p - 1}} \right\} \leq z^h_{P1} \leq \min_p N_p$$

This means the ratio of the maximum value to the minimum value of **P** grows at rate $\mathcal{O}(\sqrt{N})$. □

### 2 Procedures with any Penalties

If the penalties of being tardy and early are different, the objective is:

$$z^* = \min_{D_1} c_e E[\max(0, D_1 - C_1)] + c_t E[\max(0, C_1 - D_1)]$$

After applying the SD-MAD algorithm, the resulting objective function value where $k = \lfloor \frac{c_t}{c_t+c_e} N \rfloor$ is:

$$\frac{1}{N}\sum_{i=1}^{k} c_e \left(D_1 - x_1^i\right) + \frac{1}{N}\sum_{i=k}^{N} c_t \left(x_1^i - D_1\right)$$

**Lemma 7.** *For a given sequence, Step 2 of the SD-MAD algorithm determines the optimal start times of procedures: If there are only two procedures, the scheduled start time of the first procedure is 0 and the scheduled start time of the second procedure is $c_t$-th $(c_t + c_e)$-quantile of first procedure's duration.*

*Proof.* Let $x_1^{(1)} \leq x_1^{(2)} \leq \cdots \leq x_1^{(N_1)}$ and for any $D_1 \leq x_1^{(1)}$, as $D_1$ increases each term in the objective's summation decreases. For any $D_1$ greater than $x_1^{(1)}$ and less than $x_1^{(2)}$ as $D_1$ increases only the first term of the summation increases by an amount proportional to $c_e$ while each of the rest $N_1 - 1$ terms decreases by an amount proportional to $c_t$. This means the total decrease in the objective is greater than the increase. $D_1$ can be increased up to $x_1^{(\lceil N_1 \cdot c_t/(c_e+c_t) \rceil)}$ so that the amount of decrease in objective function value is still greater than the amount of increase increase. Increasing $D_1$ above $x_1^{(\lceil N_1 \cdot c_t/(c_e+c_t) \rceil)}$ only increases the value of the objective.

□

Again we consider a setting which we have procedures with equal standard deviation. Among all such set of procedures we would like to find two classes of examples maximizing and minimizing the objective. We formulate a mathematical programming model where the decision variables, $\mathbf{z} = \{z_1, \ldots, z_N\}$, are the sample data points, $m$ denotes their median and $\mu$ is their mean.

$$\begin{aligned} \mathbf{P}_{\min} \quad \min \quad & \frac{1}{N}\sum_{i=1}^{k} c_e \left(D_1 - z_i\right) + \frac{1}{N}\sum_{i=k}^{N} c_t \left(z_i - D_1\right) \\ \text{s.t.} \quad & \frac{\sum_{i=1}^{N} z_i^2}{N} - \left(\frac{\sum_{i=1}^{N} z_i}{N}\right)^2 = C^2 \\ & zi \geq 0 \quad \forall i \end{aligned}$$

$$\begin{aligned} \mathbf{P}_{\max} \quad \max \quad & \frac{1}{N}\sum_{i=1}^{k} c_e \left(D_1 - z_i\right) + \frac{1}{N}\sum_{i=k}^{N} c_t \left(z_i - D_1\right) \\ \text{s.t.} \quad & \frac{\sum_{i=1}^{N} z_i^2}{N} - \left(\frac{\sum_{i=1}^{N} z_i}{N}\right)^2 = C^2 \end{aligned}$$

$$zi \geq 0 \quad \forall i$$

**Theorem 4.** *Given $a \in \mathbb{R}_{\geq 0}$ there exist $b \in \mathbb{R}_{\geq 0}$ such that the optimal procedure times maximizing the objective is in the following form: $(z_1 = a, \ldots, z_{N-k} = a, z_{N-k+1} = b, \ldots, z_N = b)$ where $k = \lfloor \frac{c_t}{c_t+c_e} N \rfloor$, and the variance constraint (first constraint in (1.13)) is met.*

**Theorem 5.** *Given $c \in \mathbb{R}_{\geq 0}$ there exist $d \in \mathbb{R}_{\geq 0}$ such that the optimal procedure times minimizing the objective is in form of: $(z_1 = c, \ldots, z_{N-1} = c, z_N = d)$, and the variance constraint (first constraint in (1.13)) is met.*

To prove this theorem we follow the same steps in previous section. We use the same approach: transform the problem such that $k = \lfloor \frac{c_t}{c_t+c_e} N \rfloor$ of the data points $(y_i)$ will be non-positive and the rest $(w_i)$ non-negative so that the inverse of the empirical distribution computed at the critical fractile $(c_t/(c_e + c_t))$ is 0. The objective:

$$\ell(w, y) = \sum_{i=1}^{N-k} c_t w_i - \sum_{i=1}^{k} c_e y_i$$

The variance of the data points can be calculated same as before:

$$G^2(w, y) = \frac{\sum_{i=1}^{N-k} w_i^2 + \sum_{i=1}^{k} y_i^2}{N} - \left( \frac{\sum_{i=1}^{N-k} w_i + \sum_{i=1}^{k} y_i}{N} \right)^2$$

The model becomes similar to the model in the previous section:

$$\begin{aligned} \mathbf{P1}_{\min} \quad \min \quad & \ell(w, y) \\ \text{s.t.} \quad & G^2(w, y) = 1 \\ & w_i \geq 0 \quad \forall i \in [1, \ldots, N-k] \\ & y_i \leq 0 \quad \forall i \in [1, \ldots, k] \end{aligned} \qquad \begin{aligned} \mathbf{P1}_{\max} \quad \max \quad & \ell(w, y) \\ \text{s.t.} \quad & G^2(w, y) = 1 \\ & w_i \geq 0 \quad \forall i \in [1, \ldots, N-k] \\ & y_i \leq 0 \quad \forall i \in [1, \ldots, k] \end{aligned}$$

Since **P1** is not convex, we solve **P2** which is:

$$\begin{aligned} \mathbf{P2}_{\max} \quad \max \quad & G^2(w, y) \\ \text{s.t.} \quad & \ell(w, y) = 1 \\ & w_i \geq 0 \quad \forall i \in [1, \ldots, N-k] \\ & y_i \leq 0 \quad \forall i \in [1, \ldots, k] \end{aligned} \qquad \begin{aligned} \mathbf{P2}_{\min} \quad \min \quad & G^2(w, y) \\ \text{s.t.} \quad & \ell(w, y) = 1 \\ & w_i \geq 0 \quad \forall i \in [1, \ldots, N-k] \\ & y_i \leq 0 \quad \forall i \in [1, \ldots, k] \end{aligned}$$

Although there is a minor change in the objective function, Lemma 4 can be proven following the same steps. So instead of solving P1, an easier model, P2 can be solved to get the optimal solution of P1.

**Lemma 8.** ***P1*** *is maximized by $w_i$'s being equal to each other, $y_i$'s being equal to each other and their values being determined by the first constraint of* ***P1****.*

*Proof.* The global minimum of **P2** should satisfy the KKT conditions below:
**KKT Conditions of P2 (minimization):**

1.

$$\begin{aligned}
&\sum_{i=1}^{N-k} c_t w_i - \sum_{i=1}^{k} c_e y_i = 1 \\
&-w_i \leq 0 \qquad \forall i \in \{1, \ldots, N-k\} \\
&y_i \leq 0 \qquad \forall i \in \{1, \ldots, k\}
\end{aligned}$$

2.

$$\begin{bmatrix} 2w_1 - 2\left(\frac{\sum_{i=1}^{N-k} w_i + \sum_{i=1}^{k} y_i}{N}\right) \\ \vdots \\ 2w_{N-k} - 2\left(\frac{\sum_{i=1}^{N-k} w_i + \sum_{i=1}^{k} y_i}{N}\right) \\ 2y_1 - 2\left(\frac{\sum_{i=1}^{N-k} w_i + \sum_{i=1}^{k} y_i}{N}\right) \\ \vdots \\ 2y_k - 2\left(\frac{\sum_{i=1}^{N-k} w_i + \sum_{i=1}^{k} y_i}{N}\right) \end{bmatrix} + \mu \begin{bmatrix} c_t \\ \vdots \\ c_t \\ c_e \\ \vdots \\ c_e \end{bmatrix} + \begin{bmatrix} -\lambda_1 \\ \vdots \\ -\lambda_k \\ \gamma_1 \\ \vdots \\ \gamma_k \end{bmatrix} = 0$$

3.

$$\begin{aligned}
&\lambda_i \geq 0 \qquad \forall i \in \{1, \ldots, k\} \\
&\gamma_i \geq 0 \qquad \forall i \in \{1, \ldots, k\}
\end{aligned}$$

4.

$$\begin{aligned}
&\lambda_i w_i = 0 \qquad \forall i \in \{1, \ldots, k\} \\
&\gamma_i y_i = 0 \qquad \forall i \in \{1, \ldots, k\}
\end{aligned}$$

If $N$ is an integer multiple of $(c_t + c_e)$ , $(w_1 = \frac{c_t+c_e}{2c_t c_e N}, \ldots, w_{N-k} = \frac{c_t+c_e}{2c_t c_e N}, y_1 = -\frac{c_t+c_e}{2c_t c_e N}, \ldots, y_k = -\frac{c_t+c_e}{2c_t c_e N})$ satisfies the KKT conditions. Otherwise $w_i$'s are equal to each other, $y_i$'s are equal to each other and their values are determined by the first constraint of **P2**.

After scaling the minimizer of **P2**, the maximizer of **P1** can be found. In the optimal solution of **P1** $w_i$'s being equal to each other, $y_i$'s being equal to each other and their values being determined by the first constraint of **P1**. □

**Lemma 9.** $\left(\frac{N}{\sqrt{N-1}}, 0, \ldots, 0\right)$ *is the minimizer of* ***P1****.*

*Proof.* The objective function of **P2** is convex and the constraints are linear. Global maximum is at the extreme points of the new **P2** which are in the form of $(1/c_e, 0, \ldots, 0)$ or $(0, \ldots, 0, 1/c_t)$. After scaling $(1/c_e, 0, \ldots, 0)$ to satisfy the first constraint of **P1**, $\left(\frac{N}{\sqrt{N-1}}, 0, \ldots, 0\right)$ is found to be the minimizer of **P1**.

□

It is important to point out:

- The minimum objective value of **P1** for a fixed number of data points $N$ is equal to $K_1 \frac{\max\{c_t, c_e\}N}{\sqrt{N-1}}$.
- The maximum objective value of **P1** is $K_2 \max\{c_t, c_e\}N$.

where $K$'s are constants to make sure the variance of the data points is equal to the desired value. The ratio of the maximum value to the minimum value grows at a rate $\mathcal{O}(\sqrt{N})$.

Theorem 4 and Theorem 5 follow from Lemma 8 and Lemma 9 respectively as explained in the previous section.

**Theorem 6.** *The growth rate of the ratio of maximum objective function value to minimum objective function value using scheduling rule SD-MAD grows is* $\mathcal{O}(\sqrt{N})$*.*

### 1.3.5 Worst Case Given the Variance

**Theorem 7.** *Given the variance,* $C^2$*, and the number of data points,* $N$*, taken from a sample, the maximum value empirical distribution's range can take is equal to* $\sqrt{2NC^2}$*.*

This setting can be formulated as a mathematical program model where the decision variables, $\mathbf{z} = \{z_1, \ldots, z_N\}$, are the sample data points, $m$ denotes their median and $\mu$ is their mean.

$$\begin{aligned} \mathbf{P}_{\max} \quad \max \quad & \max(z_i) - \min(z_i) \\ \text{s.t.} \quad & \frac{\sum_{i=1}^{N} z_i^2}{N} - \left(\frac{\sum_{i=1}^{N} z_i}{N}\right)^2 = C^2 \\ & zi \geq 0 \quad \forall i \end{aligned}$$

For simplicity, WLOG we assume the median is equal to 0, half of the data points ($w$) would be non-negative and the rest ($y$) would be non-positive as before. After adding constraints which enfoce the model to make $w_1$ being the largest and $y_1$ being the smallest data points,

the objective can be rewritten as the following, where $k = \lfloor N/2 \rfloor$, $N$ is the number of data points:

$$\ell(w, y) = w_1 - y_1$$

The variance of the data points can be calculated as:

$$G^2(w, y) = \frac{\sum_{i=1}^{k} w_i^2 + \sum_{i=1}^{k} y_i^2}{N} - \left( \frac{\sum_{i=1}^{k} w_i + \sum_{i=1}^{k} y_i}{N} \right)^2$$

Since the variance is equal to a constant, ($C^2$ (first constraint in (1.13))), without loss of generality that constant is set equal to 1.

$$\begin{aligned} \mathbf{P3}_{\max} \quad \max \quad & \ell(w, y) \\ \text{s.t.} \quad & G^2(w, y) = 1 \\ & w_1 \geq w_i \quad \forall i \in [2, \ldots, k] \\ & y_1 \leq y_i \quad \forall i \in [2, \ldots, k] \\ & w_i \geq 0 \quad \forall i \in [1, \ldots, k] \\ & y_i \leq 0 \quad \forall i \in [1, \ldots, k] \end{aligned}$$

$$\begin{aligned} \mathbf{P4}_{\min} \quad \min \quad & G^2(x, y) \\ \text{s.t.} \quad & \ell(x, y) = 1 \\ & w_1 \geq w_i \quad \forall i \in [2, \ldots, k] \\ & y_1 \leq y_i \quad \forall i \in [2, \ldots, k] \\ & x_i \geq 0 \quad \forall i \in [1, \ldots, k] \\ & y_i \leq 0 \quad \forall i \in [1, \ldots, k] \end{aligned}$$

**Lemma 10.** *After scaling the variables in the optimal solution of P4, the optimal solution of P3 will be obtained.*

*Proof.* This proof is identical to the proof of the Lemma 4. □

**Lemma 11.** $\left( \sqrt{\frac{N}{2}}, 0, \ldots, 0, -\sqrt{\frac{N}{2}} \right)$ *is the maximizer of* ***P3****.*

*Proof.* Solving **P4** is easier relative to solving **P3**, since the objective function of **P4** is convex and the constraints are linear. Since **P4** is convex, any local minimum is a global minimum. If the proposed point satisfies the KKT conditions, then it is the global minimum.

**KKT Conditions of $\mathbf{P4}_{\min}$:**

1.

$$w_1 - y_1 = 1$$

$$
\begin{aligned}
w_i - w_1 &\leq 0 \quad \forall i \in [2, \ldots, k] \\
y_1 - y_i &\leq 0 \quad \forall i \in [2, \ldots, k] \\
-w_i &\leq 0 \qquad \forall i \in \{1, \ldots, k\} \\
y_i &\leq 0 \qquad \forall i \in \{1, \ldots, k\}
\end{aligned}
$$

2.

$$
\begin{bmatrix}
2w_1 - 2\left(\frac{\sum_{i=1}^{k} w_i + \sum_{i=1}^{k} y_i}{N}\right) \\
\vdots \\
2w_k - 2\left(\frac{\sum_{i=1}^{k} w_i + \sum_{i=1}^{k} y_i}{N}\right) \\
2y_1 - 2\left(\frac{\sum_{i=1}^{k} w_i + \sum_{i=1}^{k} y_i}{N}\right) \\
\vdots \\
2y_k - 2\left(\frac{\sum_{i=1}^{k} w_i + \sum_{i=1}^{k} y_i}{N}\right)
\end{bmatrix}
+ \mu \begin{bmatrix} 1 \\ \vdots \\ 1 \\ -1 \\ \vdots \\ -1 \end{bmatrix}
+ \begin{bmatrix} -\lambda_1 \\ \vdots \\ -\lambda_k \\ \gamma_1 \\ \vdots \\ \gamma_k \end{bmatrix}
+ \begin{bmatrix} -\beta_2 - \beta_3 - \cdots - \beta_k \\ \beta_2 \\ \vdots \\ \beta_k \\ 0 \\ \vdots \\ 0 \end{bmatrix}
+
$$

$$
\begin{bmatrix} 0 \\ \vdots \\ 0 \\ \eta 2 + \eta_3 + \cdots + \eta_k \\ -\eta_2 \\ \vdots \\ -\eta_k \end{bmatrix} = 0
$$

3.

$$
\begin{aligned}
\beta_i &\geq 0 \qquad \forall i \in \{2, \ldots, k\} \\
\eta_i &\geq 0 \qquad \forall i \in \{2, \ldots, k\} \\
\lambda_i &\geq 0 \qquad \forall i \in \{1, \ldots, k\} \\
\gamma_i &\geq 0 \qquad \forall i \in \{1, \ldots, k\}
\end{aligned}
$$

4.

$$
\begin{aligned}
\beta_i(w_i - w_1) &= 0 \quad \forall i \in [2, \ldots, k] \\
\eta_i(y_1 - y_i) &= 0 \quad \forall i \in [2, \ldots, k] \\
\lambda_i w_i &= 0 \qquad \forall i \in \{1, \ldots, k\} \\
\gamma_i y_i &= 0 \qquad \forall i \in \{1, \ldots, k\}
\end{aligned}
$$

$(w_1 = 0.5, w_2 = 0, \ldots, w_k = 0, y_1 = -0.5, y_2 = 0, \ldots, y_k = 0)$ satisfies the KKT conditions. $(w_1 = 0.5, w_2 = 0, \ldots, w_k = 0, y_1 = -0.5, y_2 = 0, \ldots, y_k = 0)$ is the minimizer of **P4**. After the proper scaling, $\left(\sqrt{\frac{N}{2}}, 0, \ldots, 0, -\sqrt{\frac{N}{2}}\right)$ is the maximizer of **P3**.

□

Theorem 7 follows from Lemma 11. Because the optimal solution of **P3** could be scaled so that their variance would be equal to $C^2$ as in $\mathbf{P}_{\max}$. The range is equal to the maximum value minus the minimum value which is equal to $\sqrt{2NC^2}$.

# Chapter 2

# Data-Driven Appointment Scheduling

## Notation

| | |
|---|---|
| $D_j$ | Scheduled end time (due date) of $j^{th}$ task |
| $\Delta_j$ | Scheduled time allowance of $j^{th}$ task |
| $x_j$ | Random duration of $j^{th}$ task with density $f_j$ and cdf $F_j$ |
| $N_j$ | Number of observations of $j^{th}$ task |
| $\mathbf{x_j} = \{x_j^1, \ldots, x_j^{N_j}\}$ | Previous observations of duration of $j^{th}$ task |
| $C_j$ | Actual end (completion) time of $j^{th}$ task with density $f_{C_j}$ and cdf $F_{C_j}$ |
| $S_j$ | Actual start (appointment) time of $j^{th}$ task |
| $c_e$ | Penalty per unit time of earliness ($c_0$ or $\alpha$) |
| $c_t$ | Penalty per unit time of tardiness ($c_u$or $\beta$) |
| $E_j$ | Earliness of $j^{th}$ task |
| $T_j$ | Tardiness of $j^{th}$ task |
| $n$ | Number of tasks to be scheduled per day per room |

In this chapter, we first analyze the Appointment Scheduling Problem's objective function – the expected weighted earliness and tardiness computed over the empirical joint distribution of procedures. We study the continuity and convexity of the objective function and conditions under which there is an integral optimal schedule. Secondly, we briefly review methods for computing the optimizer given the sequence of the procedures. We also present approaches for constraining the search space containing the minimizer to facilitate online decision-making about the problem as new data points arrive. Lastly, we develop sequencing heuristics for the problem.

## 2.1 Properties of the Objective Function

$\mathbf{x_j} = \{x_j^1, \ldots, x_j^{N_j}\}$ represents the previous observations of duration of the procedure in the $j$'th position in the given sequence, where each element is a positive real number.

$\mathbf{C_j} = \{C_j^1, \ldots, C_j^{M_j}\}$, where $M_j = \prod_{i=1}^{j} N_i$, represents (the support of) possible values of the completion time of the procedure in $j$'th order, which is calculated using the duration of the previous procedures $(\mathbf{x_1}, \ldots, \mathbf{x_j})$ and their scheduled end times $(D_1, \ldots, D_j)$. $\mathbf{x_j}$ and $\mathbf{C_j}$ are multisets that do not necessarily have unique entries so that they allow multiple instances of elements unlike sets. Note that $\mathbf{x_1} = \mathbf{C_1}$. The objective function is

$$\sum_j E[\max\{c_e(D_j - C_j), c_t(C_j - D_j)\}] = \sum_{j=1}^{n} \frac{1}{M_j} \sum_{i=1}^{M_j} \max\{c_e(D_j - C_j^i), c_t(C_j^i - D_j)\} \quad (2.1)$$

where $\mathbf{C_1} = \mathbf{x_1}$, $\mathbf{C_j} = \{x_j^{k_j} + \max\{D_{j-1}, C_{j-1}^{\ell}\} : k_j \in \{1, \ldots, N_j\}, \ell \in \{1, \ldots, M_{j-1}\}\}$, $M_j = \prod_{\ell=1}^{j} N_\ell$ and $D_j$ is the scheduled end time of the $j$'th procedure and also the scheduled start time of $j+1$'st procedure. In other words there is no idle time scheduled between any two procedures.

**Theorem 8.** *The objective function (2.1) is a convex, continuous piecewise linear function.*

*Proof.* We prove the continuity first. We need two properties to prove the continuity of each term in the summation of the objective function.

- Let $f$ and $g$ be continuous functions and $h(x) = \max\{f(x), g(x)\}$. Suppose there is $x_0$ such that $f(x_0) = g(x_0)$. Given $\epsilon > 0$, we need to show that $|h(x) - h(x_0)| < \epsilon$ for $|x - x_0| < \delta$, according to the definition of the continuous function. Since $f$ and $g$ are continuous, given any $\epsilon$ the following needs to hold: $|f(x) - f(x_0)| < \epsilon$ provided $|x - x_0| < \delta_f$ and $|g(x) - g(x_0)| < \epsilon$ provided $|x - x_0| < \delta_g$. Since $h(x_0) = f(x_0) = g(x_0)$, for $\epsilon$ there exists some number $\delta$ such that $|x - x_0| < \delta$ satisfying $|h(x) - h(x_0)| < \epsilon$. So $h(x) = \max\{f(x), g(x)\}$ is a continuous function.

- Suppose $f$ and $g$ are functions such that $g$ is continuous at $x_0$, and $f$ is continuous at $g(x_0)$, $f(g(x))$ is continuous at $x_0$.

Each term in the summation is continuous function. The sum of a finite number of continuous functions is a continuous function.

Next we prove that the objective function is a piecewise linear function. Denton and Gupta (2003) reformulate the problem and revise the notation to prove the convexity. We use our notation: $d_j$ denotes the time allowance for the procedure $j$. We formulate $M_n = \prod_{\ell=1}^{n} N_\ell$ different scenarios such that each scenario $i$ has one observation from each procedure $(x_1^i, \ldots, x_n^i)$. $E_j^i$ and $T_j^i$ denote the earliness and the tardiness of the procedure $j$

under the scenario $i$ respectively.

$$
\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{j=1}^{n} \frac{1}{M_n} \left( \sum_{i=1}^{M_n} c_t T_j^i + \sum_{i=1}^{M_n} \frac{1}{M_i} c_e E_j^i \right) & \\
\text{subject to} & d_1 - E_1^i + T_1^i = x_1^i, & i = 1, \ldots, M_n \\
& d_2 - E_2^i - T_1^i + T_2^i = x_2^i, & i = 1, \ldots, M_n \\
& \qquad \vdots & \qquad \vdots \\
& d_n - E_n^i - T_{n-1}^i + T_n^i = x_n^i, & i = 1, \ldots, M_n \\
& d_j \geq 0 & j = 1, \ldots, n \\
& T_j^i \geq 0, E_j^i \geq 0 & j = 1, \ldots, n, i = 1, \ldots, M_n
\end{array}
\tag{2.2}
$$

Since the objective (2.1) can be translated into the linear programming model (2.2), this problem is convex. □

**Theorem 9.** *The objective function (2.1) is Lipschitz continuous.*

*Proof.* Eriksson et al. (2013) proved that the linear combination $c_1 f_1 + \cdots + c_n f_n$ of Lipschitz continuous functions, $f_1, \ldots, f_n$ on $I$ with Lipschitz constants $L_1, \ldots, L_n$ respectively, is Lipschitz continuous on $I$ with Lipschitz constant $|c_1|L_1 + \cdots + |c_n|L_n$. Following from this theorem, it is trivial to derive the Lipschitz constant of the objective function after computing the Lipschitz constant of each term in the summation.

$$
f^i(D_1, \ldots, D_n) \;=\; \frac{1}{M_j} \sum_{j=1}^{n} \max\{c_e(D_j - C_j^i), c_t(C_j^i - D_j)\}
$$

$f^i(.)$ is piecewise linear function and any continuous piecewise linear function is globally Lipschitz continuous. If the function is evaluated at any point $(D_1, \ldots, D_n)$, it will be in the form of $f^i(D_1, \ldots, D_n) = c_1 D_1 + \cdots + c_n D_n + K$ where $c_1, \ldots, c_n$ and $K$ are constants. By the definition of the function $-\max\{c_e, c_t\} \cdot n \leq c_1 \leq \max\{c_e, c_t\} \cdot n$, $-\max\{c_e, c_t\} \cdot (n-1) \leq c_2 \leq \max\{c_e, c_t\} \cdot (n-1)$,…,$-\max\{c_e, c_t\} \leq c_n \leq \max\{c_e, c_t\}$. If we choose any two points $D^1 = (D_1^1, \ldots, D_n^1)$, $D^2 = (D_1^2, \ldots, D_n^2)$ in the domain of the objective function, if there exists a positive real constant $\kappa$ such that:

$$
|f^i(D^1) - f^i(D^2)| \leq \kappa |D^1 - D^2|.
$$

the smallest value of $\kappa$ is equal to $\max\{c_e, c_t\} \cdot n$ which is equal to the Lipschitz constant of $f^i(.)$. The Lipschitz constant of $f(.) = \sum_{i=1}^{M_n} \frac{f_i(.)}{M_n}$ is also equal to $\max\{c_e, c_t\} \cdot n$.

□

**Theorem 10.** *There exist one integral optimal solution to the appointment scheduling problem with the objective (2.1).*

We need to show that the equivalent LP model has an integral optimal solution. We define three matrices such that $\mathbf{d}$ has the coefficients of $(d_1, \ldots, d_n)$, $\mathbf{E}$ contains the coefficients of $E_i^j$ for all $j \in \{1, \ldots, n\}$ and $i \in \{1, \ldots, M_n\}$ and $\mathbf{T}$ has the coefficients of $T_i^j$ for all $j \in \{1, \ldots, n\}$ and $i \in \{1, \ldots, M_n\}$.

$$\mathbf{d} = \begin{array}{c} \begin{matrix} d_1 & d_2 & d_3 & \cdots & d_n \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \end{array} \begin{matrix} \\ i = 1 \\ i = 1 \\ \vdots \\ i = 2 \\ i = 2 \\ \vdots \\ i = M_n \end{matrix}$$

$$\mathbf{E} = \begin{array}{c} \begin{matrix} E_1^1 & E_2^1 & E_3^1 & \cdots & E_1^2 & E_2^2 & E_3^2 & \cdots & E_n^{M_n} \end{matrix} \\ \begin{bmatrix} -1 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & -1 \end{bmatrix} \end{array} \begin{matrix} \\ i = 1 \\ i = 1 \\ \vdots \\ i = 2 \\ i = 2 \\ \vdots \\ i = M_n \end{matrix}$$

$$\mathbf{T} = \begin{array}{c} \begin{matrix} T_1^1 & T_2^1 & T_3^1 & T_4^1 & \cdots & T_1^2 & T_2^2 & T_3^2 & T_4^2 & \cdots & T_n^{M_n} \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ -1 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & -1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \end{array} \begin{matrix} \\ i = 1 \\ i = 1 \\ i = 1 \\ \vdots \\ i = 2 \\ i = 2 \\ i = 2 \\ \vdots \\ i = M_n \end{matrix}$$

After concatenating $\mathbf{d}$, $\mathbf{E}$ and $\mathbf{T}$ horizontally, the resulting matrix is the constraint matrix $\mathbf{A}$.

$$\mathbf{A} = [\mathbf{d}|\mathbf{E}|\mathbf{T}]$$

**Theorem 11.** *(Ghouila-Houri, 1962) An $m \times n$ integral matrix $\mathbf{A}$ with entries $a_{ij}$ is totally unimodular if and only if for each subset of the rows $R \subseteq \{1, \ldots, m\}$ there is a partition $R = R_1 \cup R_2$ such that*

$$\sum_{i \in R_1} a_{ij} - \sum_{i \in R_2} a_{ij} \in \{-1, 0, 1\} \quad \forall j \in \{1, \ldots, n\}.$$

**Remark.** *Since the the transpose of a totally unimodular matrix is also totally unimodular one can exchange the roles of the rows and the columns in Theorem 11.*

$$\sum_{j \in R_1} a_{ij} - \sum_{j \in R_2} a_{ij} \in \{-1, 0, 1\} \quad \forall i \in \{1, \ldots, m\}.$$

Theorem 10 follows from Theorem 11. First we assign the coefficients of the variables $d_j \in R$ alternatively to $R_1$ and $R_2$ in lexicographic order. Without loss of generality assume $d_k$ is the first one assigned to $R_1$, if $T_k \in R$, assign $T_k$ to $R_2$. $E_k$ is assigned to the opposite subset with respect to $T_{k-1}$. If $T_k$ is not in $R$, then the only thing to consider is that $T_{k-1}$ and $E_k$ should be in opposite subsets. Assume $d_{k+t}$ is the second in the lexicographic order, which is in $R_2$. $T_k + 1, \ldots, T_k + t - 1 \in R$ are assigned to $R_2$ and $T_{k+t} \in R$ to $R_1$. The main idea is to separate $d_j$ from $T_j$ and $T_{j-1}$ from $E_j$. Similar choices need be done for remaining variable coefficients.

After these assignments we find a partition of the LP-model (2.2) which satisfies the Theorem 11. Thus, we conclude that the constraint matrix of the LP-model (2.2) is totally unimodular.

## Complexity of the Objective Function Evaluation

The objective function is:

$$\sum_j E[\max\{c_e(D_j - C_j), c_t(C_j - D_j)\}] = \sum_{j=1}^{n} \frac{1}{M_j} \sum_{i=1}^{M_j} \max\{c_e(D_j - C_j^i), c_t(C_j^i - D_j)\} \quad (2.3)$$

where $\mathbf{C_1} = \mathbf{x_1}$, $\mathbf{C_j} = \{x_j^{k_j} + \max\{D_{j-1}, C_{j-1}^{\ell}\} : k_j \in \{1, \ldots, N_j\}, \ell \in \{1, \ldots, M_{j-1}\}\}$, $M_j = \prod_{\ell=1}^{j} N_\ell$. The complexity to compute the set $\mathbf{C_j}$ is $\mathcal{O}(M_j)$, and the complexity of computing the expected weighted earliness and tardiness ($\frac{1}{M_j} \sum_{i=1}^{M_j} \max\{c_e(D_j - C_j^i), c_t(C_j^i - D_j)\}$) is $\mathcal{O}(M_j)$ too. The complexity to evaluate sum of the expected weighted earliness and tardiness of all procedures is $\mathcal{O}(M_n)$.

As the number of observations of a procedure increases, the complexity to compute the objective function given the schedule, $D$, grows exponentially. Assuming the procedure duration data has only integer entries and $x_{max}$ is the maximum possible value of processing durations. Begen (2010) computes the objective function value of an integer schedule, $D$, in $\mathcal{O}(n^2 x_{max}^2)$ using recursive equations for the probability distributions of the start time,

completion time, tardiness and earliness of each procedure. He creates arrays of size $j \cdot x_{max}$ for all procedures $j \in \{1, \ldots, n\}$ which hold all possible integer values of completion time $C_j$ and start time $S_{j+1}$ and their associated probabilities. The objective function is computed by taking the expectation over the probability distribution of $C_j$'s.

| | |
|---|---|
| Set $C_1 = x_1$ | $\mathcal{O}(1)$ |
| Compute the start time $S_j = \max(D_j, C_{j-1})$ and $P(S_j = k)$ for each $k \in \{0, 1, \ldots, nx_{max}\}$ | $\mathcal{O}(nx_{max})$ |
| Compute the the distribution of $C_j$ by using the formula $P(C_j = k) = Prob(S_j = k - x_j)$ | $\mathcal{O}(nx_{max}^2)$ |
| Compute the distributions for all $n$ procedure | $\mathcal{O}(n^2 x_{max}^2)$ |

Table 2.1: Complexity of evaluating the objective function value at an integer schedule assuming the procedure duration data has only integer entries (Begen, 2010).

We can use a similar setting to compute the objective function value of any schedule without restricting $D = (D_1, \ldots, D_n)$ to have only integer entries assuming the procedure duration data has only integer entries. It is useful in some cases, for instance, if we are minimizing the objective function using an iterative descent algorithm. Scheduled end time of procedure $j$, $D_j$ may have a fractional part different than 0.0. The fractional parts of $D_j$ for all $j \in \{1, \ldots, n\}$ may be different. We create arrays of size $j^2 \cdot x_{max}$ for all procedures $j \in \{1, \ldots, n\}$ which holds values of completion time $C_j$ and start time $S_{j+1}$ (all possible integer values and all integer values plus fractional parts of the previous procedures' scheduled end times) and their associated probabilities.

| | |
|---|---|
| Set $C_1 = x_1$ | $\mathcal{O}(1)$ |
| Compute the start time $S_j = \max(D_j, C_{j-1})$ and $P(S_j = k)$ for each $k \in \{0, 1, \ldots, nx_{max}, f_1, 1 + f_1, \ldots, nx_{max} + f_1, \ldots\}$ where $f_j = D_j - \lfloor D_j \rfloor$ | $\mathcal{O}(n^2 x_{max})$ |
| Compute the the distribution of $C_j$ by using the formula $P(C_j = k) = Prob(S_j = k - x_j)$ | $\mathcal{O}(n^2 x_{max}^2)$ |
| Compute the distributions for all $n$ procedure | $\mathcal{O}(n^3 x_{max}^2)$ |

Table 2.2: Complexity of evaluating the objective function value at any schedule with fractional parts assuming the procedure duration data has only integer entries.

Depending on the number of observation of each procedure and the maximum procedure duration, we may choose the method to compute the objective function with minimum complexity ($\min\{\mathcal{O}(n^3 x_{max}^2), \mathcal{O}(M_n)\}$).

## 2.2 Computation of the Optimizer

The objective function (2.1) is a convex, continuous piecewise linear function. After reformulating the problem and revising the notation we formulate a linear programming model (2.2). Denton and Gupta (2003) modeled appointment scheduling as a two-stage stochastic linear program (2-SLP) and show that if the random procedure duration distributions have a finite support, decomposition algorithms (such as the L-shaped algorithm) are efficient solving large problem instances. But if the random procedure durations are assumed to be independent, the support of the joint distribution grows exponentially as new data points are observed. If the support grows exponentially, decomposition algorithms fail to solve large problem instances (Denton and Gupta, 2003). There are some approximation algorithms proposed to find a near optimal solution assuming the procedures are independent.

There are various methods to optimize convex, continuous piecewise linear functions, including the subgradient method. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function with domain $\mathbb{R}^n$. At each iteration, the subgradient method take a step:

$$x^{(k+1)} = x^{(k)} - \alpha_k g^{(k)}$$

where $g^{(k)}$ is a subgradient of $f$ at $x^{(k)}$ and $\alpha_k$ is the step size at the $k^{\text{th}}$ iteration. Since the Subgradient Method is not following a descent direction at all times, we need to keep track of the minimum solution so far.

$$f_{best}^{(k)} = \min\{f_{best}^{(k-1)}, f^(x^{(k)})\}$$

There are various possible step size rules: constant step size, constant step length, square summable but not summable, nonsummable diminishing. For constant step size and constant step length, the subgradient algorithm is guaranteed to converge to the minimum (Boyd et al. (2003)).

$$\lim_{k\to\infty} f_{best}^{(k)} - f^* < \epsilon$$

Subgradient methods are slower than gradient descent but if the function is convex the convergence rate is $\mathcal{O}(1/\epsilon^2)$.

The objective function (2.1) is not differentiable everywhere in its domain but it is convex function with finite optimal value and a minimizer and Lipschitz continuous with a constant $\kappa$. The subgradient algorithm is guaranteed to converge to the optimal for the diminishing step size and step length rules (Boyd et al., 2003).

There is another alternative to optimize convex, continuous piecewise linear functions which is smoothing the function and then apply a descent algorithm to find the minimum. Smoothing the objective function and applying the gradient descent method may not accelerate the convergence. But instead we may benefit from decent algorithms with momentum (Qian, 1999). In other words we can use gradient descent with momentum method to minimize the smoothed function. At each iteration instead of following the negative gradient of the

smoothed function, we can follow the momentum vector which is the discounted sum of the previous gradients, thus leading to faster convergence.

$$\begin{aligned} v^{(k+1)} &= \rho v^{(k)} + \nabla f(x^{(k)}) \\ x^{(k+1)} &= x^{(k)} - \alpha_k v^{(k+1)} \end{aligned}$$

This method is known as Heavy Ball Method, which is is usually attributed to Polyak (1964). The update equations can be rewritten as:

$$x^{(k+1)} = x^{(k)} - \alpha_k \nabla f(x^{(k)}) - \rho(x^{(k-1)} - x^{(k)})$$

There is a similar algorithm known as Nesterov's accelerated method (Nesterov, 1983) converges more rapidly than the Heavy Ball method for convex functions. Ruder (2016) says Nesterov's accelerated method calculates the gradient not with respect to the to current $x^{(k)}$ but with respect to the approximate future position $x^{(k)} - \rho v^{(k)}$:

$$\begin{aligned} v^{(k+1)} &= \rho v^{(k)} + \nabla f(x^{(k)} - \rho v^{(k)}) \\ x^{(k+1)} &= x^{(k)} - \alpha_k v^{(k+1)} \end{aligned}$$

### 2.2.1 Smoothing the Objective Funtion

We may try to smooth the function and try to run a descent algorithm with momentum converging faster than descent algorithm without momentum. LogSumExp is called smooth maximum, which is a smooth approximation to the maximum function. It is commonly used in Machine Learning Algorithms.

$$\mathrm{LSE}(\mathbf{x}) = \log\big(\textstyle\sum_i \exp x_i\big).$$

LSE is an approximation to the maximum function. The proof of LSE approximating the maximum function, follows from the First Order Taylor Expansion of LSE.

$$\log\big(\textstyle\sum_i \exp x_i\big) \approx \log(\exp x_j) + \big(\sum_{i\neq j} \exp x_i\big)/\exp x_j \approx x_j = \max_i x_i.$$

For the sake of simplicity we focus on the deterministic case. In other words we assume there is only one observation for each procedure. Given there is only one set of data for all procedures, after smooting the objective function (2.1), the new objective function becomes (2.4):
Known Parameters/ Data Points:

1. $x = (x_1, \ldots, x_n)$ are positive real numbers (most of the time they are integers.)

Variables:

1. $D = (D_1, \ldots, D_n)$

$c_e$ and $c_t$ are the fixed penalties for being early or late, and $C = (C_1, \ldots, C_n)$ is also defined below.

$$g(D_1, \ldots, D_n) = \sum_{j=1}^{n} \log\left(e^{c_e(D_j - C_j)} + e^{c_t(C_j - D_j)}\right) \tag{2.4}$$

where $C_1 = x_1$, $C_j = x_j + \log\{e^{D_{j-1}} + e^{C_{j-1}}\}$

**Example**

Assume $n = 3$ and $c_e = c_t$. After substituting the definition of $C_1, C_2, C_3$ into the objective function (2.4), the objective becomes:

$$\begin{aligned} g(D_1, D_2, D_3) \;=\; & \log\left(e^{(D_1 - x_1)} + e^{(x_1 - D_1)}\right) + \\ & \log\left(e^{(D_2 - x_2)}(e^{x_1} + e^{D_1})^{-1} + e^{(x_2 - D_2)}(e^{x_1} + e^{D_1})\right) + \\ & \log\left(e^{(D_3 - x_3)}(e^{x_1+x_2} + e^{D_1+x_2} + e^{D_2})^{-1} + e^{(x_3 - D_3)}(e^{x_1+x_2} + e^{D_1+x_2} + e^{D_2})\right) \end{aligned}$$

First we would like to find the schedules $(D_1, \ldots, D_n)$ satisfying the First Order Condition

$$\nabla g(D_1, \ldots, D_n) = 0$$

.

$$\begin{aligned} \frac{\partial g}{\partial D_1} \;=\; & \frac{e^{(D_1 - x_1)} - e^{(x_1 - D_1)}}{e^{(D_1 - x_1)} + e^{(x_1 - D_1)}} + \frac{-e^{(D_2 - x_2)}(e^{x_1} + e^{D_1})^{-2}e^{D_1} + e^{(x_2 - D_2)}e^{D_1}}{e^{(D_2 - x_2)}(e^{x_1} + e^{D_1})^{-1} + e^{(x_2 - D_2)}(e^{x_1} + e^{D_1})} + \\ & \frac{-e^{(D_3 - x_3)}(e^{x_1+x_2} + e^{D_1+x_2} + e^{D_2})^{-2}e^{(x_2+D_1)} + e^{(x_3 - D_3)}e^{(x_2+D_1)}}{e^{(D_3 - x_3)}(e^{x_1+x_2} + e^{D_1+x_2} + e^{D_2})^{-1} + e^{(x_3 - D_3)}(e^{x_1+x_2} + e^{D_1+x_2} + e^{D_2})} \end{aligned}$$

$$\begin{aligned} \frac{\partial g}{\partial D_2} \;=\; & \frac{e^{(D_2 - x_2)}(e^{x_1} + e^{D_1})^{-1} - e^{(x_2 - D_2)}(e^{D_1} + e^{x_1})}{e^{(D_2 - x_2)}(e^{x_1} + e^{D_1})^{-1} + e^{(x_2 - D_2)}(e^{x_1} + e^{D_1})} + \\ & \frac{-e^{(D_3 - x_3)}(e^{x_1+x_2} + e^{D_1+x_2} + e^{D_2})^{-2}e^{D_2} + e^{(x_3 - D_3)}e^{D_2}}{e^{(D_3 - x_3)}(e^{x_1+x_2} + e^{D_1+x_2} + e^{D_2})^{-1} + e^{(x_3 - D_3)}(e^{x_1+x_2} + e^{D_1+x_2} + e^{D_2})} \end{aligned}$$

$$\frac{\partial g}{\partial D_3} \;=\; \frac{e^{(D_3 - x_3)}(e^{x_1+x_2} + e^{D_1+x_2} + e^{D_2})^{-1} - e^{(x_3 - D_3)}(e^{x_1+x_2} + e^{D_1+x_2} + e^{D_2})}{e^{(D_3 - x_3)}(e^{x_1+x_2} + e^{D_1+x_2} + e^{D_2})^{-1} + e^{(x_3 - D_3)}(e^{x_1+x_2} + e^{D_1+x_2} + e^{D_2})}$$

After setting $\nabla g$ equal to 0 and solving the three equations above, we find only one solution which is $(D_1, D_2, D_3) = (x_1, x_1 + x_2 + \log(2), x_1 + x_2 + x_3 + \log(4))$. We need to show that this critical point is the global minimum.

**The New Objective Function's Convexity**

The next step is to generalize the solution for any number of procedures $n$ where $c_e = c_t$.

**Lemma 12.** *The function $h(y) = \log(e^y + e^{-y})$ is a convex function with minimum value* $\log(2)$.

*Proof.* $h(y)$ is twice-differentiable function of a single variable, $y$. The first derivative is

$$h'(y) = 1 - \frac{2e^{-y}}{e^y + e^{-y}}$$

with the root $y = 0$. The second derivative is

$$h''(y) = \frac{4}{(x^y + e^{-y})^2}$$

which is greater than 0. Since its second derivative is always non-negative on its entire domain, then the function is convex. The minimum value of the function is $h(0) = \log(2)$. □

**Lemma 13.** *The objective function is*

$$g(D_1, \ldots, D_n) = \sum_{j=1}^{n} g_j(D_1, \ldots, D_n) = \sum_{j=1}^{n} \log\left(e^{(D_j - C_j)} + e^{(C_j - D_j)}\right)$$

*where $c_e = c_t$, $C_1 = x_1$, $C_j = x_j + \log\{e^{D_{j-1}} + e^{C_{j-1}}\}$. The minimum of the function is $n\log(2)$ and attained on $(D_1, D_2, \ldots, D_n) = (x_1, x_1 + x_2 + \log(2), \ldots, \sum_{j=1}^{n} x_j + (n-1)\log(2))$.*

*Proof.* For any $j \in \{1, \ldots, n\}$, $g_j(D_1, \ldots, D_n) = \log\left(e^{(D_j - C_j)} + e^{(C_j - D_j)}\right)$. Define $y_j = D_j - C_j$ and redefine $g_j(y_j)$ to be equal to $\log\left(e^{y_j} + e^{-y_j}\right)$. Lemma 12 states that $g_j(y_j) \geq \log(2)$ and its minimum can be attained at $y_j = 0$. So if we can find a feasible point satisfying $y_j = D_j - C_j = 0$ for each $j$, it will be the global minimum of $g_j(D_1, \ldots, D_n)$. We would like to have $D_j = C_j$ for any $j$. We set $D_1$ equal to $x_1$. By its definition $C_1$ is also equal to $x_1$. $D_1 = C_1 = x_1$ holds. The definition of $C_2$ is $x_2 + \log\{e^{D_1} + e^{C_1}\}$. So

$$C_2 = x_2 + \log\{e^{x_1} + e^{x_1}\} = x_1 + x_2 + \log(2)$$

We choose $D_2$ equal to $C_2$ which is $x_1 + x_2 + \log(2)$. Recursively we can compute $C_j$'s and set corresponding $D_j$'s equal to $C_j$. So that:

$$C_n = D_n = x_n + \log\{e^{\sum_{j=1}^{n-1} x_j + (n-2)\log(2)} + e^{\sum_{j=1}^{n-1} x_j + (n-2)\log(2)}\} = \sum_{j=1}^{n} x_j + (n-1)\log(2)$$

□

**Lemma 14.** *The objective function is*

$$g(D_1,\dots,D_n)=\sum_{j=1}^{n} g_j(D_1,\dots,D_n)=\sum_{j=1}^{n}\log\left(e^{(D_j-C_j)}+e^{(C_j-D_j)}\right)$$

*where* $c_e=c_t$ *,* $C_1=x_1$*,* $C_j=x_j+\log\{e^{D_{j-1}}+e^{C_{j-1}}\}$*.* $g(D_1,\dots,D_n)$ *is convex.*

*Proof.* The sum differentiable functions is differentiable. So

$$g(D_1,\dots,D_n)=\sum_{j=1}^{n} g_j(D_1,\dots,D_n)=\sum_{j=1}^{n}\log\left(e^{(D_j-C_j)}+e^{(C_j-D_j)}\right)$$

is differentiable. We prove first that $g_j(D_1,\dots,D_n)$ is convex. We compute $\nabla g_j$:

$$\begin{aligned}
\frac{\partial g_j(D_1,\dots,D_n)}{\partial D_j} &= \frac{e^{(D_j-C_j)}-e^{(C_j-D_j)}}{e^{(D_j-C_j)}+e^{(C_j-D_j)}}\\
\frac{\partial g_j(D_1,\dots,D_n)}{\partial C_j} &= \frac{-e^{(D_j-C_j)}+e^{(C_j-D_j)}}{e^{(D_j-C_j)}+e^{(C_j-D_j)}}\\
\frac{\partial g_j(D_1,\dots,D_n)}{\partial D_k} &= \frac{\partial g_j(D_1,\dots,D_n)}{\partial C_j}\frac{\partial C_j}{\partial D_k} \quad \forall k\in\{1,\dots,j-1\}
\end{aligned}$$

After setting $\nabla g_j$ equal to 0 and solving the all equations, we find only one solution which is $D_j=C_j$. Lemma 13 states that it is the minimum, so $g_j$ is convex. The sum of convex functions is convex, so is $g(D_1,\dots,D_n)$. □

More general proof for any penalty costs $(c_e,c_t)$ can be done followinf similar steps.

We would like to bound the smoothed function (2.4) to estimate how much worse we perform if we replace the original objective function (2.1) with (2.4).

$$\begin{aligned}
\max\{y_1,\dots,y_m\} &= \log(\exp(y_1,\dots,y_m))\\
&< LSE(y_1,\dots,y_m)\\
&< \log(m\dot{\exp}(\max\{y_1,\cdots,y_m\}))\\
&= \max\{y_1,\dots,y_m\}+\log(m)
\end{aligned}$$

By using the upperbound stated above, we can bound $g(D_1,\dots,D_n)$. In order to compute the $g_j(D_1,\dots,D_n)$ we need to compute $C_j$ first. $C_j$ can be computed recursively with $j-1$ maximum functions starting from $C_1=x_1$. We also compare $C_j$ with $D_j$, which increases the number of maximum function used to compute $g_j(D_1,\dots,D_n)$ by one.

$$\begin{aligned}
g(D_1,\dots,D_n) &= \sum_{j=1}^{n}\log\left(e^{(D_j-C_j)}+e^{(C_j-D_j)}\right)\\
&\le \sum_{j=1}^{n}\max\{c_e(D_j-C_j),c_t(C_j-D_j)\}+j\cdot\log(2)
\end{aligned}$$

## 2.3 Online Algorithm

As new observations arrive, new terms will be added to the objective function (2.1). We would like to characterize changes in the optimal sequence and optimal start times as new observations of procedure durations arrive. Our goal is to develop an online algorithm that sequences and schedules the procedures as new data points occur.

Recall that the minimizer of the sum of convex functions is not necessarily in the convex hull of their minimizers.[1] Therefore, the optimal schedule after the addition of the new observations is not necessarily in the convex hull of the previous optimal schedule and the optimal schedule computed using only the new observations. For instance assume the optimal time allowance for the first procedure is 2 hours and the new observation of that procedure is 1 hour. In the updated optimal schedule, the optimal allowance for that procedure in the new schedule is not necessarily between 1 hour and 2 hours, which is counter-intuitive. However we can bound the region containing the optimal solution, and update this bound as new observations arrive.

We consider two alternative approaches for obtaining the empirical distribution of procedures times:

1. We can directly obtain the empirical joint distribution of procedure durations from the data or scenario analysis. Then, we assume the procedure durations are independent of the order in which they are processed. In other words we have a data matrix with $n$ columns for each procedure and $M_n$ rows for different observations of the joint distribution.

2. We can assume all procedures are independent of each other and also the order they are processed in. In this case, the empirical joint distribution of the procedure durations can be constructed from the Cartesian Product of $n$ sets of the observations, one from each of the $n$ procedures to be scheduled. Procedure $x_j$ for any $j \in \{1, \ldots, n\}$ has $N_j$ different observations. Thus, there are $M_n = \prod_{i=1}^{n} N_i$ possible values of the joint distribution. This alternative is a subset of the first alternative,because the joint empirical distribution can also be assumed as a set of scenarios. We also focus on this one.

### 2.3.1 New Notation

First we analyze the first alternative above which is more general. Given the data matrix of size $M_n \times n$, where $M_n$ is the number of scenarios and $n$ is the number of procedures to be scheduled, we obtain the empirical joint distribution. It is convenient to use $\Delta_j$, time allowances, instead of $D_j$, the start times. The objective function below is equivalent to the

[1] Kuwaranancharoen and Sundaram (2018) provides an example: $f_1(x, y) = x^2 - xy + \frac{1}{2}y^2$ and $f_2(x, y) = x^2 + xy + \frac{1}{2}y^2 - 4x - 2y$ have minimizers (0,0) and (2, 0) respectively, whose sum has minimizer (1, 1).

objective function (2.1) after the substitution to get:

$$f(\Delta_1,\ldots,\Delta_n)=\frac{1}{M_n}\sum_{i=1}^{M_n}\sum_{j=1}^{n}\max\left\{c_e\left(\sum_{k=1}^{j}\Delta_k-C_j^i\right),c_t\left(C_j^i-\sum_{k=1}^{j}\Delta_k\right)\right\}$$

where $\mathbf{C_1}=\mathbf{x_1}$, $\mathbf{C_j}=\{x_j^i+\max\{\sum_{k=1}^{j-1}\Delta_k,C_{j-1}^i\}:i\in\{1,\ldots,M_n\}\}$, $M_n=\prod_{\ell=1}^{n}N_\ell$. We also substitute the definition of $C_j$ into the function:

$$f(\Delta_1,\ldots,\Delta_n)=\frac{1}{M_n}\sum_{i=1}^{M_n}\sum_{j=1}^{n}\max\left\{c_e\left(\Delta_j-x_j^i-T_{j-1}^i\right),c_t\left(x_j^i+T_{j-1}^i-\Delta_j\right)\right\}\tag{2.5}$$

where $\mathbf{T_0}=\mathbf{0}$, $\mathbf{T_j}=\{\max\{x_j^i-\Delta_j+T_{j-1}^i,0\}:i\in\{1,\ldots,M_n\}\}$, $M_n=\prod_{\ell=1}^{n}N_\ell$.

As new data points are observed, the support of the joint empirical distribution of the procedure durations may get bigger. The expectation of the earliness and tardiness (2.5) might need to be recalculated given the updated support. More terms may be added to the summation in the objective function. Assuming there are $M^+$ more scenarios, the objective (2.5) is updated as:

$$\begin{aligned}f(\Delta_1,\ldots,\Delta_n) \;=\; & \frac{1}{M_n+M^+}\sum_{i=1}^{M_n}\sum_{j=1}^{n}\max\left\{c_e\left(\Delta_j-x_j^i-T_{j-1}^i\right),c_t\left(x_j^i+T_{j-1}^i-\Delta_j\right)\right\}+\\ & \frac{1}{M_n+M^+}\sum_{i=1}^{M^+}\sum_{j=1}^{n}\max\left\{c_e\left(\Delta_j-x_j^i-T_{j-1}^i\right),c_t\left(x_j^i+T_{j-1}^i-\Delta_j\right)\right\}\end{aligned}$$

### 2.3.2 Search Space

We would like to answer certain questions without additional computation, such as:

- Is there a clear sign that the optimal schedule has been changed after the arrival of new data points?
- If there is a deterministic procedure needs to be scheduled at $j^{\text{th}}$ position for any $j\in\{1,\ldots,n\}$ order, is its optimal time allowance in the optimal schedule equal to its deterministic duration? Or should can the optimal time allowance be larger to compensate for the tardiness of the previous scheduled procedures?

If we constrain the search space after the arrival of the data points, we may answer those questions. We also use the search space in the later sections on sequencing to draw some conclusions.

In this section first we constrain the search space containing the minimizer of the appointment scheduling problem. Secondly we adapt the results to the online setting. As new data points are observed, the optimal schedule might change. After observing new data points we must update the search region containing the minimizer.

**Remark.** *$T_1 = \max\{x_1 - \Delta_1, 0\}$ is a non-increasing function of $\Delta_1$. $T_2 = \max\{x_2 - \Delta_2 + T_1, 0\}$ stays the same or decreases as $\Delta_2$ or $\Delta_1$ increases. Thus $T_j^i = \max\{x_j^i - \Delta_j + T_{j-1}^i, 0\}$ is a non-increasing function of $(\Delta_1, \ldots, \Delta_j)$.*

**Lemma 15.** *The time allowances for all procedures ($\Delta_j$s for all $j \in \{1, \ldots, n\}$) are non-negative real numbers. In the optimal solution, the value of a procedure's time allowance cannot be less than its minimum observation. ($\Delta_j \geq \min(\{x_j^i\}_{i=1}^{M_n}) \quad \forall j \in \{1, \ldots, n\}$).*

*Proof.* Assume by contradiction that the minimizer of the objective function is $\Delta' = (\Delta'_1, \ldots, \Delta'_t, \ldots, d'_n)$ where $\Delta'_t \leq \min(\{x_j^i\}_{i=1}^{M_n})$ for any $t \in \{1, \ldots, n\}$. Increase $\Delta_t$ by a sufficiently small positive number $\epsilon$ , decrease $\Delta_{t+1}$ by the same $\epsilon$ and call this point $\Delta = (\Delta'_1, \ldots, \Delta'_t + \epsilon, \Delta'_{t+1} - \epsilon, \ldots, \Delta'_n)$. The tardiness of procedure $t$ at the new point, $\Delta$, is:

$$\begin{aligned} T_t^i(\Delta'_1, \ldots, \Delta'_t + \epsilon) =& \max\{x_j^t - \Delta_t - \epsilon + T_{j-1}^i, 0\} \\ =& \max\{x_j^t - \Delta_t + T_{j-1}^i, 0\} - \epsilon \\ =& T_t^i(\Delta') - \epsilon \quad \forall i, t \end{aligned}$$

The function evaluated at $\Delta = (\Delta'_1, \ldots, \Delta'_t + \epsilon, \Delta'_{t+1} - \epsilon, \ldots, \Delta'_n)$:

$$\begin{aligned} & M_n \cdot f(\Delta) \\ =& \sum_i \sum_{j=1}^{t-1} \max\left\{c_e\left(\Delta'_j - x_j^i - T_{j-1}^i(\Delta')\right), c_t\left(x_j^i + T_{j-1}^i(d') - \Delta'_j\right)\right\} + \\ & \sum_i \max\left\{c_e\left((\Delta'_t + \epsilon) - x_t^i - T_{t-1}^i(\Delta')\right), c_t\left(x_t^i + T_{t-1}^i(\Delta') - (\Delta'_t + \epsilon)\right)\right\} + \\ & \sum_i \max\left\{c_e\left((\Delta'_{t+1} - \epsilon) - x_{t+1}^i - (T_t^i(\Delta') - \epsilon)\right), c_t\left(x_{t+1}^i + (T_t^i(\Delta') - \epsilon) - (\Delta'_{t+1} - \epsilon)\right)\right\} + \\ & \sum_i \sum_{j=t+2}^{n} \max\left\{c_e\left(\Delta'_j - x_j^i - T_{j-1}^i(\Delta')\right), c_t\left(x_j^i + T_{j-1}^i(\Delta') - (\Delta'_j)\right)\right\} \\ =& \sum_i \sum_{j=1}^{t-1} \max\left\{c_e\left(\Delta'_j - x_j^i - T_{j-1}^i(\Delta')\right), c_t\left(x_j^i + T_{j-1}^i(\Delta') - \Delta'_j\right)\right\} + \\ & \sum_i c_t\left(x_t^i + (T_{t-1}^i(\Delta')) - (\Delta'_t + \epsilon)\right) + \\ & \sum_i \sum_{j=t+1}^{n} \max\left\{c_e\left(\Delta'_j - x_j^i - T_{j-1}^i(\Delta')\right), c_t\left(x_j^i + T_{j-1}^i(\Delta') - (\Delta'_j)\right)\right\} \\ =& M_n \cdot (f(\Delta') - c_t\epsilon) \\ \leq& M_n \cdot f(\Delta') \end{aligned}$$

Thus, the objective function value at $\Delta$ is lower than the objective function value at $\Delta'$ which is assumed to be the minimizer. That is what contradicting our initial assumption. □

**Lemma 16.** *The time allowances of all procedures ($\Delta_j$s for all $j \in \{1,\dots,n\}$) are non-negative real numbers. In the optimal solution, the value of a procedure's time allowance cannot be greater than its maximum observation. ($\Delta_j \leq \max(\{x_j^i\}_{i=1}^{M_n}) \quad \forall j \in \{1,\dots,n\}$).*

*Proof.* If there is only one procedure, the minimizing time allowance is the $\frac{c_t}{(c_t+c_e)}$-quantile of the procedure duration, so the minimizer is less than or equal to the maximum procedure duration. Thus, trivially this lemma holds for the case where $n = 1$.

For $n \geq 2$, assume by contradiction that the minimizer of the objective function is $\Delta' = (\Delta'_1,\dots,\Delta'_t,\dots,\Delta'_n)$ where $\Delta'_t \geq \max(\{x_j^i\}_{i=1}^{M_n})$ for any $t \in \{1,\dots,n\}$. We evaluate the objective function at the point $\Delta = (\Delta_1,\dots,\Delta_{t-1}+\epsilon,\Delta_t-\epsilon,\dots,\Delta_n)$, where $\epsilon$ is a sufficiently small positive real number. We separate the objective function into two components. The first component is the expected total earliness and tardiness over all the scenarios for which $x_{t-1}^i + T_{t-2}^i > \Delta'_{t-1}$. Denote this set by $A$. Then $f_A(\Delta'_1,\dots,\Delta'_{t-1}+\epsilon,\Delta'_t-\epsilon,\dots,\Delta'_n)$ is equal to:

$$
\begin{aligned}
f_A(.) =& \sum_{i\in A}\sum_{j=1}^{t-2} \max\left\{c_e\left(\Delta'_j - x_j^i - T_{j-1}^i(\Delta')\right), c_t\left(x_j^i + T_{j-1}^i(\Delta') - \Delta'_j\right)\right\} + \\
& \sum_{i\in A} \max\left\{c_e\left((\Delta'_{t-1}+\epsilon) - x_{t-1}^i - T_{t-2}^i(\Delta')\right), c_t\left(x_{t-1}^i + T_{t-2}^i(\Delta') - (\Delta'_{t-1}+\epsilon)\right)\right\} + \\
& \sum_{i\in A} \max\left\{c_e\left((\Delta'_t-\epsilon) - x_t^i - (T_{t-1}^i(\Delta')-\epsilon)\right), c_t\left(x_t^i + (T_{t-1}^i(\Delta')-\epsilon) - (\Delta'_t-\epsilon)\right)\right\} + \\
& \sum_{i\in A}\sum_{j=t+1}^{n} \max\left\{c_e\left(\Delta'_j - x_j^i - T_{j-1}^i(\Delta')\right), c_t\left(x_j^i + T_{j-1}^i(\Delta') - (\Delta'_j)\right)\right\} \\
=& \sum_{i\in A}\sum_{j=1}^{t-2} \max\left\{c_e\left(\Delta'_j - x_j^i - T_{j-1}^i(\Delta')\right), c_t\left(x_j^i + T_{j-1}^i(\Delta') - \Delta'_j\right)\right\} + \\
& \sum_{i\in A} c_t\left(x_{t-1}^i + T_{t-2}^i(\Delta') - \Delta'_{t-1} - \epsilon\right) + \\
& \sum_{i\in A} \max\left\{c_e\left(\Delta'_t - x_t^i - T_{t-1}^i(\Delta')\right), c_t\left(x_t^i + T_{t-1}^i(\Delta') - \Delta'_t\right)\right\} + \\
& \sum_{i\in A}\sum_{j=t+1}^{n} \max\left\{c_e\left(\Delta'_j - x_j^i - T_{j-1}^i(\Delta')\right), c_t\left(x_j^i + T_{j-1}^i(\Delta') - \Delta'_j\right)\right\} \\
\leq& \sum_{i\in A}\sum_{j=1}^{t-2} \max\left\{c_e\left(\Delta'_j - x_j^i - T_{j-1}^i(\Delta')\right), c_t\left(x_j^i + T_{j-1}^i(\Delta') - \Delta'_j\right)\right\} + \\
& \sum_{i\in A} c_t\left(x_{t-1}^i + T_{t-2}^i(\Delta') - \Delta'_{t-1}\right) + \\
& \sum_{i\in A}\sum_{j=t}^{n} \max\left\{c_e\left(\Delta'_j - x_j^i - T_{j-1}^i(\Delta')\right), c_t\left(x_j^i + T_{j-1}^i(\Delta') - (\Delta'_j)\right)\right\}
\end{aligned}
$$

$$=f_A(\Delta')$$

The second component of the objective function consists of the expected total earliness and tardiness over all the remaining scenarios where $x^i_{t-1} + T^i_{t-2} \leq d'_{t-1}$ and $\epsilon \geq 0$. Denote this set by $B$. $f_B(\Delta'_1, \ldots, \Delta'_{t-1} + \epsilon, \Delta'_t - \epsilon, \ldots, \Delta'_n)$ is equal to:

$$\begin{aligned}
f_B(.) =& \sum_{i\in B}\sum_{j=1}^{t-2} \max\left\{c_e\left(\Delta'_j - x^i_j - T^i_{j-1}(\Delta')\right), c_t\left(x^i_j + T^i_{j-1}(\Delta') - \Delta'_j\right)\right\} + \\
& \sum_{i\in B} \max\left\{c_e\left((\Delta'_{t-1}+\epsilon) - x^i_{t-1} - T^i_{t-2}(\Delta')\right), c_t\left(x^i_{t-1} + T^i_{t-2}(\Delta') - (\Delta'_{t-1}+\epsilon)\right)\right\} + \\
& \sum_{i\in B} \max\left\{c_e\left((\Delta'_t-\epsilon) - x^i_t - (T^i_{t-1}(\Delta')-\epsilon)\right), c_t\left(x^i_t + (T^i_{t-1}(\Delta')-\epsilon) - (\Delta'_t-\epsilon)\right)\right\} + \\
& \sum_{i\in A}\sum_{j=t+1}^{n} \max\left\{c_e\left(\Delta'_j - x^i_j - T^i_{j-1}(\Delta')\right), c_t\left(x^i_j + T^i_{j-1}(\Delta') - (\Delta'_j)\right)\right\} \\
=& \sum_{i\in B}\sum_{j=1}^{t-2} \max\left\{c_e\left(\Delta'_j - x^i_j - T^i_{j-1}(\Delta')\right), c_t\left(x^i_j + T^i_{j-1}(\Delta') - \Delta'_j\right)\right\} + \\
& \sum_{i\in B} c_e\left(\Delta'_{t-1} + \epsilon - x^i_{t-1} - T^i_{t-2}(\Delta')\right) + \\
& \sum_{i\in B} c_e\left(\Delta'_t - \epsilon - x^i_t - 0\right) + \\
& \sum_{i\in B}\sum_{j=t+1}^{n} \max\left\{c_e\left(\Delta'_j - x^i_j - T^i_{j-1}(\Delta')\right), c_t\left(x^i_j + T^i_{j-1}(\Delta') - (\Delta'_j)\right)\right\} \\
=& f_B(\Delta')
\end{aligned}$$

$$\begin{aligned}
M_n \cdot f(&\Delta'_1, \ldots, \Delta'_{t-1} + \epsilon, \Delta'_t - \epsilon, \ldots, \Delta'_n) \\
&= f_A(\Delta'_1, \ldots, \Delta'_{t-1} + \epsilon, \Delta'_t - \epsilon, \ldots, \Delta'_n) + f_B(\Delta'_1, \ldots, \Delta'_{t-1} + \epsilon, \Delta'_t - \epsilon, \ldots, \Delta'_n) \\
&\leq M_n \cdot f(\Delta'_1, \ldots, \Delta'_{t-1}, \Delta'_t, \ldots, \Delta'_n)
\end{aligned}$$

Since $f(\Delta_1, \ldots, \Delta_{t-1}, \Delta_t, \ldots, \Delta_n) \geq f(\Delta_1, \ldots, \Delta_{t-1} + \epsilon, \Delta_t - \epsilon, \ldots, \Delta_n)$, this contradicts with our initial assumption.

□

**Theorem 12.** *If there is a deterministic procedure that needs to be scheduled, the optimal time allowance for that procedure is equal to its constant duration.*

*Proof.* The proof follows directly from the Lemma 15 and Lemma 16.

□

### 2.3.3 Search Space When All Procedures are Independent

Our goal is to further reduce the search space based on our second alternative to obtain the joint empirical distribution which assumes all procedures are independent of each other. The empirical joint distribution of the procedure durations is constructed such that the support of the empirical distribution is the Cartesian Product of $n$ sets of the observations, one from each of the $n$ procedures to be scheduled. The objective function becomes:

$$f(\Delta_1,\ldots,\Delta_n)=\sum_{j=1}^{n}\frac{1}{M_j}\sum_{i=1}^{N_j}\sum_{k=1}^{M_{j-1}}\max\left\{c_e\left(\Delta_j-x_j^i-T_{j-1}^k\right),c_t\left(x_j^i+T_{j-1}^k-\Delta_j\right)\right\} \quad (2.6)$$

where $\mathbf{T_{j-1}}=\{\max\{x_{j-1}^{k_{j-1}}-\Delta_{j-1}+T_{j-2}^{\ell},0\}:k_{j-1}\in\{1,\ldots,N_{j-1}\},\ell\in\{1,\ldots,M_{j-2}\}\}$,$\mathbf{T_0}=\mathbf{0}$ and $M_j=\prod_{\ell=1}^{j}N_\ell$. For notational convenience we define the expected earliness and tardiness cost of procedure $j$ to be $f_j(\Delta_1,\ldots,\Delta_n)$. The objective function can be rewritten as the following:

$$f(\Delta_1,\ldots,\Delta_n)=\sum_{j=1}^{n}f_j(\Delta_1,\ldots,\Delta_n).$$

Assuming all procedures are independent of each other, the time allowance for each procedure in the optimal solution should be greater than or equal to $\frac{c_t}{(c_t+c_e)}$-quantile of its procedure duration data. In order to prove our claim we use the following Lemmas.

**Lemma 17.** *Given* $(\Delta_1,\ldots,\Delta_{p-1},\Delta_{p+1},\ldots,\Delta_n)$ *the minimizer of*

$$f_p(\Delta_p)=\frac{1}{M_p}\sum_{i=1}^{N_p}\sum_{k=1}^{M_{p-1}}\max\left\{c_e\left(\Delta_p-x_p^i-T_{p-1}^k\right),c_t\left(x_p^i+T_{p-1}^k-\Delta_p\right)\right\}$$

*is the* $\frac{c_t}{(c_t+c_e)}$*-quantile of* $\mathbf{x_p}+\mathbf{T_{p-1}}$ *which is greater than or equal to the* $\frac{c_t}{(c_t+c_e)}$*-quantile of* $\mathbf{x_p}$.

*Proof.* The $\frac{c_t}{(c_t+c_e)}$-quantile of $\mathbf{x_p}+\mathbf{T_{p-1}}$ is greater than or equal to $\frac{c_t}{(c_t+c_e)}$-quantile of $\mathbf{x_p}$ because $\mathbf{T_{p-1}}$ has non-negative entries.

Define $\mathbf{y_p}=\mathbf{x_p}+\mathbf{T_{p-1}}$. Let $y_p^{(1)}\le y_p^{(2)}\le\cdots\le y_p^{(M_p)}$ and for any $\Delta_p\le y_p^{(1)}$, as $\Delta_p$ increases each term in the $f_p(\Delta_p)$'s summation decreases. For any $\Delta_p$ greater than $y_p^{(1)}$ and less than $y_p^{(2)}$, as $\Delta_p$ increases, only the first term of the summation increases by an amount proportional to $c_e$ while each of the remaining $M_p-1$ terms decrease by an amount proportional to $c_t$. Thus, the total decrease in the $f_p(\Delta_p)$ is greater than the increase. $\Delta_p$ can be increased up to $y_p^{(\lceil M_p\cdot c_t/(c_e+c_t)\rceil)}$ and the amount of decrease in $f_p(\Delta_p)$ value will still be greater than the amount of increase increase. Increasing $\Delta_p$ above $y_p^{(\lceil M_p\cdot c_t/(c_e+c_t)\rceil)}$ increases the value of $f_p(\Delta_p)$.

□

Define a function $g^p(\Delta_1, \ldots, \Delta_n)$ to be the expected earliness and tardiness cost of all procedures excluding the cost of procedure $p$.

$$
\begin{aligned}
& g^p(\Delta_1, \ldots, \Delta_n) \\
& \quad = \sum_{j \in \{1,\ldots,n\} \setminus \{p\}} \frac{1}{M_j} \sum_{i=1}^{N_j} \sum_{k=1}^{M_{j-1}} \max\left\{ c_e \left( \Delta_j - x_j^i - T_{j-1}^k \right), c_t \left( x_j^i + T_{j-1}^k - \Delta_j \right) \right\} \\
& \quad = f(\Delta_1, \ldots, \Delta_n) - \frac{1}{M_p} \sum_{i=1}^{N_p} \sum_{k=1}^{M_{p-1}} \max\left\{ c_e \left( \Delta_p - x_p^i - T_{p-1}^k \right), c_t \left( x_p^i + T_{p-1}^k - \Delta_p \right) \right\} \\
& \quad = \sum_{j \in \{1,\ldots,n\} \setminus \{p\}} f_j(\Delta_1, \ldots, \Delta_n)
\end{aligned}
$$

**Lemma 18.** *$g^1(\Delta_1, \ldots, \Delta_n)$ is a convex function with minimizers $(\Delta_1^*, \ldots, \Delta_n^*)$. The value of $\Delta_1^*$ is greater than or equal to $\max(\{x_1^i\}_{i=1}^{N_1})$.*

*Proof.* Assume that the first procedure is deterministic, $\mathbf{x_1} = \{\mathcal{K}_1\}$. The optimal time allowances are $(\mathcal{K}_1, \Delta_2^*, \ldots, \Delta_n^*)$ by Theorem (12). $f_1(\mathcal{K}_1, \Delta_2^*, \ldots, \Delta_n^*)$ is equal to zero.

$$
f(\mathcal{K}_1, \Delta_2^*, \ldots, \Delta_n^*) = \sum_{j \in \{2,\ldots,n\}} f_j(\mathcal{K}_1, \Delta_2^*, \ldots, \Delta_n^*) = g^1(\mathcal{K}_1, \Delta_2^*, \ldots, \Delta_n^*)
$$

Since the equation above holds, $(\mathcal{K}_1, \Delta_2^*, \ldots, \Delta_n^*)$ is the minimizer of $g^1(.)$ too. If we postpone the start of the second procedure by $\epsilon$, the optimal time allowances of the remaining procedures minimizing $g^1(.)$ does not change, stays at $(\mathcal{K}_1 + \epsilon, \Delta_2^*, \ldots, \Delta_n^*)$, because $\mathbf{T_1} = \{0\}$ for both cases.

$$
\begin{aligned}
g^1(\mathcal{K}_1, \Delta_2, \ldots, \Delta_n | \{\mathcal{K}_1\}, \mathbf{x_2}, \ldots, \mathbf{x_n}) &= g^1(\mathcal{K}_1 + \epsilon, \Delta_2, \ldots, \Delta_n | \{\mathcal{K}_1\}, \mathbf{x_2}, \ldots, \mathbf{x_n}) \quad \forall \epsilon \geq 0 \\
&= g^1(\epsilon, \Delta_2, \ldots, \Delta_n | \mathbf{0}, \mathbf{x_2}, \ldots, \mathbf{x_n}) \quad \forall \epsilon \geq 0 \\
&= g^1(0, \Delta_2, \ldots, \Delta_n | \mathbf{0}, \mathbf{x_2}, \ldots, \mathbf{x_n})
\end{aligned}
$$

This can be interpreted in the following way. Scheduling a deterministic procedure at the start of the horizon is equivalent to setting the start time of the remaining procedures to a later time instead of at time zero.

If there is a second observation of the first procedure, $\mathbf{x_1} = \{\mathcal{K}_1, \mathcal{K}_2\}$, $g^1(.)$ becomes:

$$
\begin{aligned}
& g^1(\Delta_1, \ldots, \Delta_n | \mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}) \\
& \qquad = \frac{1}{2} \left( g^1(\Delta_1, \ldots, \Delta_n | \{\mathcal{K}_1\}, \mathbf{x_2}, \ldots, \mathbf{x_n}) + g^1(\Delta_1, \ldots, \Delta_n | \{\mathcal{K}_2\}, \mathbf{x_2}, \ldots, \mathbf{x_n}) \right)
\end{aligned}
$$

Without loss of generality, assume $\mathcal{K}_2 = \mathcal{K}_1 + \epsilon$ and $\epsilon > 0$. Then $(\mathcal{K}_2, \Delta_2^*, \ldots, \Delta_n^*)$ is one of the minimizers of $g^1(\Delta_1, \ldots, \Delta_n | \{\mathcal{K}_2\}, \mathbf{x_2}, \ldots, \mathbf{x_n})$ and also $(\mathcal{K}_2, \Delta_2^*, \ldots, \Delta_n^*)$ is one of the minimizers of $(g^1(\Delta_1, \ldots, \Delta_n | \{\mathcal{K}_1\}, \mathbf{x_2}, \ldots, \mathbf{x_n})$.

The minimum possible value of $g^1(\Delta_1, \ldots, \Delta_n | \mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n})$ stays same as the number of observations increases, and the minimizer is $(\max(\{x_1^i\}_{i=1}^{N_1}), \Delta_2^*, \ldots, \Delta_n^*)$.

□

**Lemma 19.** *$g^p(\Delta_1, \ldots, \Delta_n)$ is a convex function with minimizers $(\Delta_1^*, \ldots, \Delta_n^*)$. The value of $\sum_{j=1}^{p} \Delta_j^* = D_p^*$ is greater than or equal to $\sum_{j=1}^{p} \max(\{x_j^i\}_{i=1}^{N_j})$.*

*Proof.* $g^p(\Delta_1, \ldots, \Delta_n)$ is a convex function because it can be formulated as a linear programming model similar to the LP in the proof of Theorem 8.

Compute the optimal time allowances minimizing the expected earliness and tardiness of the first $p-1$ procedures and denote the solution by $(\Delta_1^*, \ldots, \Delta_{p-1}^*)$. Compute the optimal time allowances minimizing the expected earliness and tardiness of the procedures $\{p+1, \ldots, n\}$ and denote the solution by $(\Delta_{p+1}^*, \ldots, \Delta_n^*)$. Following from the Lemma (18), if only procedures $\{p, \ldots, n\}$ are scheduled, the optimal time allowances minimizing the expected earliness and tardiness of the procedures $\{p+1, \ldots, n\}$ are $(\max(\{x_p^i\}_{i=1}^{N_p}) + \epsilon, \Delta_{p+1}^*, \ldots, \Delta_n^*)$, $\epsilon \geq 0$.

$$\begin{aligned} g^p(\Delta_1, \ldots, \Delta_n) &= \sum_{j \in \{1,\ldots,n\} \setminus \{p\}} f_j(\Delta_1, \ldots, \Delta_n) \\ &\geq \sum_{j=1}^{p-1} f_j(\Delta_1^*, \ldots, \Delta_{p-1}^*) + \sum_{j=p+1}^{n} f_j(\Delta_{p+1}^*, \ldots, \Delta_n^*) \end{aligned} \tag{2.7}$$

The equation (2.7) holds at equality if both sequences of procedures were independent of each other. Both sequences are independent when there is idle time in between, i.e. $\Delta_p \geq \sum_{j=1}^{p} \max(\{x_j^i\}_{i=1}^{N_j}) - \sum_{j=1}^{p-1} \Delta_j^*$. □

**Corollary 1.** *The minimum value of $g^p(\Delta_1, \ldots, \Delta_n)$ given $d_p = \mathcal{K}$ is non-increasing as $\mathcal{K} \to \mathcal{K} + \epsilon$ for any $\epsilon$ greater than 0.*

This corollary follows from the Lemma (19). If $\mathcal{K} \geq \sum_{j=1}^{p} \max(\{x_j^i\}_{i=1}^{N_j}) - \sum_{j=1}^{p-1} \Delta_j^*$, the minimum value of $g^p(\Delta_1, \ldots, \Delta_n)$ is equal to the global minimum. Because of the convexity of the function, when $\mathcal{K} < \sum_{j=1}^{p} \max(\{x_j^i\}_{i=1}^{N_j}) - \sum_{j=1}^{p-1} \Delta_j^*$, the minimum value of $g^p(\Delta_1, \ldots, \Delta_n)$ given $\Delta_p = \mathcal{K}$ decreases as $\mathcal{K}$ increases.

**Theorem 13.** *The time allowances for all procedures ($\Delta_j$s for all $j \in \{1, \ldots, n\}$) are non-negative real numbers. In the optimal solution, $\Delta_j$ cannot be less than the $\frac{c_t}{(c_t+c_e)}$-quantile of $\mathbf{x_j}$. ($\Delta_j \geq x_j^{(\lceil N_j \cdot c_t/(c_e+c_t) \rceil)} \quad \forall j \in \{1, \ldots, n\}$).*

*Proof.* This Theorem follows from Lemma (17) and the Corollary (1). □

### 2.3.4 Motivating Questions and Answers

In this section we ask some intuitive questions and find counterexamples to answer those questions. There are multiple questions about the optimal schedule and the sequences.

**Is the minimizer of the weighted sum of two expected earliness and tardiness over different data sets in the convex hull of the individual minimizers?**

1. Example

   First data set:

   $$\begin{aligned} x^1: \quad & x_1^1 = [27, 29, 26, 34, 27, 31, 28, 27, 26, 25, 60] \\ & x_2^1 = [26, 18, 25, 25, 20, 27, 23, 26, 23, 27, 24, 27, 20, 25, 28] \\ & x_3^1 = [4, 8, 6, 5, 3, 4, 3, 6, 7, 3, 8, 7, 6] \end{aligned}$$

   The minimizer of $f(.|x^1)$ is where $[D_1, D_2, D_3] = [29, 56, 62]$ or $[\Delta_1, \Delta_2, \Delta_3] = [29, 27, 6]$.

   Second data set:

   $$\begin{aligned} x^2: \quad & x_1^1 = [61, 27, 59, 39, 13, 25, 28, 18, 23, 32, 13] \\ & x_2^1 = [5, 44, 21, 39, 29, 11, 16, 21, 30, 12, 33, 39, 42] \\ & x_3^1 = [30, 35, 15, 43, 35, 32, 28, 35, 43] \end{aligned}$$

   The minimizer of $f(.|x^2)$ is where $[D_1, D_2, D_3] = [30, 69, 104]$, $[\Delta_1, \Delta_2, \Delta_3] = [30, 39, 35]$.

   The weighted sum is of the form:

   $$M_3^1 f(.|x^1) + M_3^2 f(.|x^2)$$

   where $M_{\dot{3}} = ||x_{\dot{1}}||_0 ||x_{\dot{2}}||_0 ||x_{\dot{3}}||_0 = N_{\dot{1}} N_{\dot{2}} N_{\dot{3}}$. The minimizer is $[D_1, D_2, D_3] = [28, 58, 85]$ or $[\Delta_1, \Delta_2, \Delta_3] = [28, 30, 27]$, which is not in the convex hull of the two minimizers above.

   Instead we concatenate two data sets to obtain the data set below, and compute the minimizer of the objective which is the expectation over the data set below.

   $$\begin{aligned} x: \quad & x_1 = [27, 29, 26, 34, 27, 31, 28, 27, 26, 25, 60, 61, 27, 59, 39, 13, 25, 28, 18, 23, 32, 13] \\ & x_2 = [26, 18, 25, 25, 20, 27, 23, 26, 23, 27, 24, 27, 20, 25, 28, 5, 44, 21, 39, 29, 11, 16, \\ & \qquad 21, 30, 12, 33, 39, 42] \\ & x_3 = [4, 8, 6, 5, 3, 4, 3, 6, 7, 3, 8, 7, 6, 30, 35, 15, 43, 35, 32, 28, 35, 43] \end{aligned}$$

   The minimizer is $[D_1, D_2, D_3] = [29, 57, 75]$ or $[\Delta_1, \Delta_2, \Delta_3] = [29, 28, 18]$.

2. Example

   Even if there is only one observation in the second data set, we still can find a counter example:

First data set:

$$x^1: \quad x_1^1 = [27, 29, 26, 34, 27]$$
$$x_2^1 = [26, 18, 25, 25, 20, 27, 23]$$
$$x_3^1 = [4, 8, 6]$$

The minimizer of $f(.|x^1)$ is where $[D_1, D_2, D_3] = [29, 54, 61]$ or $[\Delta_1, \Delta_2, \Delta_3] = [29, 25, 7]$.

Second data set:

$$x^2: \quad x_1^1 = [28]$$
$$x_2^1 = [26]$$
$$x_3^1 = [10]$$

The minimizer of $f(.|x^2)$ is where $[D_1, D_2, D_3] = [28, 54, 64]$ or $[\Delta_1, \Delta_2, \Delta_3] = [28, 26, 10]$.

The minimizer of the weighted sum is $[D_1, D_2, D_3] = [28, 54, 60]$ or $[\Delta_1, \Delta_2, \Delta_3] = [28, 26, 6]$, which is not in the convex hull of the two minimizers above.

**If the new observation drastically changes the variance of the duration data of that procedure, will the optimal sequence change?**

The initial data set is:

$x_A = [31, 27, 24, 28, 31, 29, 27, 30, 31, 30, 33, 30, 23, 22]$ $\quad Var(x_A) = 10.989$
$x_B = [21, 20, 23, 23, 28, 23, 28, 24, 22, 26, 21, 27, 25, 29, 30]$ $\quad Var(x_B) = 10.095$
$x_C = [28, 25, 19, 27, 32, 34, 23, 32, 25, 30, 27, 26, 30, 30, 31, 25]$ $\quad Var(x_C) = 15.133$

The optimal sequence is $A - B - C$ with the start times $(30, 55, 85)$ and the optimal objective value is 8.71786. A new observation of $x_A$ has arrived which is equal to 1. The variance of $x_A$ has been increased from 10.989 to 59.838 which is greater than the variances of the other procedure durations. The optimal order and the start times stay same, but the objective value is increased to 10.475.

**Only a procedure has a new data point. Is it possible that the order of the remaining procedures change after the addition of the new data point?**

The initial data set is:

$x_A = [86, 62, 113, 100, 75, 88, 51, 82, 72, 110, 73, 69, 98, 114, 58, 81, 101]$
$Var(x_A) = 375.596$
$x_B = [253, 235, 242, 233, 253, 258, 260, 239, 263, 235, 257, 257, 242]$
$Var(x_B) = 116.359$
$x_C = [39, 55, 43, 53, 51, 36, 47]$
$Var(x_C) = 52.238$
$x_D = [180, 162, 175, 185, 174, 183, 170, 185, 170, 184, 194, 184, 190, 186, 173]$
$Var(x_D) = 75.381$

The optimal sequence is $D - C - B - A$ with the time allowances $(183.0, 51.0, 253.0, 86.0)$ and objective value of 39.1326869209222. A new observation of $x_A$ has arrived which is

equal to 101. The new optimal sequence is $B - C - D - A$ with the time allowances $(253.0, 51.0, 184.0, 86.0)$ and objective value of 39.217908017908016. The order of procedures $B, C, D$ has changed.

If we schedule only $B, C, D$, the optimal sequence will be $C - D - B$ with optimal time allowances $(47.0, 184.0, 253.0)$.

**The optimal sequence and the associated optimal start times are known. After an addition of a data point to the set of one procedure's duration, given the previous optimal sequence the optimal start times stays same. Is this sequence still optimal since there is no change ?**

The initial data set is:

$$
\begin{aligned}
&x_A = [86, 62, 113, 100, 75, 88, 51, 82, 72, 110, 73, 69, 98, 114, 58, 81, 101] \\
&Var(x_A) = 375.596 \\
&x_B = [253, 235, 242, 233, 253, 258, 260, 239, 263, 235, 257, 257, 242] \\
&Var(x_B) = 116.359 \\
&x_C = [39, 55, 43, 53, 51, 36, 47] \\
&Var(x_C) = 52.238 \\
&x_D = [180, 162, 175, 185, 174, 183, 170, 185, 170, 184, 194, 184, 190, 186, 173] \\
&Var(x_D) = 75.381
\end{aligned}
$$

The optimal sequence is $D - C - B - A$ with the time allowances $(183.0, 51.0, 253.0, 86.0)$ and objective value of 39.1326869209222. A new observation of $x_D$ has arrived which is equal to 120. Given the sequence of $D - C - B - A$ the optimal time allowances are $(183.0, 51.0, 253.0, 86.0)$ which is same as before. The objective function value is 42.62487879767292. But the optimal sequence is $C - D - B - A$ and the optimal time allowances are $(47.0, 184.0, 253.0, 86.0)$. The objective function value is 42.491637039431154.

**If there is a new observation of any procedure, the minimum values of the objective function of all possible orderings increase or decrease altogether?**

This is not necessarily true. A counterexample:

The initial data set is:

$$
\begin{aligned}
&x_A = [57, 42, 41, 52, 39, 53, 45] \\
&Var(x_A) = 48.333 \\
&x_B = [249, 228, 252, 251, 253, 254, 248, 250, 253, 235, 240, 238, 247] \\
&Var(x_B) = 102.527 \\
&x_C = [116, 101, 107, 98, 112, 109, 92, 97, 93, 108, 102, 93, 107, 117, 116, 116, 104] \\
&Var(x_C) = 74.029 \\
&x_D = [177, 187, 172, 184, 193, 181, 190, 195, 176, 169, 189, 178, 185, 184, 194] \\
&Var(x_D) = 64.114
\end{aligned}
$$

If the new observation of $x_D$ is equal to one of $\{177, 178, 191, 192\}$ not all the minimum values of all orderings behave same. Some might increase while some decrease.

**Does the addition of a duration which is equal to the critical quantile of a procedure's duration change the optimal order?**

A counterexample can be created by adding multiple data points equal to median of the existing ones to the last procedure. The procedure's variability decreases so this procedure will be scheduled first after some number of additions of the same number to its data set.

### 2.3.5 Sequencing

The scheduler needs to make a decision about the sequence of the procedures also. Mancilla and Storer (2012) show that Sample Average Approximation (of expected earliness and tardiness) Appointment Sequencing and Scheduling Problem with only two scenarios is $\mathcal{NP}$-complete. Denton et al. (2007) presumes the appointment sequencing and scheduling problem minimizing the earliness and tardiness over a sample is $\mathcal{NP}$-Hard. Choi et al. (2019) show that the problem of single machine scheduling minimizing the sum of the earliness and tardiness of each job is $\mathcal{NP}$-Hard, which is the same problem. Instead of optimizing all possible $n!$ orderings, the appointment scheduling literature frequently turns to heuristics such as sequencing procedures in increasing standard deviation order. As some counterexamples in Section 2.3.4 indicate, ordering in increasing standard deviation is not necessarily optimal. In this section, we present heuristics tailored for our problem.

Theorem 13 states that the value of procedure $j$'s time allowance cannot be less than the $\frac{c_t}{(c_t+c_e)}$-quantile of $\mathbf{x_j}$ in the optimal solution. We bound the the objective function value at the schedule calculated by using the $\frac{c_t}{(c_t+c_e)}$-quantile of the procedures for each possible ordering of the procedures. Although the ordering with the minimum bound is not necessarily the best ordering, that ordering may be a good approximation to the optimal. We propose several heuristics with different complexities and performances. The first heuristic we propose computes an upper-bound on the objective function value calculated at the $\frac{c_t}{(c_t+c_e)}$-quantile of the procedures for any ordering and chooses the ordering with the lowest upper-bound at the schedule.

#### Heuristic 1

First we bound the objective function (2.6) using a triangle-like inequality.

$$f(\Delta_1, \ldots, \Delta_n) = \sum_{j=1}^{n} \frac{1}{M_j} \sum_{i=1}^{N_j} \sum_{k=1}^{M_{j-1}} \max\left\{c_e\left(\Delta_j - x_j^i - T_{j-1}^k\right), c_t\left(x_j^i + T_{j-1}^k - \Delta_j\right)\right\}$$

$$\leq \sum_{j=1}^{n} \frac{1}{M_j} \sum_{i=1}^{N_j} \sum_{k=1}^{M_{j-1}} \left[\max\left\{c_e\left(\Delta_j - x_j^i\right), c_t\left(x_j^i - \Delta_j\right)\right\} + \max\left\{c_e\left(T_{j-1}^k\right), c_t\left(T_{j-1}^k\right)\right\}\right]$$
$$= \sum_{j=1}^{n} \frac{1}{M_j} \sum_{i=1}^{N_j} \sum_{k=1}^{M_{j-1}} \left[\max\left\{c_e\left(\Delta_j - x_j^i\right), c_t\left(x_j^i - \Delta_j\right)\right\} + \max\left\{c_e, c_t\right\} T_{j-1}^k\right]$$

Define $\tau_{\mathbf{j}} = \{\max\{0, x_j^{k_j} - \Delta_j\} : k_j \in \{1, \ldots, N_j\}\}$. We assume the first schedule starts without any delay, which means $\mathbf{T_0} = \mathbf{0}$. The tardiness of the first procedure is $\mathbf{T_1} = \{\max\{x_1^{k_1} - \Delta_1, 0\} : k_1 \in \{1, \ldots, N_1\}\}$, which is the same set as $\tau_{\mathbf{j}}$. Each element in the sets of $\mathbf{T_2}, \ldots, \mathbf{T_n}$ can be bounded by elements in $\tau_{\mathbf{1}}, \ldots, \tau_{\mathbf{n}}$ recursively such that:

$$\begin{aligned}\mathbf{T_j} =& \{\max\{x_j^{k_j} - \Delta_j + T_{j-1}^{\ell}, 0\} \leq \max\{x_j^{k_j} - \Delta_j, 0\} + T_{j-1}^{\ell} : k_j \in \{1, \ldots, N_j\},\\ & \ell \in \{1, \ldots, M_{j-1}\}\}\\ =& \{\max\{x_j^{k_j} - \Delta_j + T_{j-1}^{\ell}, 0\} \leq \tau_j^{k_j} + T_{j-1}^{\ell} : k_j \in \{1, \ldots, N_j\}, \ell \in \{1, \ldots, M_{j-1}\}\}\end{aligned}$$

The elements in the set of $\mathbf{T_j}$ is bounded by the Cartesian sum of the elements in the sets of $\tau_{\mathbf{1}}, \ldots, \tau_{\mathbf{j}}$.

$$\tau_{\mathbf{j}} \leq \mathbf{T_j} \leq \tau_{\mathbf{1}} + \cdots + \tau_{\mathbf{j}} \tag{2.8}$$

By the definition of $\mathbf{T_j}$ each element in the set of $\mathbf{T_j}$ is bounded by the Cartesian sum of the elements in the sets of $\tau_{\mathbf{j}}$ and $\mathbf{T_{j-1}}$.

$$\mathbf{T_j} \leq +\tau_{\mathbf{j}} + \mathbf{T_{j-1}} \tag{2.9}$$

Since each element in the set of $\mathbf{T_j}$ is bounded by the sum of one element in each set, the sum of elements can be bounded as following:

$$\sum_{k=1}^{M_j} T_j^k \leq \sum_{i=1}^{j} \frac{M_j}{N_i} \sum_{k=1}^{N_i} \tau_i^k = \sum_{i=1}^{j} M_j \cdot \overline{\tau_i} \qquad M_j = \prod_{\ell=1}^{j} N_\ell \tag{2.10}$$

where $\overline{\tau_i} = \frac{1}{N_i} \sum_{m=1}^{N_i} \tau_i^m$ which is the average of the set $\tau_{\mathbf{i}}$ for any $i \in \{1, \ldots, n\}$.

The objective function can be bounded by:

$$\begin{aligned}& f(\Delta_1, \ldots, \Delta_n)\\ &\leq \sum_{j=1}^{n} \frac{1}{M_j} \sum_{i=1}^{N_j} \sum_{k=1}^{M_{j-1}} \left[\max\left\{c_e\left(\Delta_j - x_j^i\right), c_t\left(x_j^i - \Delta_j\right)\right\} + \max\left\{c_e, c_t\right\} T_{j-1}^k\right]\\ &= \sum_{j=1}^{n} \frac{1}{M_j} \sum_{i=1}^{N_j} \sum_{k=1}^{M_{j-1}} \max\left\{c_e\left(\Delta_j - x_j^i\right), c_t\left(x_j^i - \Delta_j\right)\right\} + \max\left\{c_e, c_t\right\} \sum_{j=2}^{n} \frac{N_j}{M_j} \sum_{k=1}^{M_{j-1}} T_{j-1}^k\end{aligned}$$

$$
\begin{aligned}
&\leq \sum_{j=1}^{n} \frac{1}{M_j} \sum_{i=1}^{N_j} \sum_{k=1}^{M_{j-1}} \max\left\{c_e\left(\Delta_j - x_j^i\right), c_t\left(x_j^i - \Delta_j\right)\right\} + \max\left\{c_e, c_t\right\} \sum_{j=2}^{n} \frac{1}{M_{j-1}} \sum_{i=1}^{j-1} M_{j-1}\overline{\tau_i} \\
&= \sum_{j=1}^{n} \frac{1}{M_j} \sum_{i=1}^{N_j} \sum_{k=1}^{M_{j-1}} \max\left\{c_e\left(\Delta_j - x_j^i\right), c_t\left(x_j^i - \Delta_j\right)\right\} + \max\left\{c_e, c_t\right\} \sum_{j=2}^{n} \sum_{i=1}^{j-1} \overline{\tau_i} \\
&= \sum_{j=1}^{n} \frac{1}{M_j} \sum_{i=1}^{N_j} \sum_{k=1}^{M_{j-1}} \max\left\{c_e\left(\Delta_j - x_j^i\right), c_t\left(x_j^i - \Delta_j\right)\right\} + \max\left\{c_e, c_t\right\} \sum_{j=1}^{n} (n-j)\overline{\tau_j}
\end{aligned}
$$

Given the schedule $\Delta_j = x_j^{(\lceil N_j \cdot c_t/(c_e+c_t)\rceil)}\ \forall j \in \{1,\ldots,n\}$, we may compute $\tau_{\mathbf{j}}$ and the average value of the set $\tau_{\mathbf{j}}$ for every $j \in \{1,\cdots,n\}$. For Heuristic 1, we then order procedures in increasing average value of the set $\tau_{\mathbf{j}}$, $\overline{\tau_j}$.

**Algorithm 1:** Sequencing Heuristic 1

1 Compute the schedule $\Delta_j = x_j^{(\lceil N_j \cdot c_t/(c_e+c_t)\rceil)}\ \forall j \in \{1,\ldots,n\}$.
2 Compute $\tau_{\mathbf{j}} = \{\max\{0, x_j^{k_j} - \Delta_j\} : k_j \in \{1,\ldots,N_j\}\}$ and $\overline{\tau_j} = \frac{1}{N_j}\sum_{k_j=1}^{N_j} \tau_j^{k_j}$ $\forall j \in \{1,\ldots,n\}$.
3 Order procedures in increasing $\overline{\tau_j}$.

### Bounding the Performance of Heuristic 1

We find an upper bound on the objective function (2.6) using a triangle-like inequality in Heuristic 1:

$$
\begin{aligned}
f(\Delta_1,\ldots,\Delta_n) &= \sum_{j=1}^{n} \frac{1}{M_j} \sum_{i=1}^{N_j} \sum_{k=1}^{M_{j-1}} \max\left\{c_e\left(\Delta_j - x_j^i - T_{j-1}^k\right), c_t\left(x_j^i + T_{j-1}^k - \Delta_j\right)\right\} \\
&\leq \sum_{j=1}^{n} \frac{1}{M_j} \sum_{i=1}^{N_j} \sum_{k=1}^{M_{j-1}} \left[\max\left\{c_e\left(\Delta_j - x_j^i\right), c_t\left(x_j^i - \Delta_j\right)\right\} + \max\left\{c_e, c_t\right\} T_{j-1}^k\right]
\end{aligned} \tag{2.11}
$$

We compute the difference, $\delta$, between the upper bound and the objective function, assuming $\mathbf{T_0} = \mathbf{0}$:

$$
\begin{aligned}
\delta \;=\; & \sum_{j=2}^{n} \frac{1}{M_j} \sum_{i=1}^{N_j} \sum_{k=1}^{M_{j-1}} \left[\max\left\{c_e\left(\Delta_j - x_j^i\right), c_t\left(x_j^i - \Delta_j\right)\right\} + \max\left\{c_e, c_t\right\} T_{j-1}^k\right] - \\
& \sum_{j=2}^{n} \frac{1}{M_j} \sum_{i=1}^{N_j} \sum_{k=1}^{M_{j-1}} \max\left\{c_e\left(\Delta_j - x_j^i - T_{j-1}^k\right), c_t\left(x_j^i + T_{j-1}^k - \Delta_j\right)\right\}
\end{aligned}
$$

$$
\begin{aligned}
= & \sum_{j=2}^{n} \frac{1}{M_j} \sum_{i \in A_j} \sum_{k=1}^{M_{j-1}} \left[ c_e \left( \Delta_j - x_j^i \right) + \max \left\{ c_e, c_t \right\} T_{j-1}^k \right] - \\
& \sum_{j=2}^{n} \frac{1}{M_j} \sum_{i \in A_j} \sum_{k=1}^{M_{j-1}} \max \left\{ c_e \left( \Delta_j - x_j^i - T_{j-1}^k \right), c_t \left( x_j^i + T_{j-1}^k - \Delta_j \right) \right\} + \\
& \sum_{j=2}^{n} \frac{1}{M_j} \sum_{i \in B_j} \sum_{k=1}^{M_{j-1}} \left[ c_t \left( x_j^i - \Delta_j \right) + \max \left\{ c_e, c_t \right\} T_{j-1}^k \right] - \\
& \sum_{j=2}^{n} \frac{1}{M_j} \sum_{i \in B_j} \sum_{k=1}^{M_{j-1}} c_t \left( x_j^i + T_{j-1}^k - \Delta_j \right)
\end{aligned}
$$

To compute $\delta$ we define two sets for each procedure, $A_j$ and $B_j$ for all $j \in \{1, \ldots, n\}$. $A_j$ consists of all of the observations of procedure $j$ duration less than $\Delta_j$ whereas $B_j$ has all the observations of procedure $j$ greater than or equal to $\Delta_j$. The number of elements in sets $A_j$ and $B_j$ are denoted by $N_j^A$ and $N_j^B$ respectively ($N_j = N_j^A + N_j^B \quad \forall j$).

$$
\begin{aligned}
& \delta = \\
& \sum_{j=2}^{n} \frac{1}{M_j} \sum_{i \in A_j} \sum_{k=1}^{M_{j-1}} \min\{(\max\{c_e, c_t\} + c_e) T_{j-1}^k, (\max\{c_e, c_t\} - c_e) T_{j-1}^k + (c_e + c_t)(\Delta_j - x_j^i)\} \\
& + \sum_{j=2}^{n} \frac{1}{M_j} \sum_{i \in B_j} \sum_{k=1}^{M_{j-1}} (\max\{c_e, c_t\} - c_e) T_{j-1}^k
\end{aligned}
$$

To simplify the difference, $\delta$, we assume the penalty for tardiness is equal to the penalty for earliness. $\delta$ becomes:

$$
\delta = \sum_{j=2}^{n} \frac{1}{M_j} \sum_{i \in A_j} \sum_{k=1}^{M_{j-1}} \min\{2T_{j-1}^k, 2(\Delta_j - x_j^i)\} = 2 \sum_{j=2}^{n} \frac{1}{M_j} \sum_{i \in A_j} \sum_{k=1}^{M_{j-1}} \min\{T_{j-1}^k, \Delta_j - x_j^i\}
$$

We have a set of $n$ procedures $P = (p_1, \ldots, p_n)$. Any ordering of procedures $P = (p_1, \ldots, p_n)$ has different a objective function (2.6) value and a value of the upper bound (2.11). The difference between two values is $\delta$ . For instance $\delta$ computed given the optimal order may be equal the largest possible value of $\delta$, whereas there can be an ordering which has smaller $\delta$ so that this ordering has smaller value of (2.11). Assume we find the order ($o'$) having the minimum value of (2.11) given the time allowances of all procedures. The maximum error of choosing the suggested order ($o'$) instead of the optimal order ($o^*$) given the time allowances of all procedures, is the difference between the maximum value of $\delta$ over all possible orderings and the minimum value of $\delta$ over all possible orderings.

The maximum value of $\delta$ among all possible orderings ($c_e = c_t$):

$$\begin{aligned}
\delta &= 2\sum_{j=2}^{n} \frac{1}{M_j} \sum_{i \in A_j} \sum_{k=1}^{M_{j-1}} \min\{T_{j-1}^k, \Delta_j - x_j^i\} \\
&\leq 2\sum_{j=2}^{n} \frac{1}{M_j} \sum_{i \in A_j} M_{j-1}(\Delta_j - x_j^i) \\
&= 2\sum_{j=2}^{n} \frac{1}{N_j} \sum_{i \in A_j} \Delta_j - x_j^i
\end{aligned}$$

which does not depend the the ordering. In other words this upper bound is same for all possible orderings.

In order to find the minimum value of $\delta$, we use the inequality (2.8):

$$\begin{aligned}
\delta &= 2\sum_{j=2}^{n} \frac{1}{M_j} \sum_{i \in A_j} \sum_{k=1}^{M_{j-1}} \min\{T_{j-1}^k, \Delta_j - x_j^i\} \\
&\geq 2\sum_{j=2}^{n} \frac{1}{M_j} \sum_{i \in A_j} \frac{M_{j-1}}{N_{j-1}} \sum_{k=1}^{N_{j-1}} \min\{\tau_{j-1}^k, \Delta_j - x_j^i\} \\
&= 2\sum_{j=2}^{n} \frac{1}{N_j N_{j-1}} \sum_{i \in A_j} \sum_{k=1}^{N_{j-1}} \min\{\tau_{j-1}^k, \Delta_j - x_j^i\}
\end{aligned}$$

This lower bound depends on the order of the procedures. Given the data $\mathbf{x} = (\mathbf{x_1}, \ldots, \mathbf{x_n})$ and $\Delta = (\Delta_1, \ldots, \Delta_n)$ we can compute $\frac{1}{N_j N_{j-1}} \sum_{i \in A_j} \sum_{k=1}^{M_{j-1}} \min\{\tau_{j-1}^k, \Delta_j - x_j^i\}$ for any two procedures. We can store the values in a matrix where rows represent the predecessors and columns represent their successors and set the diagonal elements equal to infinity. For any ordering we can compute the lower bound by summing the $n-1$ entries of the matrix corresponding to the $n-1$ predecessor-successor duos in that ordering. We would like to find the ordering which has the minimum value. This problem is same as the Asymmetric Traveling Salesman Problem (Dantzig et al., 1954).

**Lemma 20.** *The maximum error of the heuristic is the difference between the maximum value of $\delta$ over all possible orderings and the minimum value of $\delta$ over all possible orderings, which is equal to:*

$$2\sum_{j=2}^{n} \frac{1}{N_j} \sum_{i \in A_j} (\Delta_j - x_j^i) - 2\sum_{j=2}^{n} \frac{1}{N_j N_{j-1}} \sum_{i \in A_j} \sum_{k=1}^{N_{j-1}} \min\{\tau_{j-1}^k, \Delta_j - x_j^i\}$$

.

**Remark.** *If $\Delta_j$'s are less than the minimum observation of* $\mathbf{x_j}$ $\forall j \in \{1,\ldots,n\}$ *($\Delta_j \leq \min(\{x_j^i\}_{i=1}^{M_n}) \quad \forall j \in \{1,\ldots,n\}$), this error would be equal to zero. In other words if the time allowances of all procedures are less than the smallest observation of any procedure duration, the sequence minimizing the objective function is ordering procedures in increasing average value of the set* $\tau_\mathbf{j}$, $\overline{\tau_j}$.

### Heuristic 2

The upperbound in Heuristic 1 is strict for the observations of procedures which are larger then the time allowance of that procedure. However, the contribution of the observations which are smaller than the time allowance of that procedure to the upper bound is greater than or equal to their contribution to the objective function. Our goal is to find a stricter upper bound at the expense of computational complexity.

We would like to find a better upper bound on the optimal solution than the upper bound computed in Heuristic 1. We define two sets for each procedure, $A_j$ and $B_j$ for all $j \in \{1,\ldots,n\}$. $A_j$ consists of all of the observations of procedure $j$ duration less than $\Delta_j$ whereas $B_j$ has all the observations of procedure $j$ greater than or equal to $\Delta_j$. The number of elements in sets $A_j$ and $B_j$ are denoted by $N_j^A$ and $N_j^B$ respectively ($N_j = N_j^A + N_j^B \quad \forall j$).

$$
\begin{aligned}
&f(d_1,\ldots,d_n)\\
&= \sum_{j=1}^{n} \frac{1}{M_j} \sum_{i=1}^{N_j} \sum_{k=1}^{M_{j-1}} \max\left\{c_e\left(\Delta - x_j^i - T_{j-1}^k\right), c_t\left(x_j^i + T_{j-1}^k - \Delta_j\right)\right\}\\
&= \sum_{j=1}^{n} \frac{1}{M_j} \sum_{i\in A_j} \sum_{k=1}^{M_{j-1}} \max\left\{c_e\left(\Delta_j - x_j^i - T_{j-1}^k\right), c_t\left(x_j^i + T_{j-1}^k - \Delta_j\right)\right\} +\\
&\sum_{j=1}^{n} \frac{1}{M_j} \sum_{i\in B_j} \sum_{k=1}^{M_{j-1}} \max\left\{c_e\left(\Delta_j - x_j^i - T_{j-1}^k\right), c_t\left(x_j^i + T_{j-1}^k - \Delta_j\right)\right\}\\
&= \sum_{j=1}^{n} \frac{1}{M_j} \sum_{i\in A_j} \sum_{k=1}^{M_{j-1}} \left[c_e\left(\Delta_j - x_j^i\right) + \max\left\{c_e\left(-T_{j-1}^k\right), c_t\left(T_{j-1}^k\right) + (c_e + c_t)\left(x_j^i - \Delta_j\right)\right\}\right] +\\
&\sum_{j=1}^{n} \frac{1}{M_j} \sum_{i\in B_j} \sum_{k=1}^{M_{j-1}} c_t\left(x_j^i + T_{j-1}^k - \Delta_j\right)\\
&= \sum_{j=1}^{n} \frac{1}{M_j} \sum_{i=1}^{N_j} \sum_{k=1}^{M_{j-1}} \max\left\{c_e\left(\Delta_j - x_j^i\right), c_t\left(x_j^i - \Delta_j\right)\right\} + \qquad (2.12)\\
&\sum_{j=1}^{n} \frac{1}{M_j} \sum_{i\in A_j} \sum_{k=1}^{M_{j-1}} \max\left\{c_e\left(-T_{j-1}^k\right), c_t\left(T_{j-1}^k\right) + (c_e + c_t)\left(x_j^i - \Delta_j\right)\right\} + \qquad (2.13)
\end{aligned}
$$

$$\sum_{j=1}^{n} \frac{1}{M_j} \sum_{i \in B_j} \sum_{k=1}^{M_{j-1}} c_t \left(T_{j-1}^k\right) \tag{2.14}$$

The objective function has three parts: (2.12), (2.13), (2.14). The first part (2.12) is same for all possible orderings. Our goal is to find an upper bound on parts (2.13) and (2.14) using the inequalities (2.9) and (2.10). We also assume $\mathbf{T_j} = 0$ for any $j \leq 0$.

$$(2.13) \leq \sum_{j=2}^{n} \frac{1}{M_j} \left[ \sum_{i \in A_j} \frac{M_{j-1}}{N_{j-1}} \sum_{k=1}^{N_{j-1}} \max\left\{c_e\left(-\tau_{j-1}^k\right), c_t\left(\tau_{j-1}^k\right) + (c_e + c_t)\left(x_j^i - \Delta_j\right)\right\} + \max\{c_e, c_t\} N_j^A \frac{M_{j-1}}{M_{j-2}} \sum_{m=1}^{M_{j-2}} T_{j-2}^m \right]$$

$$(2.14) \leq c_t \sum_{j=2}^{n} \frac{N_j^B}{M_j} \left( \frac{M_{j-1}}{N_{j-1}} \sum_{k=1}^{N_{j-1}} \tau_j^k + \frac{M_{j-1}}{M_{j-2}} \sum_{k=1}^{M_{j-2}} T_{j-2}^k \right) = c_t \sum_{j=2}^{n} \left( \frac{N_j^B}{N_j} \overline{\tau_{j-1}} + \frac{N_j^B}{N_j} \overline{T_{j-2}} \right)$$

(2.13)+(2.14)

$$\begin{aligned}
&\leq \sum_{j=2}^{n} \frac{1}{M_j} \sum_{i \in A_j} \left[ \frac{M_{j-1}}{N_{j-1}} \sum_{k=1}^{N_{j-1}} \max\left\{c_e\left(-\tau_{j-1}^k\right), c_t\left(\tau_{j-1}^k\right) + (c_e + c_t)\left(x_j^i - \Delta_j\right)\right\} \right] + \\
&\quad \max\{c_e, c_t\} \sum_{j=2}^{n} \frac{N_j^A}{N_j} \overline{T_{j-2}} + c_t \sum_{j=2}^{n} \frac{N_j^B}{N_j} \overline{\tau_{j-1}} + \max\{c_e, c_t\} \sum_{j=2}^{n} \frac{N_j^B}{N_j} \overline{T_{j-2}} \\
&= \sum_{j=2}^{n} \frac{1}{M_j} \sum_{i \in A_j} \left[ \frac{M_{j-1}}{N_{j-1}} \sum_{k=1}^{N_{j-1}} \max\left\{c_e\left(-\tau_{j-1}^k\right), c_t\left(\tau_{j-1}^k\right) + (c_e + c_t)\left(x_j^i - \Delta_j\right)\right\} \right] + \\
&\quad \max\{c_e, c_t\} \sum_{j=2}^{n} \overline{T_{j-2}} + c_t \sum_{j=2}^{n} \frac{N_j^B}{N_j} \overline{\tau_{j-1}} \\
&= \sum_{j=2}^{n} \frac{1}{N_j N_{j-1}} \sum_{i \in A_j} \left[ \sum_{k=1}^{N_{j-1}} \max\left\{c_e\left(-\tau_{j-1}^k\right), c_t\left(\tau_{j-1}^k\right) + (c_e + c_t)\left(x_j^i - \Delta_j\right)\right\} \right] + \\
&\quad \max\{c_e, c_t\} \sum_{j=2}^{n} \overline{T_{j-2}} + c_t \sum_{j=2}^{n} \frac{N_j^B}{N_j} \overline{\tau_{j-1}}
\end{aligned} \tag{2.15}$$

where $\overline{\tau_j} = \frac{1}{N_j} \sum_{m=1}^{N_j} \tau_j^m$ which is the average of the set $\tau_\mathbf{j}$ for any $j \in \{1, \ldots, n\}$ and $\overline{T_j}$ is the average of the set $\mathbf{T_j}$ or any $j \in \{1, \ldots, n\}$ and $\overline{T_j}$. We can only compute $\mathbf{T_j}$ only if we

know the order of the procedures. We use the inequality (2.10) to bound the average of $\mathbf{T_j}$ which gives an upperbound independent of the processing order.

$$
\begin{aligned}
&(2.13)+(2.14)\\
&\leq \sum_{j=2}^{n} \frac{1}{N_j N_{j-1}} \sum_{i \in A_j} \left[ \sum_{k=1}^{N_{j-1}} \max\left\{ c_e\left(-\tau_{j-1}^k\right), c_t\left(\tau_{j-1}^k\right) + (c_e + c_t)\left(x_j^i - \Delta_j\right)\right\}\right] +\\
&\quad \max\{c_e, c_t\} \sum_{j=1}^{n-1} (n-j-1)\overline{\tau_j} + c_t \sum_{j=2}^{n} \frac{N_j^B}{N_j}\overline{\tau_{j-1}}
\end{aligned}
$$

This upper bound can be rearranged as follows:

$$
\begin{aligned}
&\sum_{j=2}^{n} \left( \frac{1}{N_j N_{j-1}} \sum_{i \in A_j} \left[ \sum_{k=1}^{N_{j-1}} \max\left\{ c_e\left(-\tau_{j-1}^k\right), c_t\left(\tau_{j-1}^k\right) + (c_e + c_t)\left(x_j^i - \Delta_j\right)\right\}\right] + c_t \frac{N_j^B}{N_j}\overline{\tau_{j-1}} \right) +\\
&\max\{c_e, c_t\} \sum_{j=1}^{n-1} (n-j-1)\overline{\tau_j} \qquad (2.16)
\end{aligned}
$$

Ignore the second component of the upper bound (2.16) for now and observe that the first component is the summation of $n-1$ terms containing variables with subscript $j$ or $j-1$. The ordering of $n$ procedures can be defined as $n-1$ predecessor-immediate successor relationships. For instance we have 3 procedures, $\mathbf{P} = (p_1, p_2, p_3)$ and the order they are processed in is $p_2 - p_1 - p_3$. Instead we can define couples of a procedure and its immediate successor such as $p_2 - p_1$ and $p_1 - p_3$. Given the predecessor and its immediate successor relationships the sum of the terms for all possible $(j-1, j)$ can be computed. We can build a matrix holding the individual terms in the first component of (2.16). The rows correspond to the predecessors and columns represent their successors. Given the order of procedures the first component of the upper bound (2.16) is sum of the $n-1$ entries of the matrix corresponding to $n-1$ predecessor-successor duos. The optimal order minimizing the sum of those terms is finding a path which minimizes the total cost which is a Asymmetric Traveling Salesman Path Problem. The graph is a fully connected graph where the nodes are the procedures and distances are individual terms in the first component of (2.16) stored in a matrix. The second component can be computed without knowing the order of the procedures. In order to evaluate the upper bound (2.16) at $d$ we need to compute an $n \times n$ matrix holding the values of the sum of the first component of (2.16) and also the average $\tau_{\mathbf{j}}$ for all procedure $j$'s.

We formulate this problem as a dynamic program. We have a set of $n$ procedures $P = (p_1, \ldots, p_n)$. The $(i, j)$ entry, $w_{i,j} = w_{p_i, p_j}$ of the distance matrix $W$ contains the value of the terms in the first component of (2.16) computed assuming that the procedure $p_j$ follows

the procedure $p_i$.

$$w_{p_i,p_j} = \frac{1}{N_{p_j} N_{p_i}} \sum_{i \in A_{p_j}} \left[ \sum_{k=1}^{N_{p_i}} \max \left\{ c_e \left( -\tau_{p_i}^k \right), c_t \left( \tau_{p_i}^k \right) + (c_e + c_t) \left( x_{p_j}^i - \Delta_{p_j} \right) \right\} \right] + c_t \frac{N p_j^B}{N_{p_j}} \overline{\tau_{p_i}}$$

The diagonal entries are empty or can be set equal to infinity for the sake of computation. Our goal is to compute the minimum of the cost function $Cost(start, end, S)$ over all possible start and end combinations

$$Cost(p_i, p_j, S) = \min_{p_k} \{ w_{p_i,p_k} + Cost(p_k, p_j, S \setminus \{p_k\}) \} + |S| \overline{\tau_i}$$

where $Cost(p_i, p_j, \emptyset) = w_{p_i,p_j}$ and $S$ is any subset of the procedures excluding the start and the end.

**Algorithm 2:** Dynamic Programming for Heuristic 2

```
for Each Procedure p_j ∈ P = (p_1, ..., p_n) do
    Schedule that procedure p_j last
    for Each Procedure p_i ∈ P \ {p_j} do
        Cost(p_i, p_j, ∅) = w_{p_i,p_j}
    end
    for For any subset S ⊂ P \ {p_i, p_j} do
        Cost(p_i, p_j, S) = min_{p_k}{w_{p_i,p_k} + Cost(p_k, p_j, S \ {p_k})} + |S| τ̄_i
    end
    Choose the ordering with minimum cost
end
```

The worst case complexity of this approach $\mathcal{O}(n^2 2^n)$, which is exponential rather than factorial $\mathcal{O}(n!)$ because it goes through all the subsets of the procedures.

We may find a sequence with lower objective function value than the objective function value computed using the ordering suggested by the heuristic 1. However the complexity of heuristic 2 is not polynomial.

### Heuristic 3

Function (2.16) overestimates function (2.15). Especially as the number of procedures ,$n$, increases, the second term in the upper bound inflates above its actual value and dominates the remaining terms. Our goal is to approximate the function (2.15) which can be computed if the order is given, because $\overline{T_j}$ for all $j \in \{1, \ldots, n-2\}$ can be computed only if the order is known. The first step is to define tardiness of procedure $j$ given the observations $(x_1^{k_1}, \ldots, x_j^{k_j})$ as a function of $(\tau_1^{k_1}, \ldots, \tau_j^{k_j})$ where $I_{\Delta_j}(T) = \max\{0, T - \max\{0, \Delta_j - x_j\}\}$.

$$T_1^{k_1} = \tau_1^{k_1} \qquad k_1 \in \{1, \ldots, N_1\}$$
$$T_2^{(k_1,k_2)} = \tau_2^{k_2} + I_{\Delta_2}(\tau_1^{k_1}) \qquad k_1 \in \{1, \ldots, N_1\}, k_2 \in \{1, \ldots, N_2\}$$

$$T_3^{(k_1,k_2,k_3)} = \tau_3^{k_3} + I_{\Delta_3}(\tau_2^{k_2} + I_{\Delta_2}(\tau_1^{k_1}))\ k_1 \in \{1,\ldots,N_1\}, k_2 \in \{1,\ldots,N_2\}, k_3 \in \{1,\ldots,N_3\}$$
$$\vdots$$

For notational brevity we use $T_{j-1}$ as the argument of the function $I_{\Delta_j}(\cdot)$. The index of $T_{j-1}$ is $m_{j-1} \in \{1,\ldots,M_{j-1}\}$ where $M_{j-1} = \prod_{\ell=1}^{j-1} N_\ell$ to define observations $(x_1^{k_1},\ldots,x_{j-1}^{k_{j-1}})$. $((k_1,\ldots,k_{j-1}) \to m_{j-1})$

$$\begin{aligned}
T_1^{k_1} &= \tau_1^{k_1} \qquad k_1 \in \{1,\ldots,N_1\} \\
T_2^{(m_1,k_2)} &= \tau_2^{k_2} + I_{\Delta_2}(T_1^{m_1}) \qquad m_1 \in \{1,\ldots,M_1\}, k_2 \in \{1,\ldots,N_2\} \\
T_3^{(m_2,k_3)} &= \tau_3^{k_3} + I_{\Delta_3}(T_2^{m_2}) \qquad m_2 \in \{1,\ldots,M_2\}, k_3 \in \{1,\ldots,N_3\} \\
&\vdots \\
T_j^{(m_{j-1},k_j)} &= \tau_j^{k_j} + I_{\Delta_j}(T_{j-1}^{m_{j-1}}) \qquad m_{j-1} \in \{1,\ldots,M_{j-1}\}, k_j \in \{1,\ldots,N_j\}
\end{aligned}$$

The objective function becomes:

$$\begin{aligned}
f &= \sum_{j=1}^{n} \frac{1}{M_j} \sum_{i=1}^{N_j} \sum_{k=1}^{M_{j-1}} \max\left\{c_e\left(\Delta_j - x_j^i - T_{j-1}^k\right), c_t\left(x_j^i + T_{j-1}^k - \Delta_j\right)\right\} \\
&= \sum_{j=1}^{n} \frac{1}{M_j} \sum_{i=1}^{N_j} \sum_{k=1}^{N_{j-1}} \sum_{m=1}^{M_{j-2}} \max\left\{c_e\left(\Delta_j - x_j^i - \tau_{j-1}^k - I_{\Delta_{j-1}}(T_{j-1}^m)\right),\right. \\
&\qquad \left. c_t\left(x_j^i + \tau_{j-1}^k + I_{\Delta_{j-1}}(T_{j-1}^m) - \Delta_j\right)\right\} \\
&= \sum_{j=1}^{n} \frac{1}{M_j} \sum_{i\in A_j} \sum_{k=1}^{N_{j-1}} \sum_{m=1}^{M_{j-2}} \max\left\{c_e\left(\Delta_j - x_j^i - \tau_{j-1}^k - I_{\Delta_{j-1}}(T_{j-1}^m)\right),\right. \\
&\qquad \left. c_t\left(x_j^i + \tau_{j-1}^k + I_{\Delta_{j-1}}(T_{j-1}^m) - \Delta_j\right)\right\} + \\
&\quad \sum_{j=1}^{n} \frac{1}{M_j} \sum_{i\in B_j} \sum_{k=1}^{N_{j-1}} \sum_{m=1}^{M_{j-2}} c_t\left(x_j^i + \tau_{j-1}^k + I_{\Delta_{j-1}}(T_{j-1}^m) - \Delta_j\right) \\
&= \sum_{j=1}^{n} \frac{1}{M_j} \sum_{i\in A_j} \sum_{k=1}^{N_{j-1}} \sum_{m=1}^{M_{j-2}} \max\left\{c_e\left(\Delta_j - x_j^i - \tau_{j-1}^k - I_{\Delta_{j-1}}(T_{j-1}^m)\right),\right. \\
&\qquad \left. c_t\left(x_j^i + \tau_{j-1}^k + I_{\Delta_{j-1}}(T_{j-1}^m) - \Delta_j\right)\right\} + \\
&\quad \sum_{j=1}^{n} \frac{1}{M_j} \sum_{i\in B_j} \sum_{k=1}^{N_{j-1}} \sum_{m=1}^{M_{j-2}} c_t\left(\tau_j^i + \tau_{j-1}^k + I_{\Delta_{j-1}}(T_{j-1}^m)\right)
\end{aligned}$$

**Remark.** $\sum_{i\in B_j}\tau_j^i = \sum_{i=1}^{N_j}\tau_j^i$ *because of the definition of* $\tau_{\mathbf{j}} = \{\max\{0, x_j^i - \Delta_j\} : i \in \{1,\dots,N_j\}\}$ *and* $B_j$ *has all the observations of procedure* $j$ *greater than or equal to* $\Delta_j$.

We would like to approximate the upper bound (2.15). In order to do that we need to approximate $\overline{T_j}$ for all $j \in \{1,\dots,n-2\}$. We find a function of $(\overline{\tau_1},\dots,\overline{\tau_j})$ approximating the average tardiness terms $\overline{T_j}$. We know the exact function of $\overline{T_j}$, for instance the average tardiness of the third procedure in order is:

$$\frac{1}{M_3}\sum_{k_1=1}^{N_1}\sum_{k_2=1}^{N_2}\sum_{k_3=1}^{N_3} T_3^{(k_1,k_2,k_3)} = \frac{1}{M_3}\left(\sum_{k_1=1}^{N_1}\sum_{k_2=1}^{N_2}\sum_{k_3=1}^{N_3}\tau_3^{k_3} + I_{\Delta_3}(\tau_2^{k_2} + I_{\Delta_2}(\tau_1^{k_1}))\right)$$

Observe that the function $I_{\Delta_j}(T) = \max\{0, T - \max\{0, \Delta_j - x_j\}\}$ returns a non-negative real number in$[0,T]$. We can approximate the output of this function with a discount factor ranging from 0 to 1. Since computing $I_{\Delta_j}(T)$ for all realizations of a procedure and then calculating the average tardiness are expensive, we try to approximate the average tardiness of a procedure by using discount factors as follows:

$$\begin{array}{rcll}
\overline{T_1} & = & \overline{\tau_1} & \\
\overline{T_2} & \leq & \overline{\tau_1} + \overline{\tau_2} & \rightarrow \quad \overline{T_2} = \gamma_1\overline{\tau_1} + \overline{\tau_2} \quad \text{where } 0 \leq \gamma_1 \leq 1 \\
\overline{T_3} & \leq & \overline{\tau_1} + \overline{\tau_2} + \overline{\tau_3} & \rightarrow \quad \overline{T_3} = \gamma_2(\gamma_1\overline{\tau_1} + \overline{\tau_2}) + \overline{\tau_3} \quad \text{where } 0 \leq \gamma_2 \leq 1
\end{array}$$

We may find a set of $\Gamma = (\gamma_1,\dots,\gamma_{n-3})$ to exactly compute $(\overline{T_1},\dots,\overline{T_{n-2}})$ given the order. The exact value of $\gamma_j$ depends on the the number of observations of the procedure $j+1$ which are less than $\Delta_{j+1}$ $(N_{j+1}^A)$ and also its distribution. Choosing different $\gamma_j$'s for each $j+1$st procedure implies we find different functions for each possible ordering, which we would like to avoid. That is why we treat $\gamma$ as a discount factor which is used to diminish the effect of previous procedure's tardinesses on the current procedure's tardiness.

(2.13)+(2.14)

$$\begin{aligned}
&\approx \sum_{j=2}^{n}\frac{1}{N_jN_{j-1}}\sum_{i\in A_j}\left[\sum_{k=1}^{N_{j-1}}\max\left\{c_e\left(-\tau_{j-1}^k\right), c_t\left(\tau_{j-1}^k\right) + (c_e+c_t)\left(x_j^i - \Delta_j\right)\right\}\right] + \quad (2.17)\\
&\max\{c_e,c_t\}\sum_{j=1}^{n-1}\left(\frac{1-\gamma^{n-j}}{1-\gamma} - 1\right)\overline{\tau_j} + c_t\sum_{j=2}^{n}\frac{N_j^B}{N_j}\overline{\tau_{j-1}}
\end{aligned}$$

**Algorithm 3:** Heuristic 3

Compute the schedule $\Delta_{p_j} = x_{p_j}^{(\lceil N_{p_j} \cdot c_t/(c_e+c_t) \rceil)}$ $\forall p_j \in \{p_1, \ldots, p_n\}$.

Compute $\tau_{\mathbf{p_j}} = \{\max\{0, x_{p_j}^{k_j} - \Delta_{p_j}\} : k_j \in \{1, \ldots, N_{p_j}\}\}$ and $\overline{\tau_{p_j}} = \frac{1}{N_{p_j}} \sum_{k_j=1}^{N_{p_j}} \tau_{p_j}^{k_j}$ $\forall p_j \in \{p_1, \ldots, p_n\}$.

Compute the matrix W where $(i,j)$'t entry of $W$ $(w_{i,j})$ is equal to

$w_{p_i,p_j} =$

$\frac{1}{N_{p_j} N_{p_i}} \sum_{i \in A_{p_j}} \left[ \sum_{k=1}^{N_{p_i}} \max \left\{ c_e \left( -\tau_{p_i}^k \right), c_t \left( \tau_{p_i}^k \right) + (c_e + c_t) \left( x_{p_j}^i - \Delta_{p_j} \right) \right\} \right] + c_t \frac{N p_j^B}{N_{p_j}} \overline{\tau_{p_i}}$.

**for** *Each Discount Rate* $\gamma \in \Gamma = (\gamma_1, \ldots, \gamma_m)$ **do**

  **for** *Each Procedure* $p_j \in P = (p_1, \ldots, p_n)$ **do**

    Schedule that procedure $p_j$ last

    **for** *Each Procedure* $p_i \in P \setminus \{p_j\}$ **do**

      $Cost(p_i, p_j, \emptyset) = w_{p_i,p_j}$

    **end**

    **for** *For any subset* $S \subset P \setminus \{p_i, p_j\}$ **do**

      **if** $\gamma == 1$ **then**

        $Cost(p_i, p_j, S) = \min_{p_k} \{w_{p_i,p_k} + Cost(p_k, p_j, S \setminus \{p_k\})\} + \left( \frac{1-\gamma^{|S|+1}}{1-\gamma} - 1 \right) \overline{\tau_i}$

      **end**

      **else**

        $Cost(p_i, p_j, S) = \min_{p_k} \{w_{p_i,p_k} + Cost(p_k, p_j, S \setminus \{p_k\})\} + |S| \overline{\tau_i}$

      **end**

    **end**

    Choose the ordering with minimum cost and add to a list.

  **end**

  For each ordering in the list evaluate the objective function at $\Delta_{p_j} = x_{p_j}^{(\lceil N_{p_j} \cdot c_t/(c_e+c_t) \rceil)}$ $\forall p_j \in \{p_1, \ldots, p_n\}$. Choose the ordering with minimum objective function value.

**end**

Given one discount rate, the complexity of finding the sequence suggested by heuristic 3 is same as the complexity of heuristic 2. Heuristic 3's solution can be improved by choosing multiple discount factors.

### Computational Comparison of the Sequencing Heuristics

We optimize the objective function of all possible orderings ($n!$) and return their minimum solution. We call this the brute force approach. Heuristics 1 requires the computation of the minimum solution of one ordering. Heuristic 2 solves an asymmetric traveling salesman problem and then optimizes the objective function corresponding the ordering given by the asymmetric traveling salesman problem. Heuristic 3 repeats the steps of the heuristic 2 multiple times. We compare the performance of the brute force approach with the performance

of the heuristics in terms of the percentage of finding the optimal, average gap percentage and the average run length.

We create data points sampled from discrete distributions. Example runs:

1. 5 procedures sampled from the distributions:

   - Uniform(50,118)
   - Binomial(400,0.6)+Uniform(235,265)
   - Binomial(500,0.1)
   - Uniform(120,168)
   - Binomial(440,0.4)

| | Heuristic 1 | Heuristic 2 | Heuristic 3 | Brute Force |
|---|---|---|---|---|
| Percentage of Finding Optimal | 35% | 55% | 70 % | 100% |
| Average Gap Percentage | 0.00504 | 0.001556 | 0.00119 | 0 |
| Average Run Length (sec) | 3.78 | 4.56 | 17.74 | 450.02 |

Figure 2.1: Computational Comparison of the Sequencing Heuristics

2. 5 procedures sampled from the distributions:

   - Uniform(170,218)
   - Binomial(336,0.5)+Uniform(145,175)
   - Binomial(440,0.48)
   - Uniform(220,268)
   - Binomial(540,0.3)

| | Heuristic 1 | Heuristic 2 | Heuristic 3 | Brute Force |
|---|---|---|---|---|
| Percentage of Finding Optimal | 15% | 10% | 35% | 100% |
| Average Gap Percentage | 0.00518 | 0.004403 | 0.00207 | 0 |
| Average Run Length (sec) | 4.45 | 5.16 | 19.69 | 514.96 |

Figure 2.2: Computational Comparison of the Sequencing Heuristics

In the examples above, the relative heuristic performance is as expected. Heuristic 3 finds the best quality solutions (in terms of both finding optimal solutions, and average gap) while heuristic 1 has the shortest average run length.

## 2.4 Data Selection

When a surgery has been scheduled, the hospital can record a variety of data: the condition of the patient, type of the procedure, surgeon, anesthesia attendant, the operating room where the procedure is going to take place etc. The historical data of a procedure consists of those recorded features, the scheduled duration of that procedure and its actual duration. The type of a procedure may not be the only feature defining the distribution of the procedure's duration. For instance the distribution of the procedure duration may differ with respect to the age of the patient. Using only the historical data of a procedure's duration while minimizing the expected total earliness and tardiness may introduce bias to the model. If there is a black box returning the distribution of the procedure given the features:

1. Bias introduced to the model because of ignoring the true procedure duration distribution may decrease.

2. In the section "Complexity of the Objective Function Evaluation" the complexity to evaluate the objective function is shown to be $\min\{\mathcal{O}(n^3 x_{max}^2), \mathcal{O}(M_n)\}$, where $x_{max}$ is the maximum value of all procedure durations and $M_n = \prod_{\ell=1}^{n} N_\ell$. If this black box can eliminate some observed procedure durations which are not probable, the complexity of computation of the objective function given a schedule may decrease so does the complexity of minimization of the objective function.

In Chapter 1, we showed that the case with two procedures ($n = 1$) is equivalent to the Data-Driven Newsvendor Problem (Section 1.2.3). $c_e$ and $c_t$ correspond to overage and underage costs respectively, $D_1$ corresponds to the order quantity and $x_1$ corresponds to the demand. The objective is of the Data-Driven Newsvendor Problem is:

$$\min_{D_1} \quad \frac{1}{N_1} \sum_{i=1}^{N_1} \max\{c_e(D_1 - x_1^i), c_t(x_1^i - D_1)\} \tag{2.18}$$

Ban and Rudin (2018) study the Data-Driven Newsvendor Problem with historical data consisting of $p$ features and their associated demand. They call the model (2.18) optimizing the newsvendor problem over an empirical distribution the 'Sample Average Approximation (SSA) approach'. They develop what they call an Empirical Risk Minimization (ERM) Algorithm to solve the newsvendor problem with feature data. The ERM approach is equivalent to high-dimensional quantile regression and can be solved by convex optimization methods. Under some assumptions about the demand models, they proved the consistency and the asymptotic optimality of the regression coefficients of ERM model as the number of observations goes to infinity, and showed the inconsistency of the SSA approach.

We adapt the The ERM approach (Ban and Rudin, 2018) to the appointment scheduling problem with only two procedures. The empirical risk function is $\beta_1(\cdot)$ among a fixed class of functions $\mathcal{B}$ and the feature matrix is the first procedure $A_1 \in \mathbb{R}^{N_1 \times p}$. The appointment

scheduling with two procedures becomes:

$$\min_{\beta_1 \in \mathcal{B}, \{\beta: \mathbb{R}^p \to \mathbb{R}\}} \frac{1}{N_1} \sum_{i=1}^{N_1} \max\{c_e(\beta_1(A_1^i) - x_1^i), c_t(x_1^i - \beta_1(A_1^i))\}$$

where $A_1^i$ is the $i^{\text{th}}$ row of the matrix $A_1$ corresponding to the feature vector of the observation $i \in \{1, \cdots, N_1\}$.

If there are more procedures i.e. $n \geq 2$, the ERM model has $n$ risk functions $\beta_j(\cdot) \ \ \forall j \in \{1, \cdots, n\}$ for each completion time $C_j \ \ \forall j \in \{1, \cdots, n\}$. The completion time of the second procedure depends on the distribution of first and second procedures durations $(\mathbf{x_1}, \mathbf{x_2})$. The feature matrix of the second procedure's completion time should contain first and second procedure's features and the interaction terms (e.g. $A_1^i \odot A_2^i$). The feature matrix of the $n^{\text{th}}$ procedure's completion time should contain all procedures' features and the all the interaction terms. The number of arguments of each risk function grows exponentially. For instance assume $\mathcal{B}$ is the set of all linear functions and ignore all the interaction terms of the feature matrices, the number of parameters we need to estimate is$p + 2p + \cdots + np = \frac{n(n+1)}{2}p$. If we include the interaction terms then the number of parameters needed to estimate grows exponentially. Let's define feature matrix $\mathcal{A}_j$ for completion time of procedure $C_j$. $\mathcal{A}_j$ has every row combination of $A_1, \cdots, A_j$ and also the interactions of those matrices if needed. The model becomes:

$$\min_{\beta_1, \cdots \beta_n} \sum_{j=1}^{n} \frac{1}{M_j} \sum_{i=1}^{M_j} \max\{c_e(\beta_j(\mathcal{A}_j^i) - C_j^i), c_t(C_j^i - \beta_j(\mathcal{A}_j^i))\}$$

This problem can be written as a linear program, so it is still convex. But as the feature space grows it becomes harder to solve the instances.

Instead of including the features into the model, we would like to preprocess the data and then optimize the original objective taking the expectation over the subset of the observations with the decision variables $D_j$ (or $\Delta_j$). Our goal is to eliminate the data points which are not likely to occur, then compute expected weighted earliness and tardiness over the empirical distribution function of the remaining data points. We use non-parametric inferential statistical methods as a heuristic to choose the subset of the observations.

### 2.4.1 Expert (Surgeon) Opinions

Sometimes, experts (such as the surgeon completing the procedure) may provide feedback about how long they think a procedure may take. Their estimates about the procedure duration can be recorded. We would like to use the information hidden in their estimates. For instance an expert may be conservative in the sense that expert's estimates always overestimate the actual duration but those estimates are correlated with the realizations. Based on the distribution of the estimates we can eliminate some data points which are not likely to repeat. Those estimates can be:

1. **Categorical:** Categorical estimates have multiple categories with natural ordering such as long, medium and short.

2. **Numerical:** Numerical estimates are most of the time integers, e.g., 2 hours 10 minutes.

Some experts are more effective at estimating the procedure time than others. Sometimes those estimates may not be correlated with the actual procedure duration. Our goal is to understand if there is any information in the expert estimates so that the data points which are unlikely to occur can be disregarded.

### Categorical Estimates

Experts provide their categorical estimates about how long the next procedure will take. They pick one of the $K$ groups which seems to be the most likely. For instance if there are 3 groups such as short, medium and long, the expert will pick one of those three.

We can find a point estimate for all groups with a linear regression after dummy coding the expert categorical estimates. In other words we can create $K-1$ binary dichotomous features representing the groups except for one not to cause multi-collinearity. $A$ is the matrix holding the binary features and the intercept, where the first column is all 1's and the remaining columns represents a group $(\mathtt{g}_2, \ldots, \mathtt{g}_K)$. The total number of observations is $N$ and the numbers of observations (procedure durations) in corresponding groups are $(N_{\mathtt{g}_2}, \ldots, N_{\mathtt{g}_K})$. The linear regression model predicting the procedure duration $x$ is:

$$x = A\beta + \epsilon$$

The closed form solution of the least square estimate of $\beta$ (Wasserman, 2013):

$$\hat{\beta} = (A^T A)^{-1} A^T x$$

where $A^T A$ is:

$$A^T A = \begin{bmatrix} N & N_{\mathtt{g}_2} & N_{\mathtt{g}_3} & \ldots & N_{\mathtt{g}_K} \\ N_{\mathtt{g}_2} & N_{\mathtt{g}_2} & 0 & \ldots & 0 \\ N_{\mathtt{g}_3} & 0 & N_{\mathtt{g}_3} & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ N_{\mathtt{g}_K} & 0 & 0 & \ldots & N_{\mathtt{g}_K} \end{bmatrix}$$

The number of observations falling onto the group $\mathtt{g}_1$ is equal to $N_1 = N - N_2 - N_3 - \cdots - N_K$. The group $\mathtt{g}_1$ is omitted not to cause multicollinearity. It can be shown that the inverse of $(A^T A)$ to be equal to:

$$(A^T A)^{-1} = \begin{bmatrix} \frac{1}{N_{\mathtt{g}_1}} & \frac{-1}{N_{\mathtt{g}_1}} & \frac{-1}{N_{\mathtt{g}_1}} & \cdots & \frac{-1}{N_{\mathtt{g}_1}} \\ \frac{-1}{N_{\mathtt{g}_1}} & \frac{1}{N_{\mathtt{g}_2}} + \frac{1}{N_{\mathtt{g}_1}} & \frac{1}{N_{\mathtt{g}_1}} & \cdots & \frac{1}{N_{\mathtt{g}_1}} \\ \frac{-1}{N_{\mathtt{g}_1}} & \frac{1}{N_{\mathtt{g}_1}} & \frac{1}{N_{\mathtt{g}_3}} + \frac{1}{N_{\mathtt{g}_1}} & \cdots & \frac{1}{N_{\mathtt{g}_1}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{-1}{N_{\mathtt{g}_1}} & \frac{1}{N_{\mathtt{g}_1}} & \frac{1}{N_{\mathtt{g}_1}} & \cdots & \frac{1}{N_{\mathtt{g}_K}} + \frac{1}{N_{\mathtt{g}_1}} \end{bmatrix}$$

The coefficients of the linear regression is:

$$\hat{\beta} = \left( \frac{1}{N_{\mathtt{g}_1}} \sum_{i \in \mathtt{g}_1} x_i, \frac{1}{N_{\mathtt{g}_2}} \sum_{i \in \mathtt{g}_2} x_i - \frac{1}{N_{\mathtt{g}_1}} \sum_{i \in \mathtt{g}_1} x_i, \cdots, \frac{1}{N_{\mathtt{g}_K}} \sum_{i \in \mathtt{c}_K} x_i - \frac{1}{N_{\mathtt{g}_1}} \sum_{i \in \mathtt{g}_1} x_i \right)$$

We need to test the significance of the coefficients $\beta$ which requires making assumptions about the distribution of the procedure durations in each group.

Assuming $K$ groups are independent and normally distributed with same variance there is another method to test the means of the categories are same. Analysis of Variance (ANOVA) is a linear model to test differences among group means in a sample by only computing the mean of each group. ANOVA and linear regression explained above are equivalent (Eisenhauer, 2006). Both use the observations of a group to test the significance of that group.

Assuming all the coefficients in the linear regression are statistically significant, given the next procedure is in group $\mathtt{g}_k$ $k \in \{1, \ldots, K\}$, the point estimate of the next procedure's duration is the average of all observations falling into the group $\mathtt{g}_k$. In other words any observation is equally likely to occur. Instead of a single point estimate we can use all the data points falling into the group $\mathtt{g}_k$ to build the empirical distribution, and minimize the expected weighted earliness and tardiness (2.1) over this empirical distribution.

We would like to avoid making any assumption about the distribution of the procedure durations in each group. We need to test whether the sample from the category $\mathtt{g}_k$ $k \in \{1, \ldots, K\}$ is different than all other samples. The permutation test is a non-parametric method for testing whether two distributions are the same (Wasserman, 2013).

**Algorithm 4:** Permutation Test (Wasserman, 2013)

1 Compute the test statistic $t_{obs} = T(x)$ using the observations.
2 Randomly permute the data and compute the statistic again using the permuted data.
3 Repeat the previous step $B$ times (Monte Carlo approximation) and let $T_1, \ldots, T_B$ denote the resulting values. (Or repeat the previous step for every possible permutation (exact test).)
4 The approximate p-value is

$$\hat{p_0} = \frac{1}{B} \sum_{j=1}^{B} \mathbb{I}(T_j > t_{obs})$$

.

Computation of $p_0$ by doing the exact test might not be computationally feasible. The standard deviation of $\hat{p_0}$ is $\sqrt{\frac{p_0(1-p_0)}{B}}$ (Efron and Tibshirani, 1994) which follows from:

$$Var(\hat{p_0}) = Var\left( \frac{1}{B} \sum_{j=1}^{B} \mathbb{I}(T_j > t_{obs}) \right) = \frac{B p_0(1-p_0)}{B^2} = \frac{p_0(1-p_0)}{B}$$

We would like to minimize the standard deviation to achieve desired precision (Golland et al., 2005):

$$\min_B \max_{p_0} \sqrt{\frac{p_0(1-p_0)}{B}} = \min_B \sqrt{\frac{1}{4B}}$$

Marozzi (2004) investigates how the choice of $B$ affects the estimation procedure, suggests approaches about choosing $B$.

If there is enough evidence that the distribution of a procedure's observations in different groups are not the same while computing the contribution of that procedure to the objective function (2.1) given the group of the next procedure's expert estimate, we can use only the observations from that group. We would like to determine if the distribution of the observations falling into different groups are the same. If there is not enough evidence to reject the hypothesis that the observations in two different groups are coming from the same distribution, we merge those two groups and create a new one. First we order the groups of the categorical expert estimates in decreasing mean of the observations. We test whether two neighboring groups are coming from the same distribution. If there is not enough evidence to reject the null hypothesis of them being coming from the same distribution, then we form a new group which contains observations coming from both groups. Otherwise those groups shouldn't be merged. We are making multiple hypothesis tests so we need multiple testing correction, such as the Bonferroni Method or the Benjamini-Hochberg Method (Benjamini and Hochberg, 1995).

The data selection algorithm if there is a single categorical feature is the Algorithm 5.

### 2.4.2 Feature Selection

In this section we assume there are multiple ($m$) categorical features. Similar to what we have done in the previous section, we can find a point estimate of the next procedure duration by building a linear regression model, after dummy coding all $m$ features:

$$x = \sum_{m=1}^{M} A_m \beta_m + \epsilon$$

The closed form solution of the estimates of the coefficients $\beta_1, \ldots, \beta_M$ is

$$[\hat{\beta}_1, \ldots, \hat{\beta}_M] = ([A_1 A_2 \cdots A_M]^T [A_1 A_2 \cdots A_M])^{-1} [A_1 A_2 \cdots A_M]^T x.$$

But we cannot further simplify the coefficients as in the previous chapter. This is because after dummy coding, each data point has features equal to 0 or 1. Since there is more than one feature, each data point (row of the feature matrix $[A_1 A_2 \cdots A_M]$) may contain more than one 1.

We redefine a linear regression model with new features which are the indicators of the Cartesian product of the $M$ features. For instance the data has two features which are surgeon and anesthesiologist. If there are $K_s$ surgeons and $K_a$ anesthesiologists we can create

**Algorithm 5:** Data Selection If There is a Single Categorical Feature

Name the groups of the categorical expert estimates such that $\mathtt{g}_1$ is the group which has the largest mean, $\mathtt{g}_2$ has the second largest mean...

$$\frac{1}{N_{\mathtt{g}_1}}\sum_{i\in\mathtt{g}_1} x_i \geq \frac{1}{N_{\mathtt{g}_2}}\sum_{i\in\mathtt{g}_2} x_i \geq \cdots \geq \frac{1}{N_{\mathtt{g}_K}}\sum_{i\in\mathtt{g}_K} x_i$$

$$\mu_{\mathtt{g}_1} \geq \mu_{\mathtt{g}_2} \geq \cdots \geq \mu_{\mathtt{g}_K}$$

Set $i = 1$, $j = 2$ and define new group $G_i$ such that $G_i = \mathtt{g}_1$.

**while** $j \leq K$ **do**

  Perform a permutation test to test whether distributions of $G_i$ and $\mathtt{g}_j$ are the same with the test statistic:

$$T(x) = \frac{1}{N_{\mathtt{G}_i}}\sum_{i\in\mathtt{G}_i} x_i - \frac{1}{N_{\mathtt{g}_j}}\sum_{i\in\mathtt{g}_j} x_i$$

  **if** *The difference is significantly large* **then**

    $i = i + 1$

    $G_i = \mathtt{g}_j$

    $j = j + 1$

  **end**

  **else**

    $G_i = G_i \cup \mathtt{g}_j$

    $j = j + 1$

  **end**

**end**

Return new groups, $G_i$'s.

$K_s \cdot K_a$ features representing each surgeon, anesthesiologist pairs. After such modification and dummy coding each row of the new feature matrix contains at most one 1. Thus we can simplify the coefficients of the model and the estimate of a procedure duration given the surgeon and the anesthesiologist would be equal to the mean of the procedure durations done by the same surgeon and the anesthesiologist. This model is a nonlinear model of the the original features which can be represented with a decision tree.

Frank and Witten (1998) proposes an algorithm to choose the features while building a classification decision tree. At each node of the decision tree a feature is chosen to split on. If a feature does not show any significant association to the class at a prespecified significance level, that feature is rejected. In other words if the classes are independent of a feature then that feature is not considered while building the tree. To judge the significance of a feature Frank and Witten (1998) apply a permutation test.

We adapt the idea of building a classification tree using a permutation test to building a regression tree with categorical features:

1. Use the top-down greedy approach: at each node consider all the features which have not been considered in the parent nodes and create new groups as in the Algorithm 5. If the number of groups created is equal to 1 for all features, stop.

2. Choose the feature such that the resulting tree has the lowest residual sum of squares (RSS).

**Examples**

We create dummy data sets where there are only three columns: procedure duration, expert opinion, and gender.

1. Data set 1 after dummy coding:

| | |
|---|---|
| Duration: | 111,120,122,125,126,126,127,128,130,132,133,135,135,136,137,139,140, 142,143,145,145,146,147,150,151,151,155,163,165,166,166,170,178,180, 185,190,198,199,202,209 |
| Medium: | 0,1,0,0,1,0,0,1,0,1,0,1,1,0,1,1,0,0,1,1,1,0,1,0,1,0,0,0,1,0,1,0,0,0,0,0,0 |
| Long: | 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,1,1,1,0,0,0,1,1,1,1,1,1 |
| Female: | 1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0 |

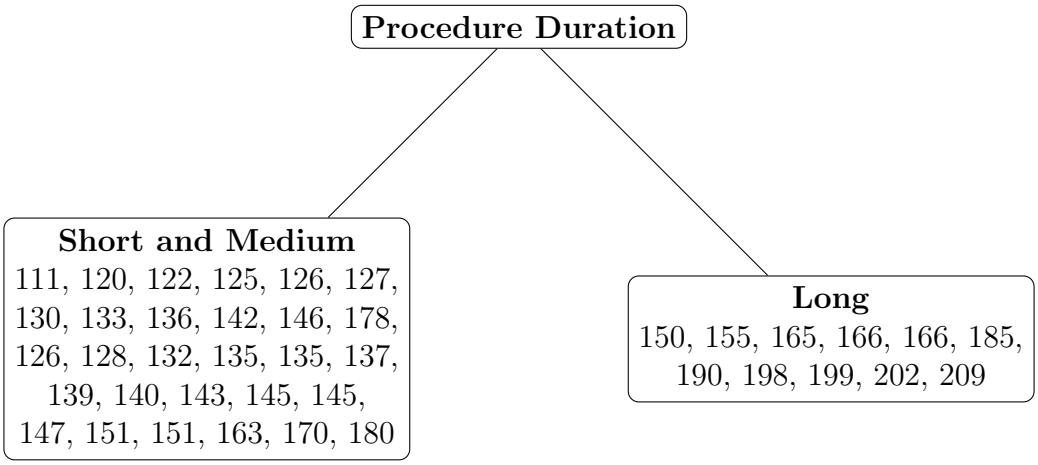



Figure 2.3: Decision tree built using the feature selection algorithm using the data set 1.

2. Data set 2 after dummy coding:

| | |
|---|---|
| Duration: | 111,120,122,125,126,126,127,128,130,132,133,135,135,136,137,139,140, 142,143,145,145,146,147,150,151,151,155,163,165,166,166,170,178,180, 185,190,198,199,202,209 |
| Medium: | 0,1,0,0,1,0,0,1,0,1,0,1,1,0,1,1,0,0,1,1,1,0,1,0,1,1,0,1,0,0,0,1,0,1,0,0,0,0,0,0 |
| Long: | 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,1,1,1,0,0,0,1,1,1,1,1,1 |
| Female: | 0,0,0,1,1,0,1,1,0,0,0,1,0,1,0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,1,1,1,1,1,1 |

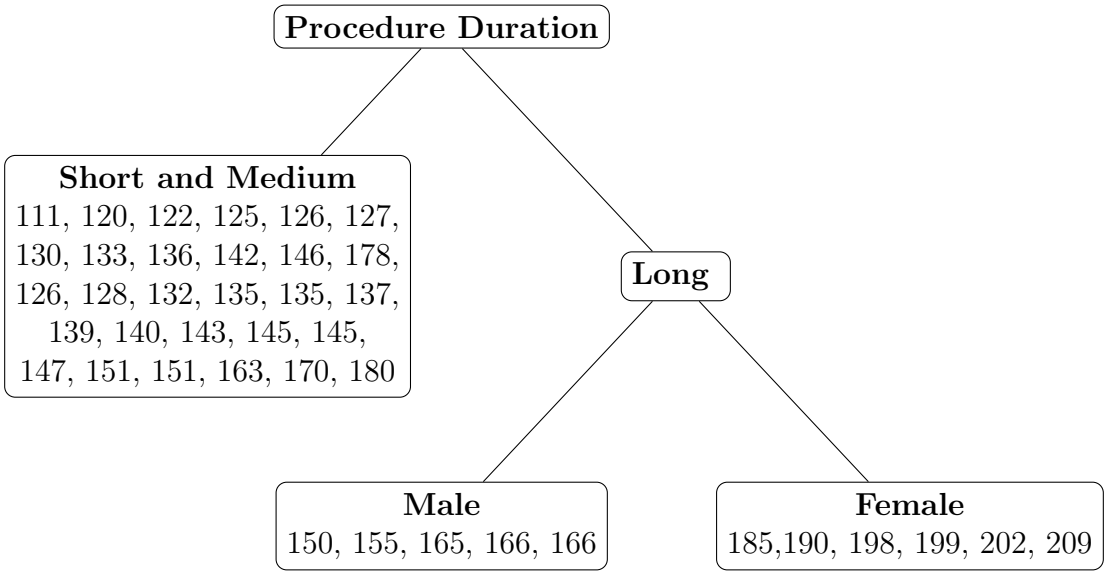



Figure 2.4: Decision tree built using the feature selection algorithm using the data set 2.

In the first example above, the procedure duration distribution does not depend on the gender of the patient. Thus, gender is not considered while branching. In the second example, at the first step we split the feature of expert opinion into two regions given by the permutation test, because this split gives lower RSS.

# Chapter 3

# Conclusion

This dissertation focuses on appointment scheduling of stochastic tasks, which in turns involves both sequencing the tasks and setting the start time of those tasks. A task waits to start if the previous task ends after its scheduled start time. If a task ends before the start time of the next task, the server stays idle. We define tardiness as waiting time and earliness as idle time. The objective of the appointment scheduling problem is to minimize the expected weighted tardiness and earliness. We avoid making any distributional assumptions about the task duration distributions, and use the historical data of task lengths to compute the expected weighted tardiness and earliness. Our work is motivated by an initial project (with UCSF) that focused on surgical scheduling where the tasks are the surgical procedures. Thus we use task and procedure interchangeably.

We study the most commonly used heuristic to sequence the procedures in the appointment scheduling literature, sequencing the procedures in increasing standard deviation of their durations, and its performance for the case where there are only two procedures. We develop sequencing heuristics tailored to appointment scheduling. The sequencing heuristics have different complexities and performances. The first heuristic we develop is polynomial, and we also bound the worst case performance of this sequencing heuristic. Also, we propose heuristics with better performance but increased complexity.

The objective function is a continuous, convex, piece-wise linear function. If the sequence of the tasks is known, the complexity of evaluating the objective function given the historical data and the schedule either depends on the number of observations or the maximum range of the procedure durations, depending on the method used. We propose an algorithm to select data points if there are categorical features correlated with procedure durations, so that the complexity of the problem might decrease. For instance, categorical surgeon estimate regarding the length of the next procedure could be the only categorical feature. We use our data selection algorithm to choose data points for all procedures, then use our sequencing heuristic to determine procedure sequences. Finally we optimize the objective function given the sequence and the data.

Assuming the same set of procedures are scheduled every day, we prove that the scheduled time allowance of a procedure cannot be smaller than the critical quantile of its duration's

distribution. For instance, after the arrival of new data point, if the scheduled end time of that procedure is less than the critical quantile of the updated distribution, we need to update the scheduled time allowance also.

In the future, we plan to develop an online algorithm that updates the schedule and the sequence of the procedures as new data points arrive, assuming that the same set of procedures are scheduled every day. We also plan to study the bias and and consistency of our data selection algorithm.

# Bibliography

Artzner, P., Delbaen, F., Eber, J.-M., Heath, D., et al. (1999). Coherent measures of risk. *Mathematical Finance*, 9(3):203–228.

Bailey, N. T. (1952). A study of queues and appointment systems in hospital out-patient departments, with special reference to waiting-times. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 185–199.

Ban, G.-Y. and Rudin, C. (2018). The big data newsvendor: Practical insights from machine learning. *Operations Research*, 67(1):90–108.

Begen, M. A. (2010). *Appointment scheduling with discrete random durations and applications*. PhD thesis, University of British Columbia.

Begen, M. A., Levi, R., and Queyranne, M. (2012). Technical note-a sampling-based approach to appointment scheduling. *Operations research*, 60(3):675–681.

Begen, M. A. and Queyranne, M. (2011). Appointment scheduling with discrete random durations. *Mathematics of Operations Research*, 36(2):240–257.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300.

Bertsimas, D. and Thiele, A. (2005). A data-driven approach to newsvendor problems. Technical report, Technical report, Massechusetts Institute of Technology, Cambridge, MA.

Boyd, S., Xiao, L., and Mutapcic, A. (2003). Subgradient methods. *lecture notes of EE392o, Stanford University, Autumn Quarter*, 2004:2004–2005.

Cayirli, T. and Veral, E. (2003). Outpatient scheduling in health care: a review of literature. *Production and Operations Management*, 12(4):519–549.

Choi, B.-C., Min, Y., and Park, M.-J. (2019). Strong np-hardness of minimizing total deviation with generalized and periodic due dates. *Operations Research Letters*, 47(5):433–437.

Chu, L. Y., Shanthikumar, J. G., and Shen, Z.-J. M. (2008). Solving operational statistics via a bayesian analysis. *Operations Research Letters*, 36(1):110–116.

Dantzig, G., Fulkerson, R., and Johnson, S. (1954). Solution of a large-scale traveling-salesman problem. *Journal of the operations research society of America*, 2(4):393–410.

Denton, B. and Gupta, D. (2003). A sequential bounding approach for optimal appointment scheduling. *Iie Transactions*, 35(11):1003–1016.

Denton, B., Viapiano, J., and Vogl, A. (2007). Optimization of surgery sequencing and scheduling decisions under uncertainty. *Health care management science*, 10(1):13–24.

Efron, B. and Tibshirani, R. J. (1994). *An introduction to the bootstrap*. CRC press.

Eisenhauer, J. G. (2006). How a dummy replaces a student's test and gets an f (or, how regression substitutes for t tests and anova). *Teaching Statistics*, 28(3):78–80.

Erdogan, S. A., Gose, A., and Denton, B. T. (2011). On-line appointment sequencing and scheduling. Technical report, Working paper, Edward P. Fitts Department of Industrial and Systems Engineering, North Carolina State University, Raleigh NC.

Eriksson, K., Estep, D., and Johnson, C. (2013). *Applied mathematics: Body and soul: Calculus in several dimensions*. Springer Science & Business Media.

Frank, E. and Witten, I. H. (1998). Using a permutation test for attribute selection in decision trees.

Ge, D., Wan, G., Wang, Z., and Zhang, J. (2013). A note on appointment scheduling with piecewise linear cost functions. *Mathematics of Operations Research*, 39(4):1244–1251.

Ghouila-Houri, A. (1962). Caractérisation des matrices totalement unimodulaires. *Comptes Redus Hebdomadaires des Séances de l'Académie des Sciences (Paris)*, 254:1192–1194.

Golland, P., Liang, F., Mukherjee, S., and Panchenko, D. (2005). Permutation tests for classification. In *International Conference on Computational Learning Theory*, pages 501–515. Springer.

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

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

Jiang, R., Shen, S., and Zhang, Y. (2015). Distributionally robust appointment scheduling with random no-shows and service durations. *Available at SSRN 2653622*.

Kaandorp, G. C. and Koole, G. (2007). Optimal outpatient appointment scheduling. *Health Care Management Science*, 10(3):217–229.

Kong, Q., Lee, C.-Y., Teo, C.-P., and Zheng, Z. (2013). Scheduling arrivals to a stochastic service delivery system using copositive cones. *Operations Research*, 61(3):711–726.

Kuwaranancharoen, K. and Sundaram, S. (2018). On the location of the minimizer of the sum of two strongly convex functions. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 1769–1774. IEEE.

Liyanage, L. H. and Shanthikumar, J. G. (2005). A practical inventory control policy using operational statistics. *Operations Research Letters*, 33(4):341–348.

Mak, H.-Y., Rong, Y., and Zhang, J. (2014). Appointment scheduling with limited distributional information. *Management Science*.

Mancilla, C. and Storer, R. (2012). A sample average approximation approach to stochastic appointment sequencing and scheduling. *IIE Transactions*, 44(8):655–670.

Marozzi, M. (2004). Some remarks about the number of permutations one should consider to perform a permutation test. *Statistica*, 64(1):193–201.

Nesterov, Y. (1983). A method for unconstrained convex minimization problem with the rate of convergence o (1/kˆ 2). In *Doklady AN USSR*, volume 269, pages 543–547.

Polyak, B. T. (1964). Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17.

Qian, N. (1999). On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151.

Rockafellar, R. T. and Uryasev, S. (2000). Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42.

Rockafellar, R. T. and Uryasev, S. (2002). Conditional value-at-risk for general loss distributions. *Journal of banking & finance*, 26(7):1443–1471.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

Sarin, S. C., Sherali, H. D., and Liao, L. (2014). Minimizing conditional-value-at-risk for stochastic scheduling problems. *Journal of Scheduling*, 17(1):5–15.

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

Wang, Z., Glynn, P. W., and Ye, Y. (2016). Likelihood robust optimization for data-driven problems. *Computational Management Science*, 13(2):241–261.

Wasserman, L. (2013). *All of statistics: a concise course in statistical inference*. Springer Science & Business Media.

Weiss, E. N. (1990). Models for determining estimated start times and case orderings in hospital operating rooms. *IIE transactions*, 22(2):143–150.

Welch, J. and Bailey, N. J. (1952). Appointment systems in hospital outpatient departments. *The Lancet*, 259(6718):1105–1108.