STAFFING FOR HOUSEKEEPING OPERATIONS

A Thesis

Presented to the Faculty of the School of Hotel Administration

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Master of Science

by

Buyun Li

August 2020

ABSTRACT

Motivated by hotel housekeeping staffing challenges, we study a model in which reusable resources require service between intervals of use by successive customers. For each day, the firm decides how many servers to work a shift and when each should begin working. Customers arrive and depart randomly over the course of the day. The firm's objective is to minimize the combined staffing cost and cost of guest waiting.

We use a sample average approximation approach to solve this problem as an integer linear program and provide structural properties for a simple version of the problem. We test the resulting policy on a discrete-event simulation using both stylized and empirical hotel guest arrival and departure data. We show that the optimization methods based scheduling heuristics can save up to 17% of the total cost compared to a common industry staffing heuristic. With less or equal labor force than the common staffing, the schedule heuristics we propose can lower the guest wait time by up to 80% at a higher guest flow level and 50% at a lower level of guest flow. The optimization methods covered by this paper can be applied by hotel, airport, car rental managers to save the labor costs and improve service readiness.

BIOGRAPHICAL SKETCH

Buyun Li is a doctoral student at Kelley School of Business, Indiana University. His academic focus is on service operation research. Staffing and scheduling problem, in particular, is the concentration of his study. Before his doctoral studies, he was a Master of Science at The Hotel School at Cornell University. Buyun has two bachelor degrees in business from both Queensland University and Beijing International Studies University. Along with the academic career, Buyun is also the COO of EarlyBirds LLC.. which is a technology company focusing on the restaurant revenue management.

This document is dedicated to family and friends.

ACKNOWLEDGEMENTS

The thesis would never be possible without my advisor, Professor Slaugh's inspiration and guidance. I can not express my appreciation enough to his kind and wise mentorship for the entire course of this thesis and my study at Cornell.

My completion of the thesis would never be possible without the support from my industry partners. It was them who provide me intuitions and basic knowledge to the subject. I would acknowledge Professor Dev's efforts in setting up this connection.

Finally, to my loving and caring partner Ruimin Zhang and parents, it was you who kept my courage when facing setbacks and it was you who motivates me when I am frustrated. The support from you are the most beautiful parts of my past two years' study.

On top of everything, I deeply appreciate those who fought with COVID-19 at the front-line. It was your bravery and sacrifice that my family and I are safely sheltered from the virus.

CONTENTS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The successful operation of a hotel depends largely on the management of housekeeping processes. Decisions related to the housekeeping workforce can have a significant impact on the hotel's profitability and the customer experience. However, hotels and other service providers face increasing challenges due to rising labor costs and diminished worker availability. Few models have connected tactical workforce decisions to customer wait times. Motivated by hotel housekeeping, we present a model of reusable resources that require attention from a service workforce and study staffing decisions to improve service readiness.

In particular, hotels face an important trade-off between housekeeping labor costs and the room readiness wait time that guests experience. Kandampully and Suhartanto (2003) report that the performance of the housekeeping department, including the responsiveness of the housekeeping satff and cleanliness of rooms, is deemed as the most significant factor for brand image and customer satisfaction. The explicit cost associated with the guest waiting mostly appears to be the compensation to guests who experienced waiting. Anecdotally, one rooms manager of a large center-city hotel reported to us that the hotel's standard practice was to offer a $50 in food and beverage credit if a room was not available for a guest upon arrival after the stated check-in time. They increased compensation of one-half that day's room rate for more significant delays or other special circumstances with guests. Many of these guests came to the hotel to attend weddings and related events; a delay in room readiness often meant that the guest had to change attire in hotel restrooms rather than the guest's room.

The United States Bureau of Labor Statistics (2020) reports that the hotel industry in the United States employs about half a million people — almost one out of every four workers in the industry — who earning a mean hourly wage of $12.76. The hourly wage in the housekeeping department increased by nearly

10% from 2016 to 2019, according to United States Bureau of Labor Statistics (2019), and the turnover rate is at a high level of 79.9% (United States Bureau of Labor Statistics 2020). In an article about the tight labor market for hotel housekeepers, the *New York Times* quotes one general manager, who said, "Everyone I speak to in the industry is having trouble getting housekeeping staffed. It's always been one of the hardest jobs to fill, and harder than ever now (Weed 2019)."

In this paper, we focus on the daily decisions related to housekeeper staffing, and connect those decisions to customer wait times. Managerial decisions in a hotel's housekeeping process include: (1) how many housekeepers should the hotel employ, and (2) what should the daily schedule be? A popular rule deployed by managers is to hire the number of housekeepers equals to the number of guestrooms divide by 13 or 14, which is the number of rooms that a housekeeper can clean in a shift, and have most of the housekeepers keep regular hours from 8 AM to 5 PM (Branson and Lennox 1988).

Emerging hotel industry trends in guest stay patterns underscore the importance of understanding housekeeping operations. Major hotel brands, including Marriott, Hilton, and IHG, have offered various types of flexible check-in and check-out policies. Some offer this as guaranteed amenity for loyalty program members, and some sell flexible stay policies as an opportunity to increase revenue. Other hotels, such as luxury resorts where guests are already paying a high room rate, simply offer early check-in and late check-out based on availability. An operations executive of one luxury resort that attracts visitors from around the world reported to us that many guests expect to have a room ready upon their arrival, regardless of posted check-in times. Hotels connected to casinos also prioritize flexibility for high-spending guests. Another trend is a 24-hour check-in/check-out policy, where the guest may check in at any time of day and occupy the room for the next 24 hours (Yahoo 2015). In today's culture, the flexibility of arriving and leaving becomes a general amenity in the hotel industry to create a unique experience of customized services for guests. The implementation of this concept is a result of guests' demands that hotels are realizing that they have to fulfill to remain in business and stay relevant (Yahoo 2015). Customer loyalty, increased profit, and customer satisfaction are major incentives for hotels implementing this policy (Suri 2015, Trejos 2019).

Programs presented as environmentally friendly actions for guests to take may increase the variability in the number of rooms to clean each day and the cleaning time of each room. Many hotels is offering guests incentives to skip their daily room cleaning, promoting the program as a "green" choice that reduces the use of cleaning chemicals, energy and water. However, Weed (2019) reported that housekeeping departments are facing operational challenges in coping these programs because extra work occurs when skipped room finally needs cleaning, and the managers are less certain about the number of housekeepers needed.

Despite the appealing competitive advantages provided by flexible arrival and departure policies and housekeeping labor reduction strategies, the implementation of these models poses significant operational challenges for hotel operations. We are the first to approach this problem with queuing model and provide analytical properties. Jones (2014) identifies several key operational decisions that would benefit from deeper analysis: (i) How long late (early) should the early check-in (late check-out) period be extended? (ii) How many guests should be allowed to check-in early (check-out late)? (iii) How should housekeepers be scheduled to adopt the new check-in and check-out patterns?

We study this problem using a model in which the completion of a customer's time using a resource is analogous to the breakdown of a server, which then needs to be repaired before use by another customer. Both the customer departure times and the repair process affect the customer queue. We present a discrete-time stochastic model of this system that does not require assumptions on the arrival or departure distributions. We use sample-path methods to develop important system properties.

We are the first to formalize stochastic and integer programming models of housekeeping operations, and we contribute analytical properties, heuristics, and numerical insights for staffing decisions. Specifically, we demonstrate the expected profit is concave in the number of housekeepers starting at one time — a helpful property for guiding the search for an optimal solution using simulation. The following managerial insights also emerge from numerical analysis in which we compare our heuristics to a common scheduling strategy at an upscale airport hotel:

1. Our proposed scheduling heuristic can reduce labor costs by 15% while maintaining the same level of customer waiting. Our results indicate that the hotel should reduce the total number of workers while shifting some workers who start in the morning to start in the afternoon.
2. Alternately, without increasing the number of housekeepers to schedule, our scheduling heuristic can eliminate 90% of total guest waiting time by starting worker later in the morning.
3. We identify alternate optimal solutions that allow managers to select a schedule that best matches the needs and preferences of the workforce, which can mitigate the labor shortage that the hotel faces.

CHAPTER 2

LITERATURE REVIEW

The housekeeper staffing problem that we define most closely resembles work in the operations management literature on the combined problems of determining staffing requirements and tour scheduling for personnel. Call center staffing motivates much of the research with this focus and has been well-studied, starting with A. C. Erlang (Brockmeyer 1948). Buffa et al. (1976) propose the following commonly-used step-wise approach to call center scheduling:

1. Forecast demand rates by period.
2. Develop a period-by-period cost function based on these forecasts, usually relying on stationary analysis of a queuing system.
3. Determine the set of feasible shifts.
4. Select a shift that minimizes the total cost in terms of customer wait times and staffing costs.

(Green et al. 2001) later named this approach as "stationary independent period-by-period" (SIPP) and demonstrated its usefulness for making staffing recommendations. However, the ability to keep an inventory of cleaned rooms makes our problem fundamentally different from call center staffing; e.g., the optimal strategy to reduce wait times of guests arriving during the evening "rush hour" may be to start housekeepers' shifts earlier in the morning. Furthermore, the fluctuations in departure and arrival rates throughout the day make it difficult to develop a stationary distribution of the hotel housekeeping process, especially if one tries for a sufficiently high number of servers to achieve stability in each period. Additionally, the need to serve both changeover and stayover rooms introduces dynamic decision-making into the problem as hotel managers decide which class of rooms for housekeepers to clean over the course of their shifts.

The operational dynamics of the housekeeping problem bear some similarity to models of machine breakdown and repair, which date to White and Christie (1958). Hotel rooms correspond to machines that break down (i.e., when a guest departs), and housekeepers are like repairmen who render the room suitable for occupation. Neuts and Lucantoni (1979) studied the relationship between the size of the repair crew and the queue length under Markovian assumptions. Tang (1997) explored the machine breakdown and repair model with a single server but relaxed assumptions on the service time distribution. By introducing the notion of a negative queue length, Moinzadeh and Aggarwal (1997) developed a production-inventory system with machine breakdowns and deterministic repair time to study the throughput of the model. Based on this work, Sabri-Laghaie et al. (2012) presented search algorithms to find the optimal number of repair crews. Delasay et al. (2012) offered strategies to maximize the throughput with stochastic breakdown and repair times. However, the highly non-stationary arrival and departure processes, as well as the finite horizon, of our problem limit the applicability of these approaches.

Another application similar to hotel housekeeping is the admissions and discharge process for hospitals. Shi et al. (2015) demonstrated the impact of discharge times on inpatient (check-in) delay due to bed availability. Based on this study Zychlinski et al. (2019) applied a fluid model in modelling the total cost of inpatient flow congestion problem in a hospital setting. Gaughan et al. (2015) examined the association between the number of nurses and the bed blocking time using regression models. Kim and Giachetti (2006) proposed overbooking strategies to decrease wait times. A key difference in this setting is that health care facilities have a high level of control over inpatient and discharge processes, while the hospitality industry is subject to more uncertainty in guest behavior(Zygourakis et al. 2014). Furthermore, while these papers model bed availability, they do not explicitly model cleaning crews or decisions related to their operation.

Methodologically, we use a sample path approach to analyze hotel housekeeper staffing due to the problem's operational complexity. Sample-path analysis has played an increasingly important role in certain applied probability problems. In particular, researchers have found advantages to analyzing queueing models based on realized sample paths (El-Taha and Stidham Jr 2012). Stidham (1982) argued that sample-path method can provide "distribution-free" analysis between variables and performance measures in the finite horizon setting. Margolius (1999) used sample path method to solve steady states of time dependent arrival queues. Sigman and Yamazaki (1992) analyzed properties of overloaded. Sample-path analysis has often been applied to generalize the arrival and service assumptions of the queuing models. In the operations literature, Kapuściński and Tayur (1998) found the optimal inventory policy of the single product, single-stage capacitated production-inventory model using a sample-path approach. Slaugh et al. (2016) used sample path approach to prove the concavity of the expected profit function and identified the optimal recirculation rule under different models in a rental managing settings. Freund (2018) analyzed the problem of allocating bicycles among stations in a bike-sharing system using sample path framework.

We are the first to explicitly model housekeeping operations and provide analytical results, although some related operational decisions in the hotel context have received attention. Bitran and Gilbert (1996) modeled reservation acceptance decisions and presented heuristics to decide how many rooms to allocate to "walk-in" customers. Soltani and Wilkinson (2010) has studied how hotels could use flexible housekeepers to enhance its performance on housekeeping efficiency. Motivated by front-desk operations, Thompson and Goodale (2006) presented a tour scheduling approach for a service workforce with heterogeneous productivity rates among workers. Sari (2017) measured housekeeper performance by a fuzzy score-based model. Kadry et al. (2017) simulated the housekeeping process for a hotel using discrete-event simulation software to reduce customer waiting and staffing costs but did not describe or analyze a formal model. Wood et al. (2005) described metrics for the performance of the hotel housekeeping operations through audit questionnaires.

CHAPTER 3

MODELS AND ANALYTICAL RESULTS

3.1 Models

We represent the hotel housekeeping process using a discrete-time model with equal-length periods over a finite time horizon to keep track of the transition of the room states. Each room is in one of the four states: occupied, vacant dirty, cleaning in process, or vacant clean. Before the guest checks out of the hotel room, the state of the room is *occupied*. The event of a guest checking out changes the state of the room from occupied to *vacant dirty*, which means the room is available for housekeepers to clean. Once an available housekeeper starts to clean the vacant dirty room, the room is updated to *cleaning in process*. Simultaneously, the housekeeper who started to clean the room is no longer available until after some time interval when cleaning is finished. After the housekeeper finishes cleaning, the room will turn to state *vacant clean* and the housekeeper is available again. Finally, upon guest arrival the hotel will assign a vacant clean room to guests if available. Otherwise, the guest will wait in a queue. Hotels generally use a first-come first-served discipline for assigning rooms to waiting guests. For simplicity, we consider all the rooms to be interchangeable, and all guest arrivals can be assigned to any rooms.

3.2 Deterministic Service Time Model with Single Start Time

We first introduce a model that assumes a deterministic cleaning time and housekeepers whose shifts extend for the entire time horizon. After defining our variables and state equations, we prove a relationship between wait time and the staffing level. The following table lists all variables used in this initial model.

Table 3.1: Notation

Notation	Description
D_n	Number of departures at n^{th} period
A_n	Number of arrivals at n^{th} period
y	Number of housekeepers
h	Units of time needed to clean a room
$V_n^c(y)$	Number of vacant clean rooms at the beginning of n^{th} period with y housekeepers
$V_n^D(y)$	Number of vacant dirty rooms at the beginning of n^{th} period with y housekeepers
$R_n(y)$	Number of available housekeepers at the beginning of n^{th} period with y housekeepers
$W_n(y)$	Number of guests waiting at the beginning of n^{th} period with y housekeepers

State Equations. The state equations specify the state transition from one period to the next. In the state equations, y is the decision variable representing the staffing level over the entire horizon. In each period $n = 1, 2, \ldots$, the number of departures and arrivals are defined as D_n and A_n, respectively. Each housekeeper requires h periods to clean one room. We focus on W_n, the queue length at the end of each period, as our primary performance measure.

At the beginning of each period, we first update the number of available housekeepers. Within any h periods, the cleaning process can start for no more than y rooms, and any job started in previous $y-1$ periods will not be finished during the current period. Therefore, the number of available servers equals to the difference between the total number of servers and the number of busy servers. We represent the number of vacant dirty rooms in each period n for a system with y housekeepers as $V_n^D(y)$ — which we may write as V_n^D for convenience — and use

$R_n(y)$ to represent the number of housekeepers available. Our equation for $R_n(y)$ captures the number of rooms for which cleaning could begin.

Available housekeepers at the start of period n**:**

$$R_n(y) = y - \sum_{i=n-h+1}^{n-1} min\{V_i^D, R_i\} \tag{3.1}$$

Number of vacant dirty rooms at start of period n**:**

$$V_n^D(y) = D_n + [V_{n-1}^D - R_{n-1}]^+ \tag{3.2}$$

The number of vacant clean rooms in period n, denoted by $V_n^C(y)$, represents the number of available rooms to assign guests. Similar to $V_n^D(y)$, the number of vacant clean rooms consists of two parts. The first part is the number of rooms for which cleaning began h periods ago completes at the beginning of period n. The second part is the number of clean rooms left from previous period.

Number of vacant clean rooms at the start of period n**:**

$$V_n^C(y) = min[V_{n-h}^D, R_{n-h}] + [V_{n-1}^C - A_{n-1} - W_{n-2}]^+ \tag{3.3}$$

The queue length represents the number of arrived guests who are waiting for a room at the end of period n, which we can also represent as the difference between the number of vacant clean rooms and the demand for the vacant clean rooms in period n. The demand includes the newly arrived guests and the waiting guests from the previous period.

Queue length at the start of period n**:**

$$W_n(y) = [W_{n-1} + A_n - V_n^C]^+ \tag{3.4}$$

The system begins with no vacant dirty or vacant clean rooms; i.e., $V_0^D(y) = V_n^C(y) = 0$. The number of rooms that can be started equals the minimum of the number of guest departures and the number of housekeepers that are available to work. Also, all guests that arrived during this period will need to join the queue as no vacant clean rooms are available.

3.3 Queue Length Convexity With Regards to Number of Servers

In this section, we use induction to show the convexity of the queue length $W_n(y)$ in each period n as a function of number of housekeepers. In the proof process, the base case assumption will be that the queue length is convex for period $1, 2, \ldots, n-1$, and the object is to prove period n is convex.

Base Case Proof. For $i \in [1, h]$, the average queue length is independent of the number of housekeepers. To be more specific:

$$W_i(y) = W_i(y+1) = W_i(y+2) = \sum_{j=1}^{i} A_j \qquad i \in [1, h], y \geq 0 \tag{3.5}$$

We know that none of the arriving guests can be checked-in, hence $W_h = \sum_{j=1}^{h} A_j$. Based on this fact, we can insert W_h into formula of W_{h+1} to get a cleaner form of the queue length. For period $h+1$, solving the queue length with y housekeepers:

$$\begin{aligned} W_{h+1}(y) &= [W_h(y) + A_{h+1} - V_{h+1}^C]^+ \\ &= [\sum_{i=1}^{h+1} A_i - min(V_1^D, R_1)]^+ \\ &= [\sum_{i=1}^{h+1} A_i - min(D_1, y)]^+ \end{aligned} \tag{3.6}$$

solving the queue length of $y+1$ housekeepers with indicator function:

$$W_{h+1}(y+1) = [\sum_{i=1}^{h+1} A_i - min(D_1, y) + I_1]^+, \tag{3.7}$$

solving the queue length of $y + 2$ housekeepers with indicator function:

$$W_{h+1}(y+2) = [\sum_{i=1}^{h+1} A_i - min(D_1, y) + I_1 + I_2]^+ \tag{3.8}$$

$$I_1 = \begin{cases} 1, & \text{if } D_1 \geq y+1. \\ 0, & \text{otherwise.} \end{cases} \quad I_2 = \begin{cases} 1, & \text{if } D_1 \geq y+2. \\ 0, & \text{otherwise.} \end{cases} \tag{3.9}$$

Given the fact that $I_2 = 1 \iff I_1 = 1$, the queue length is point-wise convex at h_1 as equation below holds:

$$W_{h+1}(y) - W_{h+1}(y+1) \geq W_{h+1}(y+1) - W_{h+1}(y+2) \tag{3.10}$$

Until this point, we have proved the convexity for period $1 \ldots h + 1$.

Induction Steps. Given that the clean process takes h period for all periods, we can know that the queue length is equivalent to the difference of arrivals and number of rooms that has finished being cleaned, which equals to the work started before $n - h$ periods. Formally, we claim that:

Claim 1.

$$W_n = [\sum_{i=1}^{n} A_i - \sum_{j=1}^{n-h} \min(V_j^D, R_j)]^+ \tag{3.11}$$

Based on the same logic, we could also rewrite the number of vacant dirty rooms since the number of vacant dirty rooms at period n equals to the all departures up to period n subtract with all work has been started before.

Claim 2.

$$V_n^D = \sum_{i=1}^{n} D_i - \sum_{j=1}^{n-1} \min(V_j^D, R_j) \tag{3.12}$$

Given the base case assumption that W_{n-1} is concave and the claim 1, we claim that the $\sum_{j=1}^{n-h-1} \min(V_j^D, R_j)$, the number of rooms started, is concave since the $\sum_{i=1}^{n} A_i$ is irrelevant of y and W_{n-1} is convex. Axiomatically, for all period earlier than period $h - 1$, concavity holds for the total number of rooms started.

Claim 3. $\sum_{j=1}^{k} \min(V_j^D, R_j)$ *is concave* $\forall k \leq n-1$.

Based on the information provided by the base case assumption, we can write out the queue length at period$n$ as follow:

1. Rewrite W_{n-1} then plug in W_n based on claim 1,

$$W_n = [W_{n-1} + A_n - V_n^C]^+$$
$$= [[\sum_{i=1}^{n} A_i - \sum_{j=1}^{n-h-1} \min(V_j^D, R_j)]^+ + A_n - V_n^C]^+$$

2. Expand the term V_n^C based on claim 2,

$$= [[\sum_{i=1}^{n-1} A_i - \sum_{j=1}^{n-h-1} \min(V_j^D, R_j)]^+ + A_n - \min(V_{n-h}^D, R_{n-h}) - [\sum_{j=1}^{n-h-1} \min(V_j^D, R_j) - \sum_{i=1}^{n-1} A_i]^+]^+$$

3. Combine the first term with the last term,

$$[\sum_{i=1}^{n} A_i - \sum_{j=1}^{n-h-1} \min(V_j^D, R_j) - \min(V_{n-h}^D, R_{n-h})]^+$$

4. Expand the term V_{n-h}^D and R_{n-h} based on equation (3.1)

$$\{\sum_{i=1}^{n} A_i - \sum_{j=1}^{n-h-1} \min(V_j^D, R_j) - \min[\sum_{i=1}^{n-h} D_i - \sum_{i=1}^{n-h-1} \min(V_i^D, R_i), y - \sum_{n-2h}^{n-h-1} \min(V_i^D, R_i)]\}^+$$

5. Break the term $\sum_{j=1}^{n-h-1} \min(V_j^D, R_j)$ into $\sum_{j=n-2h}^{n-h-1} \min(V_j^D, R_j) + \sum_{i=1}^{n-2h-1} \min(V_i^D, R_i)$, then move those two part inside the minimum function to cancel out the term $\sum_{j=1}^{n-h-1} \min(V_j^D, R_j)$,

$$\{\sum_{i=1}^{n} A_i - \min[\sum_{i=1}^{n-h} D_i, y + \sum_{i=1}^{n-2h-1} \min(V_i^D, R_i)]\}^+$$

Given the claim 3, $\sum_{i=1}^{n-2h-1} \min(V_i^D, R_i)$ is concave with respect to y, we know that $y + \sum_{i=1}^{n-2h-1} \min(V_i^D, R_i)$ is a concave function with respect to y. D_i is invariant of y, hence $\min([\sum_{i=1}^{n-h} D_i, y + \sum_{i=1}^{n-2h-1} \min(V_i^D, R_i)])$ is a concave function of y. $\sum_{i=1}^{n} A_i$ is a constant, therefore the W_n is a convex function with respect to the y provided that the base assumption is true.

Given that the the base case is true, and if the $W_{1...n-1}$ is convex, W_n is convex, the convexity is proved for all periods by induction.

Average Waiting Time Convexity With Regards to Number of Servers

Given that the queue length is a convex function with respect to the number of housekeepers. We argue that the average time that the guests spending in the queue is convex. According to the model, the total wait time experienced by all guests (T) equals the sum of queue length at each period times the period length(τ). Formally, we make the claim that:

Claim 4.

$$T = \sum_{i=1}^{n} W_i(y) \cdot \tau \tag{3.13}$$

Given that τ is a irrelevant of y, and $W_i(y)$ is a convex function, T must be a convex function with regards to the number of housekeepers τ

3.4 Restricted Stochastic Service Time Model

In this section, we relax the assumption of deterministic service times to accommodate stochastic service times. This model captures randomness in the time required to clean a room. However, we also must consider the room selection decision for choosing among the vacant dirty rooms each period — e.g., whether random, shortest job first, or longest job first. We formulate the stochastic service time model with the shortest job first ordering, which minimizes waiting time among room selection policies. Adding the cleaning time for each room requires adding a dimension to our state variables for each sample path, and we use the following notation:

Table 3.2: Notation for the Stochastic Service Time Model

Notation	Description
$D_{i,k}$	Number of departures in period i that needs k units of time to clean
A_i	Number of arrivals in the i^{th} period
y	Number of housekeepers
$x_{i,k}$	Number of rooms for which cleaning lasting k periods begins in period i
R_i	Number of available housekeepers at the beginning of the period i
q_i	Queue length in period i

$D_{i,k}$, A_i are non-negative parameters, and y is our only decision variable, given that we are assuming the job at the same time will start follow some principle.

Number of jobs to start each period. Since we have specified the shortest job first ordering, the number of rooms that can be cleaned at each time will depend on both jobs that arrived in previous periods and jobs that arrived at the same period but need a short time to finish.

$$x_{i,k} = \min\{[R_i - \sum_{l=1}^{k-1} x_{i,l}]^+, [\sum_{j=0}^{i} D_{j,k} - \sum_{m=0}^{i-1} x_{m,k}]^+\} \tag{3.14}$$

Number of available housekeepers. The number of available housekeepers at the beginning of the period is the difference in the number of total housekeepers and the number of housekeepers that are busy, which equals the number of rooms that have been started but not finished.

$$R_i = y - \sum_{j=0}^{i} \sum_{l=i-j+1}^{\infty} x_{j,l} \tag{3.15}$$

Queue length. The number of people in the queue at the end of period i equals the difference in the cumulative number of arrivals and the number of rooms that have been cleaned.

$$q_i = \{\sum_{j=0}^{i}[A_j - \sum_{l=0}^{i-j} x_{j,l}]\}^+ \tag{3.16}$$

CHAPTER 4

OPTIMIZATION METHODS: SAMPLE AVERAGE APPROXIMATION AND INTEGER LINEAR PROGRAMMING

In this chapter, we will introduce two optimization methods for solving problems related to those presented in the previous chapter. We extend these models to incorporate two key features. First, the decision variable is now a tour-planning problem for determining how many housekeepers start in each period. Second, the hotel must also clean a certain number of its room for guests who are *stayovers*, that is, neither arriving or departing on the day in question. As each housekeeper becomes available to clean a room, the hotel must choose whether to assign the housekeeper to a changeover room or a stayover room.

4.1 Deterministic Service Time Model

We begin with a basic model where the cleaning time for each room is deterministic. We first introduce data and parameters needed in the model, then introduce the objective function and constraints. Finally, we describe a simulation model for the process.

4.1.1 An Integer Programming Approach

Data and Parameters. We consider a hotel with k rooms that have equal numbers of guests checking in/out in a same day. Also, other than cleaning the changeover rooms, the housekeeper has to service all the stayover guest rooms by a certain deadline. According to the setting above, the following data and parameters constitute a sample-path:

Ω: the total number of scenarios

D_i^{ω}: number of guests checked out at time i in scenarios ω

A_i^{ω}: number of guests checked in at time i in scenarios ω

$Cost_L$: Labor cost per housekeeper per shift

$Cost_W$: Penalty cost per unit of time per guest waiting in queue

H: Expected units of time needed per cleaning

Z: Number of stayovers needed to be cleaned

T_d: Number of time units of a day

T_z: Deadline for stayovers to be cleaned

T_s: Number of time units per working shift

The arrival and departure data can be collected by point-of-sales software. The majority of hotels keep an record on when does the guests arrive at the hotel. The departure time, however, can be tricky to collect accurately as guests can check out without notifying the front desk. However, in this paper, we argue that it would not affect the accuracy of our model since it does not affect the changeover process. In addition to the changeover, it is necessary that we control stayover cleaning process in our model.

Decision Variable. We introduce the following decision variable:

y_i = number of housekeepers that should be started at time period i.

x_i^{ω} = number of room changeovers to start at period i,in scenarios ω.

z_i^{ω} = number of stayovers that should be be started at period i, in scenarios ω

Other than the major decision variables — when and how many housekeeper to start — the integer linear program also has two other sets of decision variables: the number of changeovers and stayovers to start in each period. This decision has to be re-considered at each time period as the dynamics of this decision alters according to the time. For example, the changeover is prioritized at the beginning of the day, however, the priority of the stayover cleanings will rise as closer to the deadline. Partitioning of the decision variables in such a way allows for more dynamic and flexible modeling.

States. To consistently update the variables for all time periods, we keep track of the following states:

Q_i^{ω} = number of people in the queue at time period i,in scenarios ω.

R_i^{ω} = number of housekeepers available at time period i, in scenarios ω.

$V_i^{C,\omega}$ = number of vacant clean rooms at time period i, in scenarios ω.

$V_i^{D,\omega}$ = number of vacant dirty rooms at time period i, in scenarios ω.

The Integer Program Formulation. We formulate and solve an integer program that maximizes the objective over the average of r sample paths, and determine from the optimal solution how housekeepers should be scheduled to begin their shift in each period. The proposed model is as follows:

Objective Function:

$$\min\left(\sum_{i=1}^{T_d}\sum_{\omega=1}^{\Omega}\frac{Cost_W Q_i^{\omega}}{\Omega}+Cost_L y_i\right) \tag{4.1}$$

The objective function is to minimize the total cost of the housekeeping schedule, which involves two parts: a penalty cost for having guests waiting in the queue, and a labor cost for each shift scheduled.

Constraints: The number of the rooms that can be started at time period i is subject to two constraints: a labor constraint and an availability constraint for the number of dirty vacant rooms that can be cleaned.

(1) Labor Constraint:

$\forall i \in 1..T_d, \forall \omega \in 1..\Omega$:

$$x_{i,r} + z_{i,r} \leq R_{i,r} \tag{4.2}$$

$$R_i^\omega = \sum_{i-t_s}^{i} y_i^\omega - \sum_{[l=i-H]^+}^{i-1} x_i^\omega - \sum_{[l=i-H]^+}^{i-1} z_i^\omega \tag{4.3}$$

(2) Room Constraints

$\forall i \in 1..t_d, \forall \omega \in 1..\Omega:$

$$x_i^\omega \leq \sum_{j=0}^{i} D_i^\omega - \sum_{j=0}^{i-1} x_i^\omega \tag{4.4}$$

The labor constraints ensure that the number of rooms to start in each period does not exceed the available labor capacity. The room constraints ensure that the number of rooms to start at each period does not exceed number of dirty vacant rooms at each period.

(3) Stayover Constraints

$\forall r \in 1..Rep:$

$$\sum_{i=0}^{T_z-H} z_i^\omega = Z \tag{4.5}$$

The stayover constraints guarantee that all stayover rooms are cleaned before the deadline.

(4) Queue Length

$\forall i \in 1..t_d, \forall \omega \in 1..\Omega:$

$$q_i^\omega = \sum_{j=0}^{i} A_j^\omega - \sum_{k=0}^{i-h} x_k^\omega \tag{4.6}$$

The optimal solution, y, is a vector that keeps track of the number of housekeepers we should start at different time periods. By limiting the size of this vector, we could also change number of shifts allowed in total.

In this basic model for hotel housekeeping staffing, we also need to decide the length of each time period. With a shorter length of the time period, the model gain the advantage of being more flexible and dynamic. However, the trade-off of choosing a shorter time horizon is that the solving time for the IP is longer as more decision basis will be added to the IP.

Initial States. Since the variables updated by the state equations are directly dependent on that of previous period, values for the initial states need to be specified in order to get a consistent model.

$$R_1^\omega = y_1^\omega \tag{4.7}$$

$$W_1^\omega = A_1^\omega \tag{4.8}$$

The Simulation

The integer program described can be solved whenever the hotel needs a staffing plan based on a forecast of the guest arrival and departure pattern. In the implementation, we could first fit the time distribution of the arrivals and departures, then generate a large number of sample paths to ensure that the optimal solution accommodates the randomness of the arrival/departure dynamics. If simulating for the stochastic service time model, a service time should be generated and assigned to each departures.

In each time period, there are six events drive the model. To update the states consistently, we specify the sequence of events to clarify state transitions in Table (4.1.1).

Table 4.1: Sequence of Events

Sequence	Events
1	Cleaning finished, update available housekeepers R_i^ω
2	Add rooms just finished cleaning to the inventory
3	Guests depart, update number of vacant dirty rooms
4	Assign housekeeper to vacant dirty rooms, get x_i^ω
5	Guest arrive, get A_i^ω
6	Assign cleaned rooms to guests, update queue length Q_i^ω

4.2 Stochastic Service Time Integer Programming Model

One of the disadvantage of the basic model is its sole use of the expected service time as an indication of the service time for each room. The stochastic version of the IP introduces various service time associated with each guest departure and stayover cleaning. By adding another dimension to the variables, the IP attempts to more accurately capture the future demand characteristics.

The stochastic service time model works with Γ service times. For each departure or stayover cleaning, there is a stochastic service time $\gamma \in 1..\Gamma$ associated with it. γ is an integer indicating the number of periods needed to finish this job. With new dimension of the service time, we modify some of the variables as follow:

Modified Data. $D^{\omega}_{i,\gamma}$: Number of departures at time period i that requires γ units of time to clean in scenario ω.

Modified Decision Variables $x^{\omega}_{i,\gamma}$: Number of changeovers to start at time period i that requires γ units of time to clean in scenario ω.

$z^{\omega}_{i,\gamma}$: Number of stayovers to start at time period i that requires γ units of time to clean in scenario ω.

Objective function. The objective function remains the same with the objective function in the basic model.

Constraints.

(1) Labor constraint $\forall\omega \in 1..\Omega, \forall i \in 1..T_d$:

$$x^{\omega}_{i,\gamma} \leq R^{\omega}_i - \sum_{l=1}^{\gamma-1} x_{i,l} \tag{4.9}$$

$$R^{\omega}_i = \sum_{i-T_s}^{i} y^{\omega}_i - \sum_{j=0}^{i} \sum_{\gamma=i-j}^{\Gamma} x_{j,\gamma} - \sum_{j=0}^{i} \sum_{\gamma=i-j}^{\Gamma} z_{j,\gamma} \tag{4.10}$$

(2) Room Constraint $\forall\omega \in 1..\Omega, \forall i \in 1..T_d$:

$$x^{\omega}_{i,\gamma} \leq \sum_{j=0}^{i} D^{\omega}_{j,k} - \sum_{m=0}^{i-1} x^{\omega}_{m,\gamma} \tag{4.11}$$

(3) Stayover Constraint $\forall \omega \in 1..\Omega$:

$$Z^{\omega} = \sum_{i=0}^{T_z} \sum_{\gamma=0}^{T_z - i} z_{i,\gamma}^{\omega} \tag{4.12}$$

(4) Queue length $\forall \omega \in 1..\Omega, \forall i \in 1..T_d$:

$$Q_i^{\omega} \geq \sum_{j=0}^{i} \left(A_j - \sum_{\gamma=0}^{i-j} x_{j,\gamma} \right) \tag{4.13}$$

$$Q_i \geq 0 \tag{4.14}$$

The initial states of the stochastic service model is same as the basic model.

4.2.1 The Simulation for the Stochastic Model

The simulation using the stochastic service cleaning time is similar to that of the basic model. The sequence of the events driving the model is identical to that of the basic model. The important difference is that we generate a stochastic service time for each room upon a guest departure. However, in order to be able to express the process consistently, the cleaning order has to be specified. For simplicity, in the simulation, we choose the shortest job first ordering to start to clean the vacant dirty room at the beginning of each period.

CHAPTER 5

NUMERICAL EXAMPLES AND COMPUTATIONAL RESULTS

In this section, we describe a numerical example based on stylized data to illustrate analytical results descried in §3.1 and the performance of the optimization method described in §3.2. Then, using empirical data from an upper-scale airport hotel, we simulate the performance of both models on multiple metrics. We use a discrete event simulation to evaluate the performance of staffing plans. In both examples, we assume that there is are vacant clean/dirty room left from the previous day.

5.1 Simulated Data Example

In this section we demonstrate a hotel scenario using stylized data to illustrate the models. We specify the data for arrival, departure, service time, cost parameters and guest flow by using known distributions.

Arrival Data. The timestamps for guest arrival follow a normal distribution with mean of μ_A = 5 P.M. and a standard deviation of σ_A = 4 hours. Additionally, any arrival times outside of the range of 4:01 AM and 11:59 PM are re-sampled.

Departure Data. The timestamps for guest departures follow a normal distribution with mean of μ_D = 10 P.M. and a standard deviation of σ_D = 3 hours. Additionally, the departure times are within the range of 12:01 AM and 7:59 PM.

Service Data. The timestamps for guest departures follow a normal distribution with mean of μ_S = 30 minutes and a standard deviation of σ_S = 5 minutes.

Cost Parameters. Labor cost per housekeeper per shift is $250, and the penalty cost for having a guest waiting is $1 per minute.

Guest Flow. To comprehensively present the performance of the algorithm, we displayed the stylized example in two different guest flow levels: a busy example with 400 check-ins and check-outs and a light day with 200 guests checking in and checking out. For each example, we assume there are no dirty vacant/clean rooms left from the previous day and that 300 stayover cleanings must to be performed by 5 P.M..

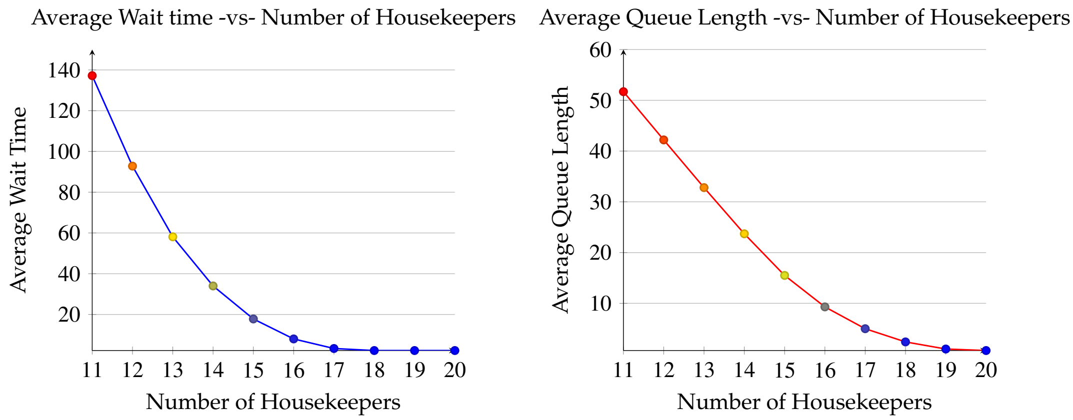
Based on the specified data and parameters, we generate 100 sample paths and input them into both models to output the calculated optimal solution of the housekeeping schedules. Then, we test the schedules to another 500 independent iterations of the discrete event simulation to measure the performance of the different schedules. Solving the optimal schedule over multiple iterations of sample path will yield a schedule that better address the randomness in the guest arrivals and departures processes.

5.1.1 Wait Times and Staffing Levels

Table 4 exhibits the simulation results of wait time and queue length differences with regard to the number of housekeepers. We assumed that all of the housekeepers start their work at the beginning of the day and maintain a full day shift. In the result, both wait time and queue length are convex decreasing function with regard to the number of housekeepers. Average waiting time decreases at a faster rate than queue length when extra housekeeper are added to the system. In this example, at around 18 housekeepers, the wait time reaches the lower bound. Another interpretation is that with 18 housekeepers, the congestion is mostly caused by the late check-out of the previous guests of the room.

Table 5.1: Queue Length with regards to Number of Housekeepers

Number of Housekeepers(y)	11	12	13	14	15	16	17	18	19	20
Average Wait Time	137.2	92.8	58.1	34.0	17.8	8.0	3.3	2.3	2.3	2.3
Average Queue Length	51.7	42.2	32.8	23.7	15.5	9.3	5.0	2.4	1.0	0.7



5.1.2 Computational Results for the Stylized Example

Table 5 shows the performance of schedules calculated by different models under different levels of guest flows on both the training data and test data. The performance metrics include the average total cost, average labor cost, average waiting time, and average queue length. To better illustrate the performance of both models, we add a control schedule that is commonly used by the hotels without schedule optimization. The control scheduling heuristic comes from interviews conducted among the hotel managers. The performance given by the discrete event simulation based on same 500 sample paths. From top to bottom, Table 5 exhibits the performance for the controlling schedule (CS), the schedule calculated by the deterministic service time model (DTST) and the schedule calculated by the stochastic service time model (SCST).

For either guest flow level, we observe that increasing the sophistication of the model provide monotonically lower total costs. At a higher guest flow, DTST schedule provides a 15% cost cut, and STCS provides a 17% cost cut. In the lower guest flow settings, DTST provides a 14.5%cost reduction, and STSC provides a 14% reduction.In the larger guest flow scenario, the DTST model schedule least housekeepers but maintained a similar or better performance than the controlling schedule. Although the schedule calculated by the SCST have a higher labor cost, the performance of the SCST schedule is significantly increased compared to DTSC model. In a lower guest flow scenario, both SCST schedule and DTSC schedule have roughly the same performance. We also observe that the marginal cost saving is decreasing along with the decreasing guest flow. It is also worthwhile noticing that the increasing sophistication of the model results in a decrease in the standard deviation of the total cost.

Table 5.2: Computational Results for Stylized Model with Test Data

Models	**Measure Metrics**			
(Guest Flow = 400, Train Data)	Total Cost ($)	Labor Cost ($)	Wait Time (min/guest)	Queue Length
CS–*(31 housekeepers start at 9:25 AM, 2 start at 12:10 PM, 4 start at 3:50 PM)*				
Mean	12784	10000	4.46	3.13
95% CI	(10127, 13894)	N/A	(2.1, 7.6)	(1.1,4.2)
Standard Deviation	556	N/A	1.3	0.70
DTST–*(31 housekeepers start at 9:25 AM, 6 housekeepers start at 3:10 PM)*				
Mean	11799	9250	3.72	2.88
95% CI	(10303, 13773)	N/A	(2.01,4.74)	(1.03,4)
Standard Deviation	336.14	N/A	0.76	0.41
STCS–*(30 housekeepers start at 9:10 AM, 6 start at 4:15 PM, 2 start at 5:05 PM)*				
Mean	10625	9500	0.91	0.80
95% CI	((9760, 13373))	N/A	(0.58,1.62)	(0.09, 1.24)
Standard Deviation height	90.12	N/A	0.34	0.24
(Guest Flow = 200, Train Data)				
CS–*(20 housekeepers start at 8:30 AM, 3 starts at 1:00PM ,and 2 start at 5:00 PM)*				
Mean	6687	6250	2.18	1.23
95% CI	(6291, 7002)	N/A	(0.82, 3.70)	(0.68,2.91)
Standard Deviation	151.74	N/A	1.18	0.70
DTST–*(18 start at 9:05 AM, 4 start at 2:10 PM, and 1 start at 4:00 PM)*				
Mean	5776	5750	0.13	0.11
95% CI	(5750,5820)	N/A	(0,0.35)	(0,0.28)
Standard Deviation	44.44	N/A	0.14	0.07
STCS–*(1 housekeeper start at 7:30 AM, 17 start at 9:25 AM, 4 start at 2:10 PM, and 1 start at 3:55 PM)*				
Mean	5754	5750	0.02	0.01
95% CI	(5750, 5764)	N/A	(0,0.07)	(0, 0.03)
Standard Deviation height	8.81	N/A	0.024	0.015
(Guest Flow = 400, Test Data)	Total Cost ($)	Labor Cost ($)	Wait Time (min/guest)	Queue Length (guests)
CS–*(35 housekeepers start at 8:30 AM, 3 starts at 1:00PM ,and 2 start at 5:00 PM)*				
Mean	11812	10000	4.53	3.20
95% CI	(10303, 12941)	N/A	(2.3, 7.6)	(1.2,4.5)
Standard Deviation	569.88	N/A	1.3	0.83
DTST–*(31 housekeepers start at 9:25 AM, 6 housekeepers start at 5:10 PM)*				
Mean	11187	9250	3.76	2.91
95% CI	(9840,13911)	N/A	(2.77,4.78)	(1.02,4.01)
Standard Deviation	330.21	N/A	0.78	0.43
STCS–*(30 housekeepers start at 9:10 AM, 6 start at 4:15 PM, 2 start at 5:05 PM)*				
Mean	10993	9500	0.93	0.82
95% CI	(9760, 13648)	N/A	(0.03,1.65)	(0.09, 1.32)
Standard Deviation height	98.2	N/A	0.35	0.24
(Guest Flow = 200, Test Data)				
CS–*(20 housekeepers start at 8:30 AM, 3 starts at 1:00PM ,and 2 start at 5:00 PM)*				
Mean	6674	6250	2.12	1.20
95% CI	(6278, 7211)	N/A	(0.81, 3.82)	(0.68,2.88)
Standard Deviation	278.24	N/A	1.12	0.69
DTST–*(18 start at 9:05 AM, 4 start at 2:10 PM, and 1 start at 4:00 PM)*				
Mean	5800	5750	0.12	0.09
95% CI	(5750,6009)	N/A	(0,0.34)	(0,0.28)
Standard Deviation	45.53	N/A	0.13	0.07
STCS–*(1 housekeeper start at 7:30 AM, 17 start at 9:25 AM, 4 start at 2:10 PM, and 1 start at 3:55 PM)*				
Mean	5754	5750	0.02	0.01
95% CI	(5750, 5766)	N/A	(0,0.07)	(0, 0.03)
Standard Deviation height	19.73	N/A	0.023	0.014

In summary, the stochastic service time model seems to be the best model: it provides the schedule that has lower labor cost, best performance ,and stability of both cost and performance. However, the STSC model adds exponentially more basis to the linear program, which result in a significantly longer run time. The run time for the branch and bound tree gets to 2% within the LP-relaxation optimal value the for the DTST is typically 15 seconds and 28 seconds for SCST at 100 replications while using CPLEX solver.

5.1.3 Queue Length Distribution with regards to Time

Figure 3 exhibits the queue length distributions at a 400 guest flow under different schedules in each hour from 4 A.M. to 12 A.M of the next day. The majority of the waiting happens after 5 P.M for all schedules. As shown in the figure, CS exhibits a two peaks of waiting at 6 P.M. and 10 P.M., whereas DTST scheduling is able to alleviate waiting around the first peak with less housekeepers in total. STSC scheduling keeps very low queue length throughout the day.

Figure 3. Average Queue Length Distribution (Guest Flow = 400)

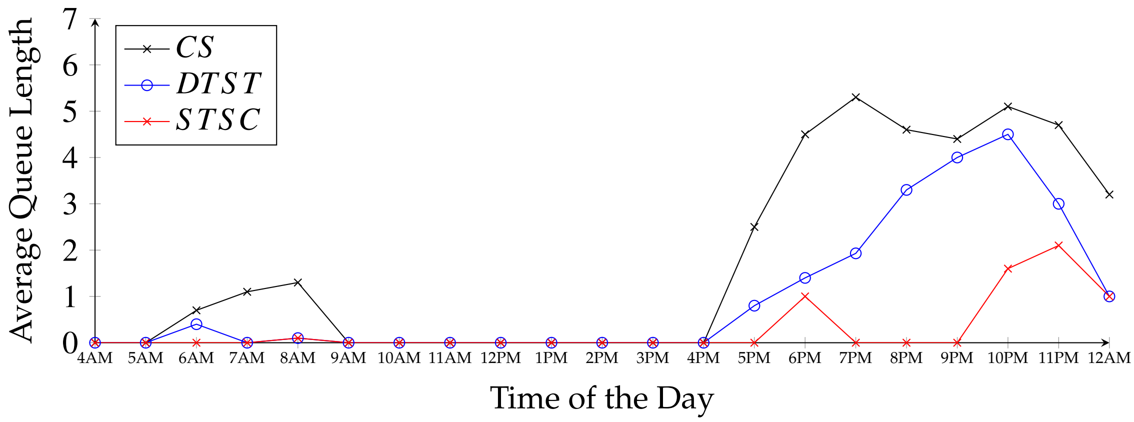
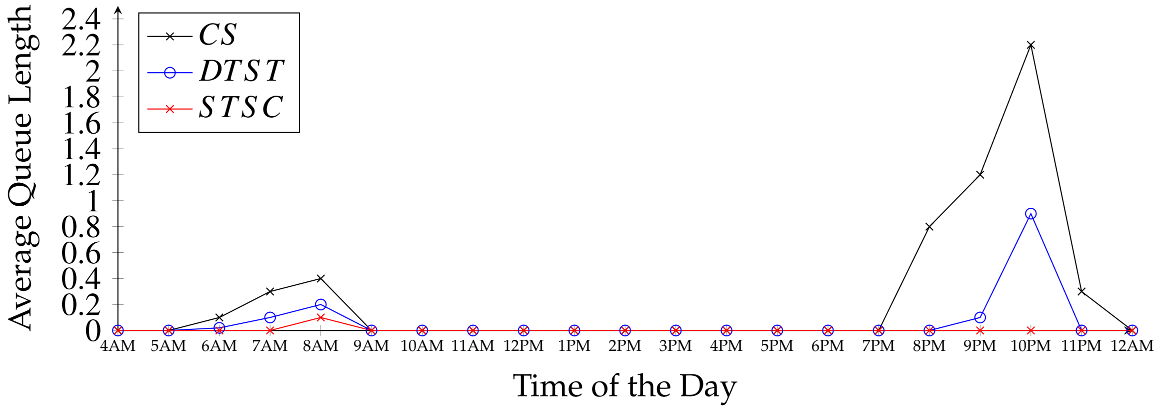


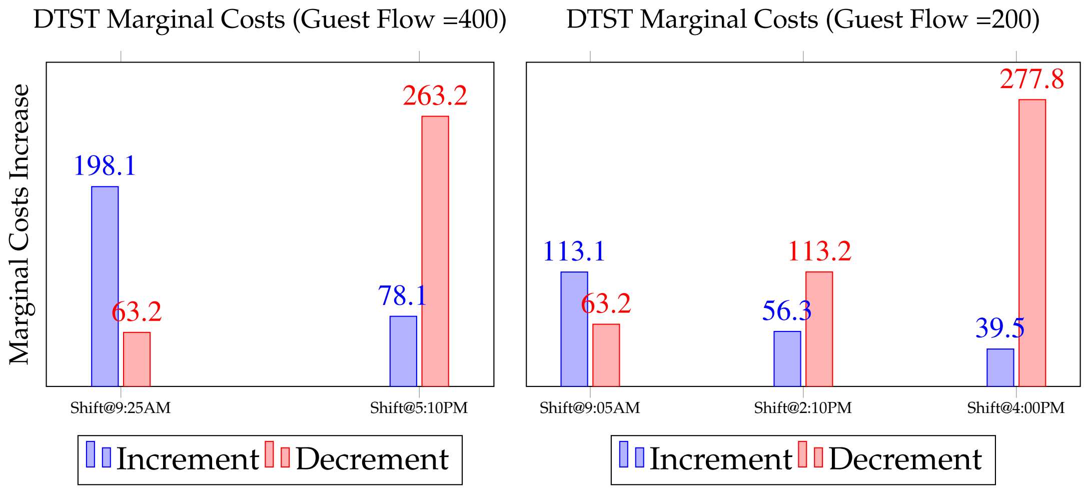
Figure 4 shows the queue length distribution with 200 guests checking in and out. In the 200 guest flow settings, all schedule have better performance regards to queue length. For CS and DTST scheduling, majority of the waiting happens in time window between 9:00 P.M. and 10:00 P.M.. The STSC scheduling is able to keep the queue empty throughout the day except for one time slot early in the morning.

Figure 4. Average Queue Length Distribution (Guest Flow = 200)

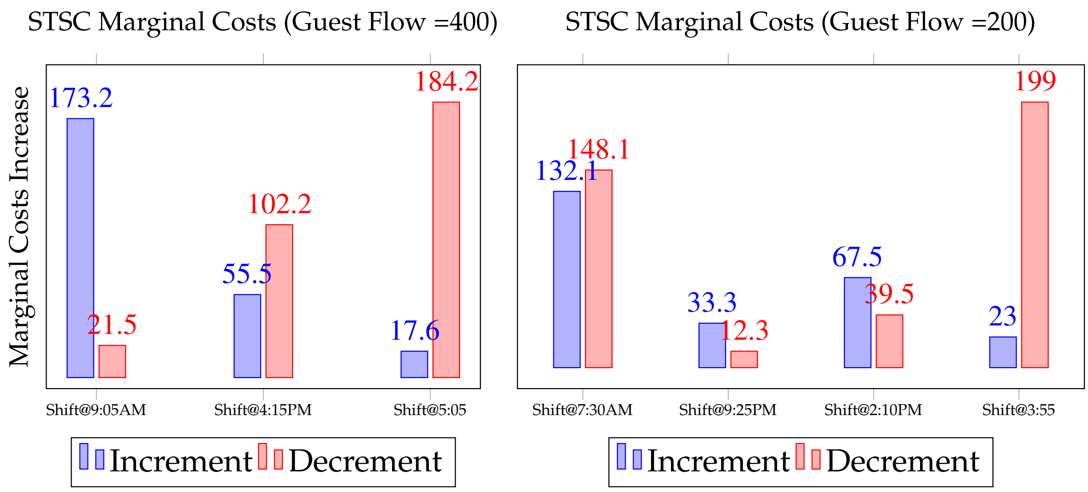


5.1.4 Sensitivity Tests

Marginal costs of the number of housekeepers. Figure 4 exhibits the marginal costs when increase/decrease one housekeeper to each schedule of DTST scheduling at different shifts with different guest flows. We observe that the increment of number of housekeepers results in higher cost increase in the morning, however, the total cost is more sensitive to decrementing the number of housekeepers in the afternoon period. Overall, the total cost DTST scheduling is sensitive to the change of the numeber of housekeepers.



The figure below shows the marginal increased costs for the STSC scheduling. In general, STSC is less sensitive than that of DTST scheduling.



Cleaning Duration Sensitivity

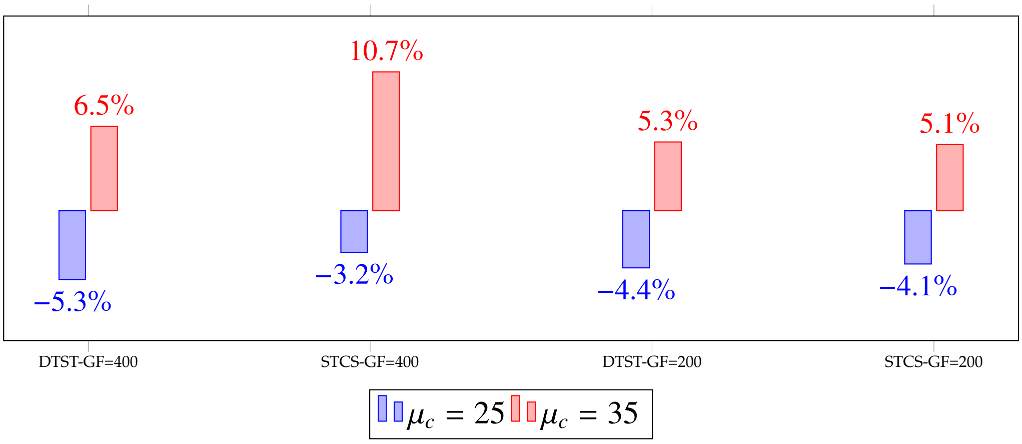
We examined the sensitivity of the performance of each schedule with regard to the change of cleaning duration. Table 6 shows the mean of the total cost, labor cost, average waiting time, and queue length when the cleaning duration is increase or decreased by 5 minutes. The total cost is positively correlated with cleaning duration. We also observe that the all performances of the schedules are more sensitive to the cleaning duration when the guest flow is high.

Table 5.3: Computational Results for Different Cleaning Time with Test Data

Models	**Measure Metrics**			
(Guest Flow = 400, μ_c = 25 mins)	Total Cost ($)	Labor Cost ($)	Wait Time (min/guest)	Queue Length
DTST–*(17 start at 9:15 AM, 15 start at 2:55 PM),4 starts at 4:00 PM*				
Mean	10164	9000	2.91	2.19
95% CI	(9496,11545)	N/A	(0.75,3.86)	(1.03,4)
Standard Deviation	311.16	N/A	0.76	0.41
STCS–*(20 start at 9:00 PM, 14 start at 12:05 PM, 3 start at 4:25 PM)*				
Mean	9546	9250	0.74	0.61
95% CI	(9224, 9898)	N/A	(0.03,1.62)	(0.09, 1.24)
Standard Deviation height	90.12	N/A	0.34	0.24
Guest Flow = 400, μ_c = 35 mins				
DTST–*(1 housekeeper start at 6:50 AM, 28 start at 9:10 PM, 10 start at 2:25 PM)*				
Mean	12442	10000	3.63	2.23
95% CI	(11772,13909)	N/A	(1.87,5.06)	(0.81,4.32)
Standard Deviation	322.12	N/A	0.81	0.77
STCS–*(2 start at 7:15 AM, 17 start at 8:50 AM, 16 start at 9:55 AM, 6 start at 2:55 PM, and 3 start at 5:05)*				
Mean	11952	10500	1.13	0.68
95% CI	(11710,13212)	N/A	(0.55,1.78)	(0.33, 0,98)
Standard Deviation height	191.21	N/A	0.49	0.26
Guest Flow = 200, μ_c = 25 mins	Total Cost ($)	Labor Cost ($)	Wait Time (min/guest)	Queue Length
DTST–*(1 housekeeper starts at 7:35 AM, 19 starts at 9:10 AM, and 2 start at 4:00 AM)*				
Mean	5532	5500	0.08	0.08
95% CI	(5508,5624)	N/A	(0.02,0.31)	(0,0.22)
Standard Deviation	44.14	N/A	0.12	0.06
STCS–*(1 housekeepers start at 7:35 AM, 19 start at 9:10 AM, 1 start at 2:55, 1 start at 4:05 PM)*				
Mean	5520	5500	0.05	0.06
95% CI	(5507,5620)	N/A	(0.02,0.30)	(0,0.21)
Standard Deviation height	39.14	N/A	0.09	0.06
Guest Flow = 200, μ_c = 35 mins				
DTST–*(1 housekeepers start at 7:35 AM, 21 start at 9:00 PM, 1 start at 2:25, 1 start at 4:30 PM)*				
Mean	6084	6000	0.21	0.15
95% CI	(6006,6223)	N/A	(0.015,0.57)	(0,0.28)
Standard Deviation	40.01	N/A	0.14	0.11
STCS–*(1 housekeeper start at 7:25 AM, 20 start at 8:50 AM, 2 start at 1:35 PM, and 1 start at 3:55 PM)*				
Mean	6072	6000	0.18	0.014
95% CI	(6005,6206)	N/A	(0.014,0.51)	(0,0.26)
Standard Deviation height	38.77	N/A	0.13	0.11

Figure 5 shows the percentage change of the mean total cost, wait time, and queue length of either scenarios when compared to the original 30 minutes mean cleaning time scenarios. Both scheduling methods are more sensitive to the the increase of the cleaning duration.

Figure 5. Total Cost Change with regard to Cleaning Duration



Sensitivity Over Scheduling Flexibility

As shown above, a flexible and dynamic schedule can provide cost savings and enhanced performance, however, in the empirical settings, it is impossible to control the shift starting time in such a detailed manner. We explore the effect of relaxed control by only allow shifts start at whole hours (e.g. 5 P.M. , 6 P.M. etc.) Table 7 exhibits the computational results when housekeepers are only allowed to start the whole hours for both models at each guest flow levels.

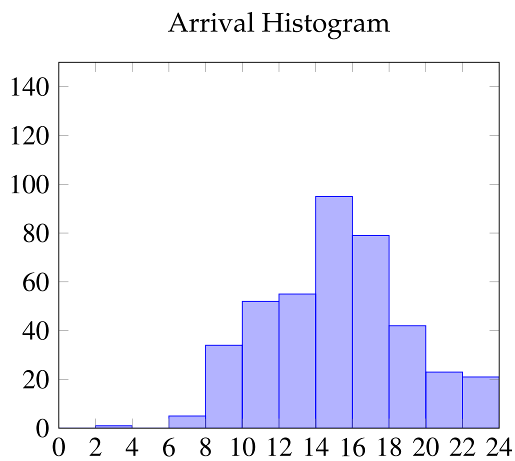
Table 5.4: Computational Results for Shifts Start at Whole Hours with Test Data

Models	**Measure Metrics**			
(Guest Flow = 400)	Total Cost ($)	Labor Cost ($)	Wait Time (min/guest)	Queue Length
DTST–*(1 housekeeper starts at 7:00AM, 24 start at 8:00 AM, 12 start at 11:00 PM, 3 starts at 4:00 PM)*				
Mean	12342	10000	2.5	2.12
95% CI	(10593,14104)	N/A	(1.17,5.26)	(1.03,5)
Standard Deviation	311.16	N/A	0.76	0.71
STCS–*(1 housekeeper starts at 7:00AM, 20 start at 8:00 AM, 5 start at 9:00 PM, 10 starts at 10:00 PM, 5 starts 4:00 PM)*				
Mean	11793	10000	1.98	1.66
95% CI	(10297,12880)	N/A	(0.76,3.45)	(0.93,3.98)
Standard Deviation height	188.32	N/A	0.76	1.21
(Guest Flow = 200)				
DTST–*(1 housekeeper start at 7:00 AM, 20 start at 8:00 AM, 2 start at 3:00 PM, 1 start at 4:00 PM)*				
Mean	6061	6000	0.15	0.08
95% CI	(6022,6101)	N/A	(0.057,0.25)	(0.02,0.17)
Standard Deviation	22.3	N/A	0.81	0.04
STCS–*(1 housekeeper start at 7:00 AM, 20 start at 8:00 AM, 2 start at 3:00 PM, 1 start at 4:00 PM)*				
Mean	6061	6000	0.15	0.08
95% CI	(6022,6101)	N/A	(0.057,0.25)	(0.02,0.17)
Standard Deviation height	22.3	N/A	0.81	0.04

We observe that the sensitivity of the STCS is significantly higher than that of DTST scheduling. STCS schedule's mean cost has raised around 5% when the whole hour constraints is imposed, whereas that of DTST only raised 3%. More interestingly, when the guest flow is at a lower level, the DTST and STCS are offering the same schedule, which has a 4.5% higher cost than the more flexible schedule.

5.2 Computational Results for the Empirical Data Example

The show the applicability in the empirical settings, we report the computational results based on real data from a west coast upper-scale airport hotel. In this specific hotel, the daily average number of arrivals and departures are around 300. Noticeably, this hotel has implemented the early-check-in and late check-out policy. As shown in the histograms below, there are around 40% of the guests checked in before 2 pm and 40% checked out after 12 pm. We uniformly draw 500 sample path fro the empirical data, which has 400 guest flows for each sample-path. Based on those sample paths, we solve for the optimal schedule using both models. Table 8 shows the computational results for the empirical data. The controlling schedule is shown below.



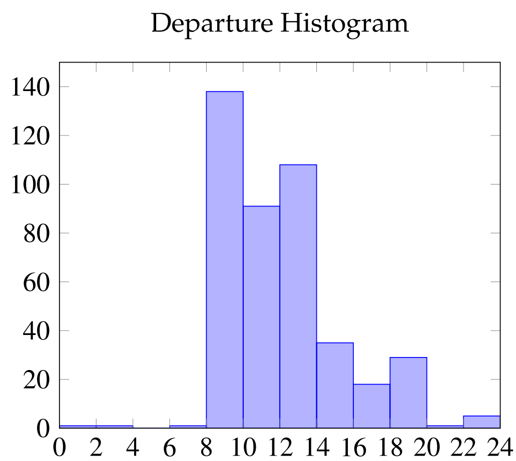


Table 5.5: Computational Results for Empirical Example

Models	**Measure Metrics**			
(Guest Flow = 400)	Total Cost ($)	Labor Cost ($)	Wait Time (min/guest)	Queue Length
DTST				
Mean	12184	10500	4.21	3.32
95% CI	(11332, 12894)	N/A	(2.2, 6.75)	(1.2,5.0)
Standard Deviation	523	N/A	1.3	1.10
STCS				
Mean	11924	11000	2.12	1.97
95% CI	(11232, 12678)	N/A	(1.2, 4.09)	(1.0,3.87)
Standard Deviation height	445	N/A	0.92	1.10
(Guest Flow = 200)				
DTST				
Mean	6128	6000	0.32	0.23
95% CI	(6049,6213)	N/A	(0.13,0.51)	(0.11,0.34)
Standard Deviation	37.2	N/A	0.12	0.04
STCS				
Mean	6099	6000	0.25	0.19
95% CI	(6038,6201)	N/A	(0.08,0.5)	(0.02,0.17)
Standard Deviation height	39.3	N/A	0.09	0.04

Due to the fact that this hotel has adopted the early check-in/late check-out policy, the total cost and the wait time are both larger than that of stylized example. Also, the utilization of the housekeepers has gone down as the number of housekeepers scheduled in the empirical example is larger than that of the stylized data.

CHAPTER 6

FUTURE WORK

In this paper, we have discussed the modeling and analysis of the initial model of the housekeeping process. Although our model has generalized the hotel arrival and departure process, we see the potential directions that should be considered. First, we can study the model under stochastic settings of the service time. Also, according to the on-site investigation. Adding a infinite server queue to approximate the travel time of the housekeepers between cleaning may also be necessary. In housekeeper scheduling, we could explore the various policies regarding to the starting time of the housekeeping shifts. Ultimately, the authors are developing a close form total cost function with respect to the number of housekeepers and their shift schedules under a generalized arrival, departure and service time distributions.

BIBLIOGRAPHY

Gabriel R Bitran and Stephen M Gilbert. Managing hotel reservations with uncertain arrivals. *Operations Research*, 44(1):35–49, 1996.

Joan Cameron Branson and Margaret Lennox. *Hotel, hostel and hospital housekeeping*. Edward Arnold, 1988.

E Brockmeyer. A survey of ak erlang's mathematical works. *Danish Academy of Technical Sciences*, 2:101–126, 1948.

Elwood S Buffa, Michael J Cosgrove, and Bill J Luce. An integrated work shift scheduling system. *Decision Sciences*, 7(4):620–630, 1976.

Mohammad Delasay, Bora Kolfal, and Armann Ingolfsson. Maximizing throughput in finite-source parallel queue systems. *European Journal of Operational Research*, 217(3): 554–559, 2012.

Muhammad El-Taha and Shaler Stidham Jr. *Sample-path analysis of queueing systems*, volume 11. Springer Science & Business Media, 2012.

Daniel Freund. Models and algorithms for transportation in the sharing economy. 2018.

James Gaughan, Hugh Gravelle, and Luigi Siciliani. Testing the bed-blocking hypothesis: does nursing and care home supply reduce delayed hospital discharges? *Health economics*, 24:32–44, 2015.

Linda V Green, Peter J Kolesar, and João Soares. Improving the sipp approach for staffing service systems that have cyclic demands. *Operations Research*, 49(4):549–564, 2001.

MichaelF̃orrest Jones. How lenient are hotels when it comes to checking out late?, Jun 2014. URL https://www.forbes.com/sites/quora/2012/06/13/how-lenient-are-hotels-to-checking-out-late/.

Seifedine Kadry, Aremn Bagdasaryan, and Mohammad Kadhum. Simulation and analysis of staff scheduling in hospitality management. In *2017 7th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO)*, pages 1–6. IEEE, 2017.

Jay Kandampully and Dwi Suhartanto. The role of customer satisfaction and image in gaining customer loyalty in the hotel industry. *Journal of Hospitality & Leisure Marketing*, 10(1-2):3–25, 2003.

Roman Kapuściński and Sridhar Tayur. A capacitated production-inventory model with periodic demand. *Operations Research*, 46(6):899–911, 1998.

Seongmoon Kim and Ronald E Giachetti. A stochastic mathematical appointment overbooking model for healthcare providers to improve profits. *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and humans*, 36(6):1211–1219, 2006.

Barbara Haas Margolius. A sample path analysis of the m t/m t/c queue. *Queueing Systems*, 31(1-2):59–93, 1999.

Kamran Moinzadeh and Prabhu Aggarwal. Analysis of a production/inventory system subject to random disruptions. *Management Science*, 43(11):1577–1588, 1997.

Marcel F Neuts and David M Lucantoni. A markovian queue with n servers subject to breakdowns and repairs. *Management Science*, 25(9):849–861, 1979.

Kamyar Sabri-Laghaie, Mostafa Mansouri, Arash Motaghedi-Larijani, and Gholamreza Jalali-Naini. Combining a maintenance center m/m/c/m queue into the economic production quantity model with stochastic machine breakdowns and repair. *Computers & Industrial Engineering*, 63(4):864–874, 2012.

Irem Ucal Sari. A performance measurement model minimizing housekeeping personnel costs. *International Journal of Fuzzy Systems*, 19(3):776–787, 2017.

Pengyi Shi, Mabel C Chou, JG Dai, Ding Ding, and Joe Sim. Models and insights for hospital inpatient operations: Time-dependent ed boarding time. *Management Science*, 62(1):1–28, 2015.

Karl Sigman and Genji Yamazaki. Fluid models with burst arrivals: A sample path analysis. *Probability in the Engineering and Informational Sciences*, 6(1):17–27, 1992.

Vincent W Slaugh, Bahar Biller, and Sridhar R Tayur. Managing rentals with usage-based loss. *Manufacturing & Service Operations Management*, 18(3):429–444, 2016.

Ebrahim Soltani and Adrian Wilkinson. What is happening to flexible workers in the supply chain partnerships between hotel housekeeping departments and their partner employment agencies? *International Journal of Hospitality Management*, 29(1):108–119, 2010.

Shaler Stidham. Sample-path analysis of queues. In *Applied probability—computer science: the interface*, pages 41–70. Springer, 1982.

Charu Suri. New hotel checkout time: Anytime you want, Sep 2015. URL https://www.nytimes.com/2015/09/20/travel/hotels-late-check-out.html.

Ying Hui Tang. A single-server m/g/1 queueing system subject to breakdowns—some reliability and queueing problems. *Microelectronics Reliability*, 37(2):315–321, 1997.

Gary M Thompson and John C Goodale. Variable employee productivity in workforce scheduling. *European journal of operational research*, 170(2):376–390, 2006.

Nancy Trejos. Check in, check out anytime you'd like at these hotels, Nov 2019. URL https://skift.com/2019/11/21/check-in-check-out-anytime-youd-like-at-these-hotels/.

United States Bureau of Labor Statistics. Annual total separations rates by industry and region, 2019. URL https://www.bls.gov/oes/2016/may/oes372012.htm&https://www.bls.gov/oes/2019/may/oes372012.htm. Last accessed 16 November 2019.

United States Bureau of Labor Statistics, 2020. URL https://www.bls.gov/oes/current/oes372012.htm;https://www.bls.gov/oes/current/naics4_721100.html.

Julie Weed. Streamlining hotel housekeeping in a tight labor market, Nov 2019. URL https://www.nytimes.com/2019/11/26/business/hotel-housekeepers.html?smid=nytcore-ios-share.

Harrison White and Lee S Christie. Queuing with preemptive priorities or with breakdown. *Operations research*, 6(1):79–95, 1958.

Donald F Wood, Patrick J Moreo, and Gail Sammons. Hotel housekeeping operational audit: A questionnaire approach. *International journal of hospitality & tourism administration*, 6(3):1–10, 2005.

Yahoo. Why hotels are saying goodbye to check-in and check-out times, Apr 2015. URL https://www.yahoo.com/lifestyle/check-into-these-hotels-with-no-check-in-or-117470375077.html. Last accessed 16 November 2019.

Noa Zychlinski, Avishai Mandelbaum, Petar Momčilović, and Izack Cohen. Bed blocking in hospitals owing to scarce capacity in geriatric institutions—cost minimization via fluid models. *Manufacturing & Service Operations Management*, 2019.

Corinna C Zygourakis, John D Rolston, James Treadway, Susan Chang, and Michel Kliot. What do hotels and hospitals have in common? how we can learn from the hotel industry to take better care of patients. *Surgical neurology international*, 5(Suppl 2): S49, 2014.