Supply Chain and Service Operations with Demand-Side Flexibility

Yeqing Zhou

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
under the Executive Committee
of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2021

Abstract

Supply Chain and Service Operations with Demand-Side Flexibility

Yeqing Zhou

In this thesis, we consider improving supply chain and service systems through demand-side management. In Chapters 1 and 2, we focus on a new notion of flexibility that has emerged in e-commerce called consumer flexibility. Motivated by the fact that some customers may willingly provide flexibility on which product or service they receive in exchange for a reward, firms can design flexible options to leverage this consumer flexibility for significant benefit in their operations. In Chapter 1, we consider the context of online retailing where consumer flexibility can be realized through opaque selling, where some specific attributes of the products are not revealed to the customer until after purchase. In Chapter 2, we focus on the context of online booking systems for scheduled services where consumer flexibility can be realized through large time windows. The main findings are on the power of limited flexibility using simple flexible options with just a small fraction of customers willing to be flexible.

In Chapter 3, we study the issue of congested elevator queuing systems due to the requirement of social distancing during a pandemic. We propose simple interventions for safely managing the elevator queues, which require no programming of the elevator system and only

manage passenger behaviors. The key idea is to explicitly or implicitly group passengers going to the same or nearby floor into the same elevator as much as possible. Simulations and stability analysis show that our proposed interventions significantly reduce queue length and wait time.

Table of Contents

List of Figures

List of Tables

Acknowledgements

First and foremost, I must acknowledge my advisor, Adam Elmachtoub. Adam is an amazing advisor who led me into the Ph.D. program, guided me through the journey and always encourage me to go further. Thank you, Adam, for introducing me to the world of research, spending countless hours with me working on the papers and presentations, and helping me so much in this special job market year. Your passion, enthusiasm, and optimism truly inspire me and I will take them along with me in my academic journey.

I would also like to thank the other members of my thesis committee – David Yao, Cliff Stein, Jing Dong, and Yehua Wei – for spending time serving in my committee. David, Cliff, and Yehua have been great coauthors and have given me invaluable advice and help throughout my years as a Ph.D. student. Special thank to David, Yehua, and Will Ma, for writing me recommendation letters for job applications. I truly appreciate your time and support in the past job market season. I also want to thank Omar Besbes, Carri Chan, Yuri Faenza, and Vineet Goyal for their advices and guidance in my job market year.

My grateful thanks also go to Donald Goldfarb, Daniel Bienstock, Jose Blanchet, and Awi Federgruen for their great courses in various fundamental domains of operations research. I would also like to thank the staff team in the IEOR department for planning so many fun events and making the Ph.D. journey so wonderful. A special thanks to Lizbeth Morales

who has been so helpful throughout the process.

I am grateful to have met so many wonderful people at Columbia. My labmates Micheal Hamilton, Ryan McNellis, Xiao Lei, Harsh Sheth, and Yunfan Zhao are such nice guys and always available to be a helping hand. Xiao has also been a great collaborator. It was a great a pleasure to work with you and to turn our class project into a paper together. I want to thank Sai Mali Ananth for being a such a wonderful and proactive collaborator. I never imagined to start a project and make progress in such an efficient way without meeting in person! A special mention to Zhili Lin who contributed to the beauty and joy of my life at Columbia. It is indeed my pleasure to have you as my first and best friend in New York City.

I especially thank my partner Yunjie Sun, for his support in my entire Ph.D. journey. You have brought so much fun into my life. Especially in the past year which was so intense and stressful, your encouragement has been the most important piece for me to carry on.

Finally, I would like to thank my parents Yanping Bu and Wei Zhou, who inspired me to pursue the doctoral degree and an academic career. My father has taught me to appreciate life and explore the interesting parts in almost everything. My mother has been a role model and taught me to be calm and confident when facing difficulties. The love, trust, and confidence you provided have shaped me into the person I am today. I love you all!

To my family.

Introduction

In this thesis, we study the value of demand-side flexibility in supply chain management and service operations. Specifically, we focus on how to reduce costs or improve efficiency by simply managing or influencing user behavior.

In the highly competitive online marketplace, retailers and service providers are tempted to increase the number of options offered to maximize the chance of a purchase. However, there is a fundamental tradeoff between increasing the number of options and the difficulty in managing the inventory/capacity. To address this fundamental tradeoff, we study an innovative concept – consumer flexibility – defined as a consumer's explicit willingness to provide flexibility on which product or service option they may receive in exchange for a reward. Online retailers can leverage consumer flexibility through opaque products. An opaque product is a product where some specific attributes are not revealed to the customer until after purchase. The retailer can utilize consumer flexibility to balance inventory levels and reduce supply chain costs. In online service platforms (such as healthcare, home maintenance, and grocery delivery), a customer makes an appointment by choosing one option from a list of regular time windows. Consumer flexibility can be realized through large time windows, which are composed of multiple regular time windows. Customers that offer time flexibility allow the service providers to better utilize their service capacities.

In the first part of the thesis, we study a multi-product joint replenishment problem with opaque products. In this setting, consumer flexibility is leveraged through a k-opaque option where customers select k products from which the seller allocates one to the customer. We tap into a novel connection between opaque selling and the balls-into-bins framework. We show that selling opaque products can be done using a simple inventory-balancing policy and can yield substantial cost savings for online retailers. We find that even with limited flexibility, i.e., offering 2-opaque products and with only a small fraction of customers being flexible, the seller can achieve significant inventory cost savings, which is on the same order of magnitude as the cost savings from a fully flexible scheme. Moreover, our study provides practical guidance to retailers that the first-order effect comes from adopting opaque products with a small incentive, while re-optimizing the inventory policy only provides a second-order value. This work is detailed in Chapter 1 and Elmachtoub et al. (2019).

In the context of online service platforms, we also demonstrate the power of limited flexibility with a simple large time window design and only a small fraction of customers choosing the large time windows. In Chapter 2, we analyze the capacity pooling benefit using large time windows. We demonstrate that there are diminishing returns to the increased fraction of customers being flexible, and the benefit of a limited flexible design where large time windows are composed of two consecutive regular time windows can capture most of the total capacity pooling benefit. In particular, we investigate the large time window design in a long chain structure. In the process flexibility literature, it is well-known that incrementally adding flexibility has increasing returns (supermodularity), and as a byproduct 'closing the loop' provides the most benefit. We found that supermodularity does not hold in our model, and that 'closing the loop' typically provides the least value in time window design. The

lack of supermodularity in our setting is a rather fortunate outcome, as a high capacity utilization can be achieved without offering too many choices.

In the third part of the thesis, we turn our focus to an operational challenge that arises in a pandemic. Elevator capacity in high rise buildings during a contagious pandemic is reduced dramatically to allow for social distancing. Such a reduction, combined with the commonly used first-come first-serve queuing policy, can cause large queues to build up in lobbies. To address this issue, we propose simple interventions for safely managing the elevator queues, which require no programming of the elevators and only manage passenger behavior. The key idea is to explicitly or implicitly group passengers going to the same floor into the same elevator as much as possible. We use mathematical modeling, epidemiological principles, and simulation to design and evaluate our interventions. We prove theoretical results on stability conditions and provide simulations using data from a real-world building that explain how our proposed interventions significantly reduce queue length and wait time. Our proposed interventions can easily be implemented in any building, even historical buildings with outdated technology. This work is detailed in Chapter 3 and Ananthanarayanan et al. (2020).

Chapter 1

The Value Of Flexibility From Opaque Selling

An opaque product is a product where some secondary attribute is not revealed to the customer until after purchase. Selling opaque products has become popular on e-commerce platforms where the hidden attribute is often color or style, thanks to its obvious advantage in risk pooling the demand. The objective of this study is to quantify the value of consumer flexibility that underlies opaque selling. We consider a setting in which an online retailer sells N products that only differ in a certain secondary attribute, and in addition offers a k-opaque option where customers select k products from which the seller allocates one to the customer. We assume that the prices are exogenously set such that q fraction of the customers select the k-opaque option. We refer to the tuple (q, k) as the degree of opacity, where the flexibility of the system is increasing in both q and k. When $q = 0$ or $k = 1$, our setting reduces to traditional non-opaque selling with no flexibility, and when the degree of opacity is $(1, N)$, this corresponds to a fully flexible scenario where every customer is willing to receive any product.

We find that even with a minimal degree of opacity, with q very small and $k = 2$, the seller can achieve significant cost savings which we quantify precisely. Remarkably, we find that the cost savings from this minimal degree of opacity is on the same order as the fully flexible case corresponding to $(1, N)$. This finding has practical managerial implications, as achieving

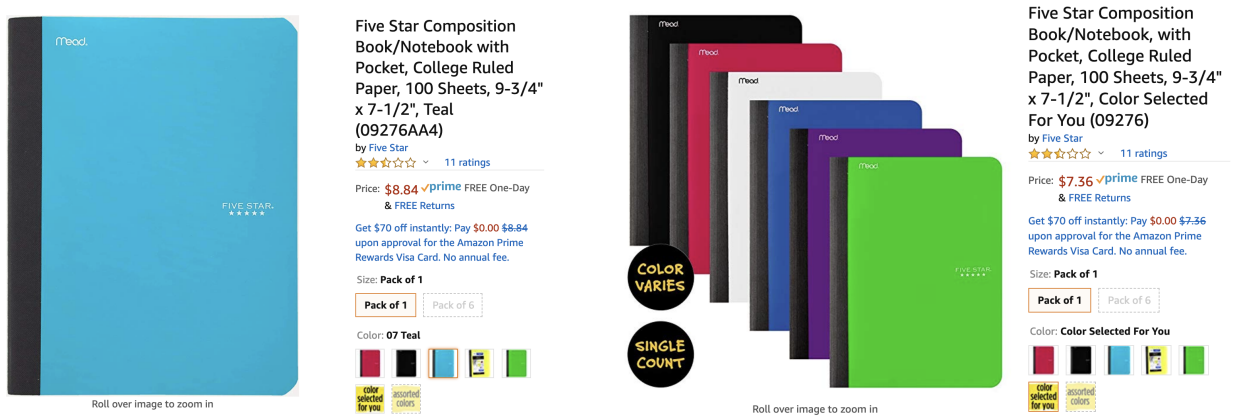
$(1, N)$ is practically infeasible since it would require substantial incentives by the seller to materialize. Our proofs rely on analyzing a simple balancing policy to allocate products to opaque customers, and explores a novel connection to the balls-into-bins framework. As a byproduct of our analysis, we show that the simple balancing policy is asymptotically optimal. Finally, we provide numerical experiments that suggest our main insight, namely that a small degree of opacity provides advantages akin to a fully flexible system, holds under various extensions of our core setting.

1.1 Introduction

Consider a common setting of an online retailer selling several products that are similar in function and quality but may differ in a secondary attribute such as color or style. Under this setting, inevitably some customers are indifferent among their top choices and may be willing to select an opaque option if offered. Namely, an opaque product is one in which the customer voluntarily provides the seller the flexibility to choose which of the regular products they will receive. We refer to the practice of selling opaque product options alongside the regular products as *opaque selling*, and we refer to customers who purchase the opaque product as *opaque customers*. For example, Amazon.com usually offers opaque products in the form of a "Colors May Vary" option (along with a small reward) in merchandise categories such as stationery, baby products, toys, and apparel. At the same time, customers are also allowed to pick a specific color if they want, as seen in Fig. 1.1. In these cases, the number of colors is typically around 3-10 (e.g., notebooks with six different colors).

It is perhaps no surprise that opaque selling allows sellers to tap into *consumer flexibility*,

Figure 1.1: An Example of N-Opaque Product



Note. This is an example of an N-opaque product on Amazon.com. The N-opaque product is represented by a "Color Selected For Your" option on the right, with $1.48 discount. Customers can still choose a specific color if they want (teal in this example) as shown on the left.

which is the explicit willingness of a consumer to receive one of several options. One clear advantage of opaque selling lies in the effect of risk pooling: the demand volatility (variance) over individual products can be substantially reduced when they are (partially) aggregated into demand for an opaque product. In turn, the retailer can strategically allocate opaque products to reduce inventory costs. Yet, how to effectively capture this risk pooling effect via a properly implemented opaque selling scheme is rather nontrivial. The central aim of our work is to quantify the value of consumer flexibility from opaque selling, and describe how much flexibility, or "opacity", is needed to capture this value.

One important dimension of an opaque selling scheme is the number of potential products a customer may receive when choosing the opaque product. Shall one require opaque customers to be willing to accept any one of the N colors as in the Amazon example in Fig. 1.1? Or, can we allow opaque customers to specify a small number, k, of colors of their choice from which the seller will choose one? Fig. 1.2 illustrates a concrete example of k-opaque products, where $k = 1$ corresponds to a non-opaque customer who only wants a

specific color, and $k = 2$ corresponds to an opaque customer who is willing to take either one of two colors (both specified by the customer) but not the other colors offered. By extension, $k = N$ then corresponds to a fully opaque customer who will accept all N colors.

Figure 1.2: An Illustration of a 2-Opaque Product

Option: **Choose 1 Color**

Choose 1 Color | Choose 2 Colors

Color: **Blue**

Blue | Red | Green | White | Black

Price: $9.99

Option: **Choose 2 Colors**

Choose 1 Color | Choose 2 Colors

Color: **Blue or Green** (final color selected for you)

Blue | Red | Green | White | Black

Price: $8.99

Note: This is an illustration of how retailers can sell 2-opaque products in practice. The "Choose 1 Color" option represents $k = 1$ where a customer chooses a specific color (blue). The "Choose 2 Colors" option represents $k = 2$ where the customer chooses a 2-opaque product and specifies 2 colors (blue and green in this example).

Another essential dimension of an opaque selling scheme is the fraction of customers that select the opaque product, which we denote by $q \in [0, 1]$. Obviously, under the same incentive structure, a larger k value will reduce q. Alternatively, the larger the k value, the higher the incentive (e.g., discounts, gift coupons, reward points, etc) the retailer will need to provide in order to attract the same fraction q of opaque customers.

Thus, central to opaque selling is the *degree of opacity* as measured by the tuple (q, k). When $q = 1$ and $k = N$, we have complete opacity: all customers are opaque and every one is fully opaque, achieving the maximal level of flexibility. The other extreme is $q = 0$ or

$k = 1$: every customer is non-opaque, resulting in zero flexibility. The latter case reduces to traditional non-opaque selling, whereas the former case may require prohibitive investment in terms of price discounts or other incentives in order to materialize. In order to capture the value of flexibility, we consider the inventory cost savings achieved by various degrees of opacity. Our results rely on analyzing a simple balancing policy to allocate products to opaque customers, and explores a novel connection to the balls-into-bins framework. The primary contributions in this paper are the following novel insights. We show that for a minimal degree of opacity $(q, k) = (\delta, 2)$ with δ close to 0 that: *(i)* the cost savings compared to the non-opaque traditional setting is substantial and *(ii)* the cost savings is on the same order as the fully flexibly case where $(q, k) = (1, N)$.

Before describing our contributions and analysis in detail, several remarks are in order. First, the value, or benefit, of opaque selling we focus on here is savings in inventory cost (for ordering replenishments and holding the goods), as achieved by the degree of opacity (q, k). We do not explicitly take into account the additional cost to incentivize the opaque customers, assuming it is already implied by the given (q, k). (Clearly, this additional cost must be an increasing function of (q, k), which is context specific and can be characterized via market analysis or machine-learned from data.) In any event, our main conclusion, as alluded to above, indicates that all one needs is a minimal degree of opacity—and hence, a minimal level of incentive cost—to reap most of the benefit. In fact, our computational experiments suggest that the most profitable strategy corresponds to a minimal degree of opacity. Second, as our objective is to quantify the value of consumer flexibility embodied in opaque selling, we isolate the degree of opacity from other considerations such as more complex inventory cost structures and replenishment policies. To this end, as well as to

facilitate exposition, we utilize the most basic setting with no lead times and use a simple (joint) order-up-to policy to replenish inventory. In the numerical section, we do however provide detailed examples that include settings with lead times, backlogging, lost sales, and policies with improved order-up-to levels. Yet the results all suggest that this added sophistication does not fundamentally change the value of flexibility brought forth by opaque selling.

1.1.1 Summary of Main Contributions

The main contributions of this paper are as follows.

First, our model reveals fundamental insights regarding consumer flexibility in the context of opaque selling. Specifically, we quantify the cost savings benefit as a function of the number of products N and the order-up-to level S. We show that for a minimal degree of opacity with q roughly on the order of $\sqrt{\frac{\log N}{S}}$ and $k = 2$, the cost benefit is at least on the order of $\sqrt{\frac{\log N}{S}}$. Note that $\sqrt{\frac{\log N}{S}}$ can imply significant cost savings as it scales gracefully in S, while simultaneously not being too prohibitive to achieve the desired level of q. Second, we show that the cost savings under this minimal degree of opacity is on the same order as the cost savings from the fully flexible case where $q = 1$ and $k = N$. Our results suggest that our notion of consumer flexibility share similar insights as the general literature on flexibility, namely that "a little flexibility goes a long way".

In terms of methodology, our analysis explores a novel connection between opaque selling and the balls-into-bins model. In the latter model, a given number of balls are sequentially and randomly thrown into a set of bins and one is interested in the maximum load across

all the bins. We show that under the right coupling, one can relate the CDFs of the number of customers served between consecutive replenishments in our system and the maximum loaded bin. Using this framework, we utilize results from the balls-into-bins literature to prove fundamental properties of a simple and natural balancing policy for allocating opaque products. These properties help us to precisely characterize the cost savings under opaque selling. We also are able to prove that the balancing policy is surprisingly not optimal, but it is indeed a near-optimal policy with an error that decays rapidly in the order-up-to level.

Finally, we conduct extensive numerical experiments which yield several further insights. When the retailer considers the overall profit, we demonstrate that 2-opaque selling, a limited flexible scheme, is even more effective than the fully flexible design ($k = N$). Under various parameter settings and three commonly-used choice models, we find out that 2-opaque products can attract more customers to choose the opaque option (higher q) with a significantly lower discount level, which leads to higher savings in inventory costs and a higher increase in profit. We also show that our theoretical insights hold computationally under settings with lead time, backlogging, and lost sales. Moreover, we find that there is a marginal benefit to re-optimize the order-up-to level when introducing opaque products, allowing the seller to reap most of the benefits with minimal changes to their inventory policy.

1.1.2 Literature Review

The opaque selling strategy and similar tactics have been extensively studied in the literature of revenue management (Gallego and Phillips (2004), Jiang (2007), Fay and Xie (2008), Jerath et al. (2009), Xiao and Chen (2014), Elmachtoub and Hamilton (2021)), where the

focus is on the ability to do price discrimination, to create new market segmentation, or to better allocate limited resources to customers who are willing to pay more. For all of these studies, the objective is to maximize revenue from a finite amount of inventory in a finite horizon, and all costs are essentially sunk costs which is characteristic of the travel industry. In contrast, we consider an online retailer who sells year-round goods where profit margins are typically razor thin and reducing costs can result in significant profit gains. Moreover, the incentive structure for opaque products is inherently different: a customer may choose the "color may vary" option for a stapler in exchange for a coupon or reward points that they may never use, simply because the customer is indifferent to the color choice. Such behavior is highly unlikely when purchasing airline tickets or hotel rooms where only large price markdowns would attract consumers to choose the opaque options. Therefore, in this paper we consider opaque selling as an innovation to leverage consumer flexibility to provide a risk pooling benefit which we quantify through inventory cost savings. The only previously known paper that considers inventory costs in the dynamic management of opaque products is Elmachtoub et al. (2015), who focus on the $N = 2$ case where the balancing policy is shown to be optimal and a Markov chain analysis on the difference of inventory levels between the 2 products yields the cost savings benefit. Our paper considers more practical scenarios where $N \geq 3$ and k-opaque products are offered, in which case the balancing policy is surprisingly no longer optimal, but still near-optimal. We also note that our analysis requires completely different techniques than that of Elmachtoub et al. (2015).

Consumer flexibility, referring to a consumer's willingness to receive one of multiple options that are offered to them instead of a specific option, has been adopted by many online platforms but lacks attention in the literature. One paper on the subject is Ströhle

et al. (2018), who study the benefits of spatial and temporal customer flexibility in car-sharing services using a real-world data set, where in this application, customers give up the exact knowledge of the pick-up and drop-off time and location. In a recent paper, Tao et al. (2020) consider a multi-server queueing system with flexible customers who would join the shortest line among d randomly generated lines. Their results suggest that having just a small fraction of flexible customers can benefit the system tremendously in terms of the average waiting time. The general concept of flexibility has been studied extensively in the literature of manufacturing, service, and supply chain management (see Wang et al. (2019) for a recent survey), where flexibility comes from the supply side. In particular, process flexibility refers to a firm's ability to deal with demand uncertainty by having production processes that can produce multiple types of products. Jordan and Graves (1995) demonstrated that a partial flexibility structure, the long chain design, can accrue most of the benefits achieved by a fully flexible scheme. The effectiveness of the long chain and other designs with limited flexibility has been investigated theoretically in many recent works (e.g. Simchi-Levi and Wei (2012), Wang and Zhang (2015), Désir et al. (2016)). Other studies consider the benefit of flexibility in online allocation problems (Asadpour et al. (2019)) and queueing systems (Tsitsiklis and Xu (2012), Tsitsiklis and Xu (2017)). Consumer flexibility is intrinsically different from the previous notions of flexibility mainly because it comes from the demand side and some key characteristics can not be captured by the models in the literature (see Appendix 1.8 for a detailed comparison). However, we do find that the notion of 'a little flexibility goes a long way" still applies in the context of consumer flexibility.

The inventory model we study is borrowed from the literature on multi-product inventory problems with joint replenishment costs. These problems with joint replenishment costs

are notoriously difficult, and may have very complex optimal policies even in simple two-item settings (see Ignall (1969)). Silver (1965) considered a two-item inventory system and proposed a simple joint ordering rule that orders both items up to an order-up-to level whenever the inventory of either of them drops to zero. This is a special case of the can-order policy, which was introduced by Balintfy (1964) and considered in Silver (1974), Atkins and Iyogun (1988), Melchiors (2002). Because of the inherent difficulties within the multi-product inventory problem, we shall restrict ourselves to the use of a simple can-order policy, and focus on how to allocate demand for opaque products as well as the cost benefit achieved by various degrees of opacity.

Part of our solution approaches in this paper are inspired by previous studies on the balls-into-bins model, which is a classical model that demonstrates the power of limited flexibility in load balancing when the requests arrive sequentially and the fulfillment decisions are made in real time. For classical results and variants of the balls-into-bins model, see Raab and Steger (1998), Peres et al. (2010), and the survey by Richa et al. (2001). Recently, Asadpour et al. (2019) use balls-into-bins to study an online resource allocation problem.

The paper is organized as follows. In Section 1.2, we present the model and the opaque product allocation policy we shall consider. In Section 1.3, we show the a minimal degree of opacity provides enough consumer flexibility to generate significant cost savings, which are on the same order as a fully flexible opaque selling scheme. We prove our results by developing a fundamental connection with the balls-into-bins problem. In Section 1.4, we provide structural results of the optimal opaque product allocation policy and prove asymptotic optimality of the balancing policy. In Section 1.5, we conduct numerical experiments to

validate our findings under several extensions of our model. Finally, we conclude in Section 1.6.

1.2 Model

In this section, we formally introduce the opaque selling strategy, define the degree of opacity, describe the inventory dynamics, characterize costs, and discuss a natural balancing policy for opaque product allocation.

In our model, the online retailer offers N horizontally differentiated products, i.e., each product is the same except for its color or style. In addition to the N products, the retailer offers a k-opaque product option to the customer for some fixed k. A customer who purchases a *k-opaque product* will first choose k out of the N products they are willing to receive, and then will be allocated any of the k products at the sole discretion of the retailer (see Fig. 1.2). Note that to implement a k-opaque product, the retailer does not display all possible subsets of size k. Rather, the customer simply indicates that they would like to purchase a k-opaque product and then selects their top k products. We remark that in our model, a 1-opaque product is equivalent to no opaque selling since customers must explicitly choose their favorite product, which leads to no flexibility in the allocation decision of the retailer. For the N-opaque product option, the customers are not able to indicate any preferences and can be allocated any of the N products. However, the N-opaque product clearly provides the most flexibility to the retailer, if selected.

Purchasing customers arrive according to a stochastic process with i.i.d. interarrival times with mean $\frac{1}{\lambda}$. Each customer will consume one unit of a product. With probability

q, the customer purchases a k-opaque product, where all subsets are equally likely to be chosen. With probability $1-q$, the customer purchases one of the N products, each with equal probability. We refer to the special case of $q=0$ as the traditional model, where no opaque selling is conducted. The case where $q=1$ is a best-case scenario where all customers are flexible.

The focus of our study is to understand how the degree of opacity (q,k) contributes to the cost savings induced by opaque selling. Specifically, we shall quantify the cost savings in the *limited flexible* case where $q>0$ is very small and $k=2$, and also show it is on the same order as the *fully flexible* case where $q=1$ and $k=N$. In both cases, we benchmark against the *traditional strategy* where there is no opaque selling. We remark that under all comparisons we assume that the overall demand is unchanged, meaning that the savings is what would happen if one can convert q fraction of the customers to purchase k-opaque products. We remark that we do not explicitly describe how to attain q as this is highly context-dependent (incentives can be reward points, discounts, coupons, etc.). Rather, our purpose is to prove that a minimal degree of opacity is sufficient. In Section 1.5.2, we do provide numerical experiments on a scenario where the overall demand and q explicitly depend on k and the opaque price discount.

Next, we describe the dynamics and the corresponding costs of our model. We assume the retailer orders N products from the same manufacturer, and pays a joint ordering cost K for each replenishment. There is a holding cost h for holding one unit of a product for one time unit. Due to the notorious complexity of multi-item inventory management, we shall limit our theoretical analysis to the case where there is no lead time, backlogging, or lost sales. In this case, the retailer would follow an $(0,S)$ policy, where an order is placed

exactly when the inventory level of a product drops to 0. When an order is triggered, all N products are replenished to the order-up-to level S. We call the time between two consecutive replenishments (orders) a replenishment cycle. Since the demand is symmetric, the order-up-to level S is the same across all N products. Note that in our model S is exogenously given and assumed to be the same under all strategies considered, which implies our results hold even when the retailer does not re-optimize their order-up-to level.

The only decision the retailer has to make is which of the products to allocate when a customer purchases an opaque product option. Thus, a feasible policy π for the retailer maps each possible inventory state and the subset of products chosen by an opaque customer to a single product in the subset that will be allocated to that opaque customer. An optimal opaque allocation policy π^* minimizes the combined long-run average costs of the system.

Next, we introduce the key metric in order to compute the costs. We define the random variable $R^\pi(q,k)$ as the number of units consumed in a replenishment cycle, i.e., the number of customers seen from the inventory state $\vec{S} := (S,\ldots,S)$ until the inventory of one product drops to the reorder point $s=0$, when k-opaque products are provided and q fraction of the customers choose the opaque option under the opaque product allocation policy π. Using the first two moments of this key random variable, we can express the long run average ordering cost and the long run average holding cost in closed form. For any policy π, the long run ordering cost per unit sold, $\mathcal{K}^\pi(q,k)$, can be expressed as

$$\mathcal{K}^\pi(q,k) := \frac{K}{\mathbb{E}[R^\pi(q,k)]}, \tag{1.1}$$

and the long run holding cost per unit sold, $\mathcal{H}^\pi(q,k)$, can be expressed as

$$\mathcal{H}^\pi(q,k) := \frac{(2NS+1)\mathbb{E}[R^\pi(q,k)] - \mathbb{E}[R^\pi(q,k)^2]}{2\lambda\mathbb{E}[R^\pi(q,k)]}h \tag{1.2}$$

$$\leq \frac{2NS+1-\mathbb{E}[R^\pi(q,k)]}{2\lambda}h. \tag{1.3}$$

The derivation of these expressions follows directly from Lemma 1 of Elmachtoub et al. (2015), although their cost savings analysis does not extend to the case where $N \geq 3$ and k-opaque products are offered. The upper bound on the long run holding cost in Eq. (1.3) simply follows from Jensen's inequality. For the special case of $q = 0$, there is no opaque product allocation policy necessary and we denote $R(0) := R^\pi(0,k)$, $\mathcal{K}(0) := \mathcal{K}^\pi(0,k)$, and $\mathcal{H}(0) := \mathcal{H}^\pi(0,k)$.

1.2.1 Balancing Policy

In this subsection, we present a natural *balancing* policy for allocating products to opaque customers when demands are symmetric across different types of products, which is the focus of this paper. Simply put, the balancing policy fulfills demand for opaque products by using the product with the highest on-hand inventory level (with ties broken arbitrarily). The motivation comes from noticing that the ordering cost expression in (1.1) only depends on the chosen policy via the the quantity $\mathbb{E}[R^\pi(q,k)]$. As $\mathbb{E}[R^\pi(q,k)]$ increases, the ordering cost rate decreases which is natural since orders occur less frequently. For the holding cost in Eq. (1.2), we cannot make a similar statement since it involves the second moment. Nevertheless, we observe that the upper bound of the long run holding cost in Eq. (1.3) decreases in $\mathbb{E}[R^\pi(q,k)]$. Therefore, a natural policy would be to maximize $\mathbb{E}[R^\pi(q,k)]$.

As one might expect, the policy that maximizes $\mathbb{E}[R^\pi(q,k)]$ is indeed the balancing policy, which we formalize in Proposition 1.2.1 below and prove in Appendix 1.7. For convenience, we define the balancing action as choosing the maximum inventory product for an opaque customer, and we let $\mathcal{M}$ be the policy that always takes the balancing action for opaque customers.

Proposition 1.2.1. *The balancing policy $\mathcal{M}$ maximizes $\mathbb{E}[R^\pi(q,k)]$ among all possible opaque fulfillment policies.*

Surprisingly, in Section 1.4 we show that the balancing policy is not necessarily optimal, although it is near-optimal.

1.3 Quantifying the Value of Flexibility

In this section, we seek to understand the value of selling opaque products in terms of the total cost savings stemming from the risk pooling effect. Specifically, we quantify the benefit of a scheme with degree of opacity (q,k), under the balancing policy $\mathcal{M}$, to a scheme with no opaque selling ($q=0$), with all other parameters remaining equal. Our goal is to characterize the cost savings in terms of four key quantities: N, S, k, and q. More specifically, we define the *relative cost savings* of having q fraction of customers purchasing k-opaque products by

$$\frac{\mathcal{K}(0)+\mathcal{H}(0)-\mathcal{K}^{\mathcal{M}}(q,k)-\mathcal{H}^{\mathcal{M}}(q,k)}{\mathcal{K}(0)+\mathcal{H}(0)}. \tag{1.4}$$

The relative cost savings is simply the decrease in cost from having a degree of opacity (q,k) normalized by the cost having no flexibility ($q=0$). By bounding this quantity, we provide

a guarantee on the cost saving benefit from opaque selling which can help retailers decide what degree of consumer flexibility should be targeted. We shall make the assumption in this section that $S \gg \log N$, which is clearly satisfied in most practical settings (e.g. $S = 100, N = 5$). We first summarize and discuss our main results in Theorems 1.3.1, 1.3.2 and 1.3.3 and then provide the key analysis later in Section 1.3.1. The proofs are provided in Appendix 1.7.

Theorem 1.3.1 (**The Value of Flexibility**). *The relative cost savings induced by opaque selling is* $\Omega\left(\sqrt{\frac{\log N}{S}}\right)$ *under the balancing policy, for any* $k \geq 2$ *and any* $q = \Omega\left(\sqrt{\frac{\log N}{S^{1-\epsilon}}}\right)$ *where* $\epsilon > 0$ *is fixed.*

The relative cost savings are at least on the order of $\sqrt{\frac{\log N}{S}}$, which decays rather slowly as S (order-up-to level) gets large, and increases as N (the number of products) increases. This ensures that there can be significant savings even with large S, and we note that the dependency on $\frac{1}{\sqrt{S}}$ is tight from Elmachtoub et al. (2015) when $N = 2$. We also note that as the bound decays towards 0, the minimum requirement for q also decays gracefully at a similar rate. This implies that as the S grows, the benefit indeed decreases although the required amount of opaque customers also decreases. It is also the case that $\sqrt{\frac{\log N}{S}}$ can represent significant cost savings, while at the same time be a very achievable target for q. Finally, we note that this cost saving guarantee holds for all $k \geq 2$ and all q exceeding the threshold provided. In Theorem 1.3.2 we show that the cost savings are approximately evenly split between the holding and ordering costs, while in Theorem 1.3.3 we show that the cost savings of a limited degree of opacity is on the same order as a fully flexible system.

Theorem 1.3.2 (**Ordering and Holding Cost Savings Are Same Order**). *The relative cost savings of both the ordering costs and holding costs are on the same order under the balancing policy, for any given* $k \geq 2$ *and* $q = \Omega\left(\sqrt{\frac{\log N}{S^{1-\epsilon}}}\right)$ *where* $\epsilon > 0$ *is fixed.*

Theorem 1.3.2 says that the relative cost savings of both the order and holding costs are approximately equal under general conditions on the degree of opacity. This theorem is rather striking as it holds for any cost parameters K and h, and independent of how S may have been chosen. Thus, whether the main drivers of costs are due to holding inventory (large capital investment or lots of storage space), due to ordering costs (frequent and expensive replenishments), or both, we are guaranteed to drive both costs down (by at least $\sqrt{\frac{\log N}{S}}$ from Theorem 1.3.1). Now suppose the order-up-to level S was chosen such that the holding and ordering costs are approximately equal, which is natural in inventory settings when trying to minimize cost (using intuition from the standard EOQ model). Then Theorem 1.3.2 implies that introducing a certain degree of opacity reduces both costs down by an approximately equal amount. Thus, the holding and ordering costs will still be approximately the same using the same order-up-to level S, implying that there is little benefit to modifying S after introducing the opaque selling scheme. This intuition is verified via computational experiments in Section 1.5.3.

Theorem 1.3.3 (**Limited and Full Flexibility Are Same Order**). *Consider a limited flexible scheme with* $k = 2$ *and any* $q = \Omega\left(\sqrt{\frac{\log N}{S^{1-\epsilon}}}\right)$ *for some* $\epsilon > 0$ *under the balancing policy. The relative cost savings is the same order of magnitude as the cost savings of a fully flexible scheme (*$k = N$ *and* $q = 1$*) under the optimal allocation policy.*

Theorem 1.3.3 validates the well-known principle that *a little bit of flexibility provides*

almost all the value as full flexibility. In particular, the limited 2-opaque product with a small fraction of opaque customers is essentially as powerful as the fully flexible scheme with N-opaque products and all customers being opaque. In fact, this equivalence is achieved even when the limited flexibility scheme relies on the balancing heuristic while the fully flexible scheme uses the optimal policy (which coincidentally turns out to be the balancing policy). We can thus conclude that limited flexibility in both dimensions can achieve the same order of cost savings as the maximum possible savings in the fully flexible scheme with the optimal allocation policy. Note however, that our model and analysis is fundamentally different than other results on limited flexibility in the literature, see Appendix 1.8 for details. To complement our theoretical findings, we provide numerical experiments in Section 1.5.1 that show the limited effect of increasing the degree of opacity towards full flexibility. In the next subsection, we dive into the key steps to prove Theorems 1.3.1, 1.3.2 and 1.3.3.

1.3.1 Balls-into-Bins Connection

At the heart of our analysis lies a fundamental connection of our inventory model with opaque selling to the balls-into-bins model. The general framework for the balls-into-bins model is that there are m balls that are thrown according to a well-defined process into N bins. Specifically, the process is characterized by a distribution vector $\mathbf{p} = (p_1, \ldots, p_N)$, where p_i is the probability a ball is placed in the i-th most loaded bin. Of primary interest in the literature is the load of the maximally loaded bin after all m balls have been thrown, which we denote by the random variable $M(m)$. We shall draw on key results on $\mathbb{E}[M(m)]$ for particular choices of the distribution vector.

To make the connection to our model, we shall make the analogy of customers to balls and products to bins. We shall think of the inventory system beginning with full inventory levels at S for each product, and the balls-into-bins system beginning with each bin empty. An opaque product allocation policy that seeks to balance inventory, $\mathcal{M}$, equates to a ball-throwing policy that seeks to minimize the load imbalance across the bins. We shall select $\mathbf{p}$ so that we can exactly couple the product each customer receives using our balancing policy to the bin that each ball lands in. Specifically, under the balancing policy with N-opaque selling, the product with highest on-hand inventory is sold with probability $\frac{1-q}{N} + q$, and any other product is sold with probability $\frac{1-q}{N}$. This corresponds to a distribution vector of $\mathbf{p} = (\frac{1-q}{N}, \ldots, \frac{1-q}{N}, \frac{1-q}{N} + q)$, where the least loaded bin has an extra probability of q to be chosen due to opaque customers. At the other extreme where $k = 1$ or $q = 0$, the distribution vector is simply $\mathbf{p} = (\frac{1}{N}, \ldots, \frac{1}{N})$. For k-opaque products, the probability that the i-th most available product is given to a customer under the balancing policy is $\frac{1-q}{N} + q\frac{\binom{N-i}{k-1}}{\binom{N}{k}}$ when $i \geq k$ and $\frac{1-q}{N}$ when $i \leq k-1$. To couple this to the balls-into-bins process, the loading vector would correspond to exactly these probabilities.

Using our analogy, our goal is to be able to connect $\mathbb{E}[M(m)]$ to $\mathbb{E}[R^{\mathcal{M}}(q,k)]$, which would help us to characterize the performance of the balancing policy. Unfortunately, there is one major complication that prevents making this connection seamless. In the inventory management setting, we do not know the number of customers there will be in advance, but know the exact load of the maximally chosen product (S). However, in the balls-into-bins model we know exactly the number of balls thrown (m), but do not know the load of the maximally chosen bin exactly. Fortunately, we can relate the CDFs of $M(m)$ and $R^{\mathcal{M}}(q,k)$ at key quantities according to the following lemma.

Lemma 1.3.1. *Suppose $M(m)$ is governed by a distribution vector* $\mathbf{p}$ *and it is coupled with the choices of the balancing policy $\mathcal{M}$. Then, $\mathbb{P}[R^{\mathcal{M}}(q,k) \leq m] = \mathbb{P}[M(m) \geq S]$.*

Proof. As described previously, we shall couple the choice of the systems, beginning with full inventory and empty loads, so that when a customer chooses the product with the i^{th} smallest inventory the corresponding ball lands in the bin with the i^{th} highest load. If the maximum loaded bin has at least S balls in it by the m^{th} throw, then the corresponding product has had S customers and a replenishment was triggered. Thus, $\{M(m) \geq S\}$ implies that $\{R^{\mathcal{M}}(q,k) \leq m\}$. If a replenishment has occurred by the m^{th} customer, then the depleted product corresponds to a bin with at least S balls in it. Thus, $\{R^{\mathcal{M}}(q,k) \leq m\}$ implies that $\{M(m) \geq S\}$ which proves the result. □

Generally speaking, using Lemma 1.3.1 we can bound $\mathbb{E}[R^{\mathcal{M}}(q,k)]$ for different values of $q \in [0,1]$. However, there is a fundamental phase transition at $q = 0$ which requires separating the analysis in the case $q = 0$ and $0 < q \leq 1$. In the case of $q = 0$, the corresponding distribution vector for the balls-into-bins process is the uniform allocation vector, which is studied in Raab and Steger (1998) and whose results we leverage to prove the following bound on $\mathbb{E}[R(0)]$.

Lemma 1.3.2. $\mathbb{E}[R(0)] = NS - \theta_S$, *where* $\theta_S = \Omega\left(N\sqrt{S \log N}\right)$.

The proof of Lemma 1.3.2 is provided in Appendix 1.7.

Next, we consider the case where $0 < q \leq 1$, and aim to derive a lower bound for $\mathbb{E}\left[R^{\mathcal{M}}(q,k)\right]$. We shall leverage a balls-into-bins result of Peres et al. (2010), which provides upper bounds on the expected gap between the maximum load, $M(m)$, and the average

load for a class of well-behaved distribution vectors. Specifically, if the distribution vector is designed so that more weight is given to bins with lower loads, then the expected gap between the maximum and average loaded bin can be bounded by a constant independent of the number of balls thrown. The exact condition for $\mathbf{p}$ is that $p_{\frac{N}{3}} \leq \frac{1-4\epsilon}{N}$ and $p_{\frac{2N}{3}+1} \geq \frac{1+4\epsilon}{N}$, where ϵ factors into the gap constant. Their result also holds for any vector $\mathbf{p}'$ that majorizes $\mathbf{p}$, which allows us to apply their result to the vector $(\frac{1-q}{N}, \ldots, \frac{1-q}{N}, \frac{1-q}{N} + q)$. Their result relies on analyzing a Schur-convex potential function that can directly bound the tail CDF of the gap, which allows us to bound the CDF of $R^{\mathcal{M}}(q, N)$ thanks to Lemma 1.3.1. The final lower bound on $\mathbb{E}[R^{\mathcal{M}}(q, N)]$ is in Lemma 1.3.3 below.

Lemma 1.3.3. *For any $\epsilon \in (0, 1)$ and for any $q = \Omega(\frac{1}{S^{1-\epsilon}})$, $\mathbb{E}\left[R^{\mathcal{M}}(q, N)\right] = NS - \theta_q$ where $\theta_q = O\left(\frac{N}{q} \log \frac{N}{q}\right)$.*

The proof of Lemma 1.3.3 is provided in Appendix 1.7. Note that θ_q, the expected number of units that are not sold in a replenishment cycle, is a constant independent of S. It implies that the number of customers served between replenishments is NS minus a constant depending on q, but independent of S. This is in contrast to Lemma 1.3.2 where the number of units not sold, θ_S, is at least on the order of $\sqrt{S}$. We bound q away from 0 to ensure that the result is meaningful and θ_q is less than NS. Since the value of S is usually large, the lower bound on q is typically small. Therefore, even if there is only a small proportion of opaque customers, $\mathbb{E}\left[R^{\mathcal{M}}(q, N)\right]$ can be much larger than $\mathbb{E}\left[R(0)\right]$, which suggests a significant cost savings potential of opaque selling.

Finally we consider the 2-opaque selling model. Lemma 1.3.1 also holds for 2-opaque selling when using the corresponding distribution vector and an identical coupling argument.

In the 2-opaque product model, with probability q, a customer will choose the opaque option and randomly select 2 out of N products, and with probability $1-q$, the customer will choose uniformly at random. The corresponding distribution vector connects the 2-opaque policy to the $(1+\beta)$-choice process, which is the balls-into-bins model studied in Peres et al. (2010). The $(1+\beta)$-choice process is defined as the following: balls are placed into N bins using a two-choice scheme with probability β and a random choice with probability $1-\beta$. In a two-choice scheme, two bins will be randomly generated and the ball will go to the bin with lower load. With $\beta = q$, we can directly apply the analysis in Peres et al. (2010) and use the connection stated in Lemma 1.3.1 to compute $\mathbb{E}[R^{\mathcal{M}}(q,2)]$.

Lemma 1.3.4. *For any $\epsilon \in (0,1)$ and for any $q = \Omega\left(\frac{1}{S^{1-\epsilon}}\right)$, $\mathbb{E}[R^{\mathcal{M}}(q,2)] = NS - \bar{\theta}_q$, where $\bar{\theta}_q = O\left(\frac{N}{q}\log\frac{N}{q}\right)$.*

The proof of Lemma 1.3.4 is provided in Appendix 1.7. Lemma 1.3.4 is derived using the same technique as Lemma 1.3.3. Lemma 1.3.4 directly uses the result in Peres et al. (2010) while the derivation of Lemma 1.3.3 further requires a majorization result. Note that given N and q, the difference between $\mathbb{E}[R^{\mathcal{M}}(q,N)]$ and $\mathbb{E}[R^{\mathcal{M}}(q,2)]$ is a constant independent of the value of S.

The proofs of Theorems 1.3.1 and 1.3.3 can be derived from the results in Lemma 1.3.2, 1.3.3 and 1.3.4, with a bound on $\mathbb{E}[R(0)^2]$ provided in Lemma 1.7.3 in Appendix 1.7.

1.4 On the Optimality of the Balancing Policy

In this section, we seek to verify that the balancing policy is indeed the right policy to use. We only provide analysis for N-opaque selling. The analysis for general k is similar and

omitted for improved exposition. For the two item case ($N = 2$), Elmachtoub et al. (2015) showed that the balancing action is optimal in every inventory state. However, for general N, we show that there exists a fundamental difference between the multi-item case and the two-item case. We prove that the balancing action, though not always optimal when $N \geq 3$, is asymptotically optimal.

We first provide an exact dynamic program to find the optimal policy for N-opaque selling. Since an inventory policy only makes decisions whenever an opaque customer arrives, we can model the problem as a semi-Markov decision process that minimizes the long run average cost per period, where the the state space of the dynamic program corresponds to the current inventory levels. In this formulation, each state transition period corresponds to one customer arrival. Thus, the holding cost incurred per unit per period is equal to $\frac{h}{\lambda}$. The set of all possible inventory states are of the form $x = (x_1, \ldots, x_N)$, where for each product i, $x_i \in \mathbb{Z}$ and $1 \leq x_i \leq S$. For any state x, if product i is consumed by the customer (directly or via an opaque product), then the inventory state becomes $x - e_i$ if $x_i > 1$ and $\vec{S}$ if $x_i = 1$. (Here e_i denotes the i^{th} unit vector.) Note that once the last unit of a product is consumed, i.e., the product stocks out, all inventory levels are immediately replenished to $\vec{S}$.

Let $g(x)$ be the inventory cost incurred at state x, including the expected holding cost per customer arrival and ordering cost if a replenishment occurred. The formula of $g(x)$ is

$$g(x) = \begin{cases} K + \frac{h}{\lambda} NS, & \text{if } x = \vec{S}, \\ \frac{h}{\lambda} \sum_{i=1}^{N} x_i, & \text{otherwise.} \end{cases}$$

Let γ^* denote the long-run average cost per unit sold under the optimal policy. We refer to

$g(x) - \gamma^*$ as the relative cost incurred in state x. With probability $\frac{1-q}{N}$, a customer arrives for product i and the next state is $x - e_i$ if $x_i > 1$ and $\vec{S}$ if $x_i = 1$. With probability q, an opaque customer arrives and the retailer chooses to transition to the state with the lowest cost. The total relative cost-to-go (under the optimal policy) until the next replenishment, $J(x)$, is defined according to the Bellman equations

$$J(x) = \begin{cases} g(x) - \gamma^* + \frac{1-q}{N} \sum_{i|x_i>1} J(x - e_i) + q \min\{0, \min_{i|x_i>1}\{J(x - e_i)\}\} & \text{if } \min_i x_i = 1 \\ g(x) - \gamma^* + \frac{1-q}{N} \sum_i J(x - e_i) + q \min_i\{J(x - e_i)\} & \text{if } \min_i x_i \geq 2 \end{cases} \tag{1.5}$$

By Proposition 7.4.1 of Bertsekas (2017), we have $J(\vec{S}) = 0$. Thus when $\min_i x_i = 1$ and an opaque customer arrives, one has the option to transition to a state with 0 cost, as reflected in the first case of Eq. (1.5). The values of $J(\cdot)$ and γ^* can be obtained by value iteration techniques. Note we can also infer the optimal policy π^* from (1.5) by the corresponding decisions for opaque customers.

There is an equivalent way to express $J(x)$. We extend the notation of the random variable $R^\pi(q, N)$ to $R^\pi(q, N; x)$, which denotes the number of customer to be served from inventory state x to the next replenishment under policy π. $R^\pi(q, N) = R^\pi(q, N; \vec{S})$ by definition and we assume that if $x = \vec{S}$, then x can be omitted in $R^\pi(q, N; x)$ for simplicity. For any state $x \neq \vec{S}$, $J(x)$ can be interpreted as the total relative cost incurred before going back to state $\vec{S}$, which is equivalent to the total relative cost incurred until next

replenishment. Then for any $x \neq \vec{S}$, $J(x)$ can also be written as

$$J(x) = \mathbb{E}\left[\sum_{j=1}^{R^{\pi^*}(q,N;x)} \left(\frac{h}{\lambda}\left(\sum_{i=1}^{N} x_i - j + 1\right) - \gamma^*\right)\right]$$
$$= \mathbb{E}[R^{\pi^*}(q,N;x)]\left(\frac{h}{2\lambda}\left(2\sum_{i=1}^{N} x_i + 1\right) - \gamma^*\right) - \frac{h}{2\lambda}\mathbb{E}[R^{\pi^*}(q,N;x)^2]. \quad (1.6)$$

1.4.1 Optimal Structure

In this subsection, we provide analysis on the structure of the optimal opaque fulfillment policy with N-opaque products. First, we provide upper and lower bounds on the optimal long run average cost γ^* in the following lemma. We use the result in Lemma 1.4.1 multiple times in the rest of the paper.

Lemma 1.4.1. $\frac{h}{2\lambda}(NS+N) + \frac{K}{NS} \leq \gamma^* \leq \frac{h}{2\lambda}(NS+\theta_q+1) + \frac{K}{NS-\theta_q}$.

The proof of Lemma 1.4.1 is provided in Appendix 1.7 and directly leverages Lemma 1.3.3. Note that if we assume that h, K, λ, N are all constants and S is sufficiently large, then from Lemma 1.4.1 we can conclude that γ^* is on the order of $\Theta(S)$.

To partially characterize the optimal structure in Theorem 1.4.1, we construct two subsets of inventory states. In one subset, the balancing action is optimal, while in the other the balancing action is no longer optimal. When the total inventory is no more than half of the original amount, $NS/2$, then the balancing action is optimal. When there is 1 unit of a product remaining but the total inventory is very high, then inventory balancing is sub-optimal.

Theorem 1.4.1 (Properties of the Optimal Allocation Policy). *(a) For any state x satisfying $\sum_{i=1}^{N} x_i \leq \lfloor \frac{NS}{2} \rfloor$, the balancing action is optimal.*

(b) Consider any state x where there is a product with only one unit on-hand. If

$$\sum_{i=1}^{N} x_i \geq \frac{NS+\theta_q}{2} + 1 + \frac{\lambda K/h}{NS-\theta_q} + \frac{2N^2-(1-q)N}{2(1-q)^2},$$

then the optimal action is not the balancing action.

The proof of Theorem 1.4.1 is provided in Appendix 1.7. In part (a) of Theorem 1.4.1, we show that the immediate reward in dynamic program (1.5) is always negative when the total inventory is less than $NS/2$, and as more customers are served, the cost-to-go function becomes smaller. Therefore, we want to serve as many customers as possible until the next replenishment to minimize the long run average cost.

In part (b) of Theorem 1.4.1, we provide a sufficient condition for the balancing action to be suboptimal. If there exists one product with only one unit on hand, and at the same time the total inventory is still quite high, then the balancing action is worse than the action that allocates the product with one unit on hand, and thus it is not optimal. Such a state corresponds to a large imbalance in inventory. For example, state $(1, S, \ldots, S)$ satisfies the condition in part (b). In this extreme example, the average holding cost in the current replenishment cycle is likely to be much higher than the long run average cost γ^* with high probability, no matter what policy is used. If there is replenishment and the inventory is back to state $\vec{S}$, the retailer will then pay γ^* per period on average (by definition). Hence, instead of staying in the current replenishment cycle with relatively higher costs, the retailer would prefer to replenish as soon as possible to restart the process. Thus given an opportunity to

allocate an opaque product, the retailer will choose the product with one unit left. Note that in this example, although the optimal action is different from the balancing action, the intuition behind the optimal action does not change: we want the inventory state to be more balanced, and going back to state $\vec{S}$ can be considered as a special action that leads to balancing.

The result in part (b) does not contradict with the optimality of the balancing policy when $N = 2$. Indeed, when $N = 2$, no inventory state can satisfy the condition. If one product has only one unit on-hand, then the total inventory must be less than or equal to $1 + S$, which is less than total inventory requirement in part (b).

Next, we show that the balancing policy, although not optimal, is near-optimal in the asymptotic sense, as described in Theorem 1.4.2 below.

Theorem 1.4.2 (**Balancing Policy is Near-Optimal**). *The balancing policy $\mathcal{M}$ has cost at most $1 + \frac{\theta_q}{NS-\theta_q}$ times the cost of the optimal policy.*

The proof of Theorem 1.4.2 is provided in Appendix 1.7. When S is large enough, the cost of balancing policy $\mathcal{M}$ is guaranteed to be close to the cost of optimal policy, within $\Theta(\frac{1}{S})$. Furthermore, from the results in Theorem 1.3.1, the order of cost saving achieved by the balancing policy $\mathcal{M}$ depends on $\frac{1}{\sqrt{S}}$, thus making a $\frac{1}{S}$ improvement negligible. So to capture the value of opaque selling (the cost savings), one can simply apply the balancing policy, rather than the optimal policy, and obtain almost all of the potential benefits from consumer flexibility.

We numerically test the performance of the balancing policy versus the optimal policy. For large S, Theroem 1.4.2 guarantees the near-optimality of the balancing policy. However,

for moderate or small S values, the theoretical bound in Theorem 1.4.2 might be loose. In fact, numerically, we observe that when S value is small, the balancing policy is essentially optimal. In our experiment, we solve for the optimal policy using relative value iteration. The cost for the balancing policy can be computed using equation (1.1) and (1.2), where $\mathbb{E}[R^{\mathcal{M}}(q,N)]$ and $\mathbb{E}[R^{\mathcal{M}}(q,N)^2]$ can be exactly computed using recursion. We fix $N = 3$ and $\lambda = 1$. All the other parameters are chosen randomly in moderate ranges. The value of S varies between 30 and 50, the holding cost h is randomly chosen between 0.5 and 1.5, and the ordering cost K is between 500 and 1500. We randomly pick q to be between 10% and 20%. Note that when $q = 0$ or $q = 1$ then $\mathcal{M}$ is optimal. 300 scenarios are randomly generated and we computed the relative cost difference for the two policies. In 127 scenarios, the balancing policy is optimal for every state. In the remaining 173 scenarios, the balancing policy is not optimal in every state, but costs approximately the same as the optimal policy. The maximum relative cost difference over all simulations is approximately 0.005%, which for all practical purposes is insignificant.

1.5 Numerical Experiments

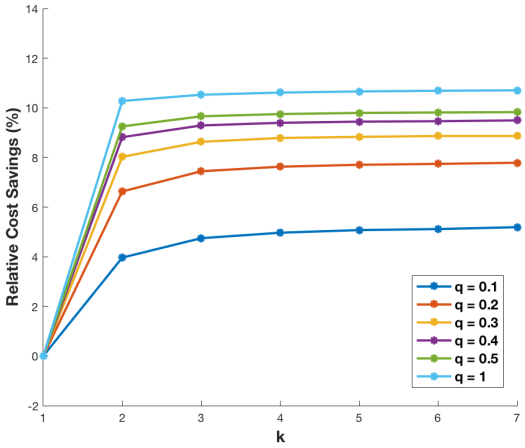
In this section, we provide numerical evidence to support our theoretical findings and consider several interesting extensions.

1.5.1 Small Degree of Opacity is Good Enough

We now provide numerical evidence that describes the cost advantage of opaque selling as well as the comparison between k-opaque selling and N-opaque selling. The set up of

the experiment is the following: a retailer sells a product with N different colors and we compare a non-opaque scheme to a k-opaque selling scheme. We fix the cost parameters to be $h = 1, K = 1000, \lambda = 1$. We try different percentage of opaque customers q from 10% to 100% and all k values from 1 to N. The results are summarized in Figure 1.3 and Table 1.1.

Figure 1.3: Cost Savings for k-Opaque Products



Note. $N = 7, S = 100, K = \$1000, h = \$1, \lambda = 1$.

Table 1.1: Percentage of Cost Savings Captured by 2-Opaque Comparing with N-Opaque (%)

q	$S = 50$	$S = 100$	$S = 150$	$S = 200$
1	95.24	96.24	97.32	97.35
0.9	95.15	96.17	97.24	97.35
0.8	94.98	96.00	97.19	97.24
0.7	94.56	95.77	97.00	97.08
0.6	93.88	95.37	96.58	96.72
0.5	92.73	94.67	96.08	96.34
0.4	90.91	93.37	95.17	95.55
0.3	87.53	91.16	93.61	94.15
0.2	82.55	86.60	89.86	90.91
0.1	75.40	77.42	81.09	82.51

Note. $N = 6, K = \$1000, h = \$1, \lambda = 1$.

In Figure 1.3 we plot the relative cost savings as defined by (1.4) for various degrees of opacity (q, k). Here we fix the reorder level S to be 100 and $N = 7$. In Figure 1.3, each curve represents a fixed q value. When $k = 1$, it is equivalent to the no opaque selling strategy so there is no savings. First, we can observe that the curve for $q = 0.5$ is very close to the curve for $q = 1$. Even when $q = 0.1$, the relative cost savings are around 40% of the savings of $q = 1$, for any k from 2 to N. Therefore, a small q value guarantees a large proportion of the total potential savings, no matter which k-opaque product is offered. Another observation from Figure 1.3 is that changing k from 1 to 2 can significant increase the cost savings, while increasing k from 2 to N does not provide much improvement. Especially for q greater than

0.3, the savings are almost the same for all k from 2 to N. Even for $q = 0.1$, the difference between 2-opaque and N-opaque is small. Therefore, for a fixed q value, 2-opaque products can capture most of the potential savings of the N-opaque products.

To further illustrate the effectiveness of 2-opaque products, in Table 1.1 we provide $\frac{\mathcal{K}(0)+\mathcal{H}(0)-\mathcal{K}^{\mathcal{M}}(q,2)-\mathcal{H}^{\mathcal{M}}(q,2)}{\mathcal{K}(0)+\mathcal{H}(0)-\mathcal{K}^{\mathcal{M}}(q,N)-\mathcal{H}^{\mathcal{M}}(q,N)}$, which is the percentage of cost savings captured by 2-opaque selling comparing with N-opaque selling. In this experiment $N = 6$ and other parameters are the same as the previous experiment. When the value of S and q are large, 2-opaque products can capture over 95% of the cost savings comparing with N-opaque selling. Even when S and q are small, 2-opaque products still capture 75% of the cost savings. We also observe that as S increases, the difference of the performance between 2-opaque products and N-opaque products decreases. Overall, Figure 1.3 and Table 1.1 validates our main result in Theorem 1.3.3 that a minimal degree of opacity can capture most of the cost savings of the fully flexible scheme corresponding $(1, N)$.

Note that in this subsection, we compare different k-opaque selling strategies based on the same q value. However, in order to achieve same level of q, the required reward for an N-opaque product would be much higher than that of a 2-opaque product. Therefore, in the next subsection, we provide a more detailed computational experiment to illustrate that 2-opaque products are even more powerful than N-opaque products in terms of the overall profitability.

1.5.2 Experiment on Profitability

In this subsection, we extend our setting to the case where a retailer uses discount to attract opaque customers in order to maximize profit (although other reward strategies are possible). We design a numerical experiment to evaluate the profitability of 2-opaque selling and N-opaque selling. Our experiment represents a setup where the retailer knows the cost parameters, inventory policy, and consumers' choice behaviors. The price is chosen to maximize profit in the traditional selling strategy without opaque products. Then, keeping the price and inventory policy fixed (a conservative approach), the retailer computes the optimal discount level when considering a given opaque selling strategy. Our baseline for comparison is the optimal traditional selling strategy.

To specify customer behavior, we use three commonly used random utility models. We set the mean valuation for an item to be 100 and standard deviation to be 10. We assume the customer valuations for every product are given by independent random variables, and for the no purchase option, the valuation is 0. We also assume the customers are risk-neutral, meaning they value the opaque product as the average of their valuations. We use the Uniform distribution, Normal distribution, and Logistic distribution for consumers' valuations. Note that the case of Logistic distribution is equivalent to the so called Multinomial Logit choice model.

The experiment procedure can be summarized as the following:

- We fix an order-up-to level S for the traditional, 2-opaque, and N-opaque strategies.
- We then find the optimal price p^* that maximizes the profit of the traditional strategy.
- Fixing the price p^* and the order-up-to level S, we then find the corresponding optimal

discounts for each of the 2-opaque and N-opaque strategies.

- Finally, we compute the inventory costs, revenue, and profit under given scenarios for the traditional, 2-opaque, and N-opaque strategies.

Note that we use the same price p^* and S for each strategy because in practice the retailer would not be expected to change their core pricing and inventory structure when introducing opaque products. We let $N = 5$, the holding cost $h = \$1$ per day per unit and the arrival rate $\lambda = 50$ customers per day. We consider a per unit purchasing cost of \$50 per unit. We vary the set up cost K from \$1000 to \$10000. Our choice to vary along the ordering cost parameter is motivated by the fact that this parameter may vary the most in practice. For the order-up-to level S, we use values of 100, 150, and 200.

We summarize the output from the experiment in Table 1.2. We compute the discount and the percentage of opaque customers for N-opaque selling and 2-opaque selling. In addition, we compute the change in profit, revenue, cost, and number of purchasing customers compared with the no opaque selling strategy. Here all the values are computed by simulation. In Table 1.2, N-opq represents the result for N-opaque selling and 2-opq represents the result for 2-opaque selling.

In Table 1.2, using 2-opaque selling strategy provides higher increase in profit, which is mainly contributed by the inventory cost savings. 2-opaque products provide approximately the same level of revenue increase. Under the same price and order-up-to level, the optimal discount for 2-opaque products is much lower than the discount for N-opaque products, while the fraction of customers who buy 2-opaque products is much higher than for the N-opaque strategy. By using 2-opaque products, the retailer can offer a much lower discount

Table 1.2: Comparison of 2-Opaque Product and N-Opaque Product on Profitability

Uniform Distribution

(K, S)	discount (%)		q (%)		cost saving per unit sold (%)		increase in demand (%)		change in revenue (%)		increase in profit (%)	
	N-opq	2-opq	N-opq	2-opq	N-opq	2-opq	N-opq	2-opq	N-opq	2-opq	N-opq	2-opq
(1000,100)	6.29	1.06	7.52	27.01	4.72	7.61	0.63	0.44	0.16	0.15	0.42	1.18
(5000,100)	7.01	1.34	11.52	33.42	6.44	8.50	1.14	0.65	0.32	0.20	1.91	3.42
(10000,100)	7.62	1.52	15.76	37.00	7.56	8.94	1.70	1.10	0.48	0.53	5.26	8.77
(1000,150)	6.69	1.07	9.52	27.15	5.39	6.76	0.84	0.39	0.19	0.10	0.60	1.33
(5000,150)	6.84	1.25	10.50	31.25	5.73	7.23	0.98	0.51	0.25	0.12	1.48	2.58
(10000,150)	7.18	1.34	12.52	33.20	6.33	7.49	1.42	0.78	0.51	0.33	3.30	5.11
(1000,200)	6.69	1.07	9.52	27.15	5.21	6.21	0.84	0.39	0.19	0.10	0.85	1.62
(5000,200)	7.05	1.06	11.64	26.92	5.74	6.28	1.05	0.39	0.22	0.10	1.43	2.56
(10000,200)	7.01	1.25	11.52	31.33	5.72	6.60	1.14	0.56	0.32	0.17	2.63	4.08

Normal Distribution

(K, S)	discount (%)		q (%)		cost saving per unit sold (%)		increase in demand (%)		change in revenue (%)		increase in profit (%)	
	N-opq	2-opq	N-opq	2-opq	N-opq	2-opq	N-opq	2-opq	N-opq	2-opq	N-opq	2-opq
(1000,100)	6.21	1.87	11.60	37.00	6.50	8.79	1.23	1.11	0.50	0.41	0.76	1.13
(5000,100)	6.73	1.95	14.65	38.12	7.47	9.18	1.97	1.83	0.97	1.08	2.79	4.22
(10000,100)	7.25	2.22	18.78	42.65	8.39	9.51	3.14	2.60	1.73	1.62	7.32	9.88
(1000,150)	6.21	1.48	11.62	30.63	6.24	7.17	1.26	0.61	0.53	0.15	1.02	1.25
(5000,150)	6.21	1.48	11.62	30.63	6.12	7.28	1.26	0.61	0.53	0.15	1.97	2.60
(10000,150)	6.98	2.42	16.62	45.78	7.39	8.56	2.56	2.62	1.37	1.48	4.50	5.70
(1000,200)	6.21	1.28	11.82	25.53	5.98	6.18	1.24	0.43	0.50	0.10	1.25	1.57
(5000,200)	6.54	2.05	13.75	39.66	6.57	7.56	1.87	1.69	0.96	0.86	2.46	3.01
(10000,200)	6.54	1.85	13.75	36.29	6.39	7.18	1.87	1.41	0.96	0.73	3.64	4.56

Logistic Distribution

(K, S)	discount (%)		q (%)		cost saving per unit sold (%)		increase in demand (%)		change in revenue (%)		increase in profit (%)	
	N-opq	2-opq	N-opq	2-opq	N-opq	2-opq	N-opq	2-opq	N-opq	2-opq	N-opq	2-opq
(1000, 100)	5.45	1.59	9.91	32.25	5.66	8.47	0.95	1.10	0.40	0.58	0.74	1.50
(5000, 100)	6.30	2.07	15.53	40.69	7.65	9.40	1.85	2.06	0.85	1.20	2.80	4.32
(10000, 100)	6.72	2.24	18.51	43.17	8.38	9.64	3.09	3.29	1.81	2.29	7.92	10.91
(1000, 150)	5.45	1.49	9.91	30.43	5.48	7.43	0.95	1.00	0.40	0.54	0.96	1.72
(5000, 150)	5.92	1.68	12.84	33.78	6.43	7.82	1.45	1.41	0.68	0.84	2.20	3.36
(10000, 150)	6.18	1.96	14.51	38.68	6.87	8.21	1.96	2.17	1.04	1.39	4.47	6.31
(1000, 200)	5.45	1.49	9.91	30.43	5.30	6.87	0.95	1.00	0.40	0.54	1.24	2.04
(5000, 200)	5.74	1.68	11.68	33.96	5.82	7.13	1.20	1.28	0.52	0.70	2.09	3.12
(10000, 200)	6.20	1.67	14.75	33.66	6.47	7.15	1.78	1.49	0.84	0.91	3.58	5.05

Note. $N = 5, h = \$1, \lambda = 50$. The purchasing cost is \$50 per unit. Consumers' valuation of an item is a random variable with mean 100 and standard deviation 10, where the distribution is specified in the table.

while simultaneously having more customers purchasing opaque products, which leads to more savings in inventory cost. We also find that the performance was similar under the

three random utility models. This implies that the effectiveness of 2-opaque products is not restricted to a specific choice model.

1.5.3 Extensions to General Inventory Settings

In this subsection, we explore the benefit of opaque selling in broader inventory settings with lead time, backorders, and lost sales. We are interested in the cost savings with limited degree of opacity. Specifically, we compute the relative cost savings as defined in Eq. (1.4) for four schemes with degree of opacity $(q = 0.1, k = 2)$, $(q = 0.2, k = 2)$, $(q = 0.1, k = N)$ and $(q = 0.2, k = N)$. Moreover, we shall also see how much extra benefit can be provided by re-optimizing the can-order policy when introducing opaque products.

We consider a setting where a retailer offers $N = 5$ products. There is one major joint ordering cost K and no individual ordering costs. We simulate 1 million customer arrivals to approximate the long run average cost of a system, since no closed-form expressions are available and the dynamic programming approach suffers from the curse of dimensionality. We refer to Atkins and Iyogun (1988) and Melchiors (2002) for commonly used cost parameters in multi-product inventory systems and choose $h = \$6$ per unit per day, a backorder cost of $b = \$10$ or $\$30$ per unit per day, and the one-time lost sale penalty of $l = \$30$ or $\$50$ per unit. The arrival rate of customers is $\lambda = 20$ per day. We also vary the replenishment lead time L from 0 to 3.

The inventory replenishment policy we consider here is the can-order policy (s, c, S), that is when the inventory level of some item drops to s, then the retailer orders all items whose inventory level is lower than c back to S. Usually c is controlled by the individual ordering

cost and we consider the case where there is only one major ordering cost and no individual ordering cost. Therefore, letting $c = S-1$ is reasonable and we just abbreviate the can-order policy as an (s, S) policy.

The results are summarized in Table 1.3. We specify the parameters of the inventory system in the first group of columns. For a given set of parameters, we first consider the traditional selling strategy without opaque products, search for the optimal (s, S) policy (in column "No OPQ") and compute the long run average costs. Next, we consider the 2-opaque scheme while keeping the (s, S) policy the same, and compute the relative cost savings (in column "2-OPQ"). Then, again with the same (s, S) policy, we replace 2-opaque options with the N-opaque option, and compute the relative cost savings comparing with the no opaque selling strategy (in column "N-OPQ"). Finally, we keep the N-opaque option and re-optimize the (s, S) ordering policy (in column "With N-OPQ"). The relative cost savings compared with the no opaque selling strategy is recorded in the "N-OPQ with re-optimzied policy" column. In all experiments, we use the on-hand inventory balancing policy to fulfill opaque demands. Intuitively, when there is lead time, the retailer wants to avoid the expensive lost sales or backorders, and allocating the highest on-hand inventory product to opaque customers can help avoid such expensive costs.

The first set of experiments in Table 1.3 uses the setting in our theoretical results, where lead time is zero and we do not allow for backorders or lost sales. We choose three different ordering cost levels and two values of q. When K is at a small and medium level, the optimal (s, S) policy does not change when adding the opaque products and therefore re-optimizing the ordering policy does not provide any extra value here. When K is large, the optimal (s, S) policy changes, but the resulting change in cost saving is small. Re-optimizing the

Table 1.3: Performance of Opaque Products under General Inventory Settings

No backlogging/lost sales, $L = 0$

Parameters		Optimal (s, S) policy		Relative Cost Savings (%)		
		No OPQ	With N-OPQ	2-OPQ	N-OPQ	N-OPQ with re-optimized policy
$q = 0.1$	$K = 100$	(0, 7)	(0, 7)	2.90	4.22	4.22
	$K = 1000$	(0, 19)	(0, 19)	3.83	5.20	5.20
	$K = 10000$	(0, 58)	(0, 54)	3.96	5.21	5.37
$q = 0.2$	$K = 100$	(0, 7)	(0, 7)	5.67	7.93	7.93
	$K = 1000$	(0, 19)	(0, 19)	7.04	9.14	9.14
	$K = 10000$	(0, 58)	(0, 55)	6.88	8.26	8.44

Backlogging

Parameters			Optimal (s, S) policy		Relative Cost Savings (%)		
			No OPQ	With N-OPQ	2-OPQ	N-OPQ	N-OPQ with re-optimized policy
$q = 0.1$	$b = 10$	$L = 1$	(-3, 9)	(-2, 9)	3.27	5.65	5.85
		$L = 2$	(1, 14)	(1, 14)	4.76	8.03	8.03
		$L = 3$	(5, 18)	(5, 18)	5.86	7.90	7.90
	$b = 30$	$L = 1$	(0, 11)	(0, 11)	4.80	7.78	7.78
		$L = 2$	(4, 16)	(4, 16)	6.21	9.45	9.45
		$L = 3$	(9, 20)	(8, 20)	7.03	10.44	11.46
$q = 0.2$	$b = 10$	$L = 1$	(-3, 9)	(-2, 9)	5.56	8.59	10.13
		$L = 2$	(1, 14)	(1, 14)	8.21	12.54	12.54
		$L = 3$	(5, 18)	(5, 18)	10.16	15.06	15.06
	$b = 30$	$L = 1$	(0, 11)	(0, 11)	9.04	13.57	13.57
		$L = 2$	(4, 16)	(4, 15)	11.06	15.95	16.10
		$L = 3$	(9, 20)	(8, 20)	12.51	17.41	18.82

Lost sales

Parameters			Optimal (s, S) policy		Relative Cost Savings (%)		
			No OPQ	With N-OPQ	2-OPQ	N-OPQ	N-OPQ with re-optimized policy
$q = 0.1$	$l = 30$	$L = 1$	(2, 12)	(2, 12)	4.05	6.50	6.50
		$L = 2$	(6, 16)	(6, 16)	4.15	6.96	6.96
		$L = 3$	(9, 19)	(9, 19)	4.58	7.57	7.57
	$l = 50$	$L = 1$	(3, 13)	(3, 13)	4.66	7.00	7.00
		$L = 2$	(7, 17)	(7, 17)	5.30	8.02	8.02
		$L = 3$	(11, 21)	(11, 21)	5.50	8.22	8.22
$q = 0.2$	$l = 30$	$L = 1$	(2, 12)	(2, 12)	7.44	11.28	11.28
		$L = 2$	(6, 16)	(6, 16)	7.55	11.60	11.60
		$L = 3$	(9, 19)	(9, 19)	8.32	12.72	12.72
	$l = 50$	$L = 1$	(3, 13)	(3, 12)	8.65	12.43	12.53
		$L = 2$	(7, 17)	(7, 17)	9.46	13.58	13.58
		$L = 3$	(11, 21)	(11, 21)	9.95	14.12	14.12

Note. $N = 5, h = \$6, \lambda = 20$. $K = \$150$ in the backlogging and lost sales case.

inventory ordering policy only provides less than 0.2% increase in cost savings while solely adding opaque product provides the majority of cost savings.

The second experiment allows backorders but no lost sales and the third experiment allows lost sales but no backorders. In both experiments, we fix the joint ordering cost K to be \$150 and vary the lead time L and q value. First, we can observe that opaque products still provide significant inventory cost savings in general settings and a limited flexible scheme (2-opaque products) is still very effective. Moreover, the optimal ordering policy barely changes after adding N-opaque products, and the resulting increase in cost saving is also marginal in all cases.

Motivated by the fact that the retailer may not be able to adjust the replenishment policy due to an established supply chain contract, or some physical constraints, our theoretical analysis does not re-optimize the inventory ordering policy. Indeed, our experiments suggest that the cost saving benefits mainly come from consumer flexibility, where adjusting replenishment policy adds little benefit. In the case with only holding costs and ordering costs, an optimal replenishment policy would roughly balance the average holding cost and ordering cost. In Theorem 1.3.2, we show that adding opaque products can reduce the average holding cost and ordering cost by the same order. After adding opaque products, the originally optimal replenishment policy will still roughly balance the holding and ordering cost, and therefore it is likely to remain optimal. This provides a theoretical explanation of the findings from our numerical experiment.

Overall, we can conclude that under various inventory settings, *i)* opaque selling is a powerful tool to drive inventory costs down, *ii)* a limited flexible design can capture most of the cost saving benefits, and *iii)* adding the opaque product itself provides almost all of the

cost saving benefits while adjusting the ordering policy only provides a second-order value (oftentimes no value).

1.6 Conclusion and Discussion

In this paper, we study the value of an opaque selling strategy as a vehicle to leverage consumer flexibility in online retailing. The flexibility can essentially be used to risk pool the demand and drive down inventory costs. Our key theoretical result, under a particular but revealing model, is that limited flexibility (a small degree of opacity) has the same order of cost savings as a fully flexible scheme. Specifically, i) having a small proportion of customers being flexible is nearly as optimal as all customers buying opaque products; ii) we can restrict to the use of 2-opaque products, rather than N-opaque products, in terms of capturing this cost benefit. We anticipate that such 2-opaque selling tactics may become increasingly popular both in research and practice.

Our study provides several practical guidance to online platforms who want to embrace consumer flexibility. The main takeaway is that the first order effect comes from adopting such a strategy with a small incentive, while optimizing the incentive (q value) only adds secondary benefits. Since the magnitude of the benefit does not rely on a specific relationship between the reward and q, it is unnecessary to model the consumers' behavior against the incentive for buying an opaque product. This incentive may come from reward points, discounts, or coupons, for example. Instead, a market survey (on consumers' reactions to some incentive design) would be enough to convince a retailer to adopt the opaque selling strategy and estimate q. Also, the retailer should consider a limited flexible design such as 2-opaque

products. If the same reward is offered, then 2-opaque options would attract many more consumers to be flexible and the overall benefit can be anticipated to be more significant. In the case where consumer flexibility is incentivized by discounts, limited flexible design tends to increase more profit overall than full flexibility ($k = N$), since it is significantly easier to induce customers to purchase 2-opaque products (and trivially easier to convince a small q versus $q = 1$ fraction of them).

We believe our results provide strong theoretical motivation for online platforms to embrace consumer flexibility, especially especially in the context of retail where cost reductions can lead to significant profit gains. Future work may consider alternative settings that embrace consumer flexibility, as well as analyzing the value of flexibility from opaque selling in more complex supply chain environments.

1.7 Additional Proofs

Proof of Proposition 1.2.1. We only provide the proof for the case $k = N$. For general k values, the proof is essentially the same.

To maximize $\mathbb{E}\left[R^{\pi}(q, N)\right]$, we can formulate the problem as a dynamic program. Let $W(x) := \max_{\pi} \mathbb{E}\left[R^{\pi}(q, N; x)\right]$ denote the maximum possible expected number of customers served before the next replenishment. We can write the dynamic programming recursion as

$$W(x) = \begin{cases} 1 + \frac{1-q}{N} \sum_{i=1}^{N} W(x - e_i) + q \max_{i=1,\ldots,N} W(x - e_i), & \text{if } \min x_i \geq 1, \\ 0, & \text{if } \min x_i = 0. \end{cases} \tag{1.7}$$

To show that the balancing policy $\mathcal{M}$ maximizes $\mathbb{E}[R^{\pi}(q)]$, we only need to show that in any inventory state x such that $x_1 \geq 1$ and $x_{i^*} = \max x_i$,

$$\max_{i=1,\ldots,N} W(x - e_i) = W(x - e_{i^*}). \tag{1.8}$$

To prove (1.8), it suffices to show that for any i and j such that $1 \leq x_i \leq x_j$,

$$W(x - e_i + e_j) \leq W(x). \tag{1.9}$$

We can prove (1.9) inductively on the total inventory on-hand in state x. In the base case, we consider the inventory state with the smallest possible inventory on hand. The state x must be $(1, 1, \ldots, 1)$. In this case, Eq. (1.9) holds trivially because $W(x - e_i + e_j) = 0$ and $W(x) = 1$.

Assume that the inductive hypothesis (1.9) holds for any state where the total inventory on hand is less than or equal to $l - 1$. Then for an inventory state x such that $\sum_{k=1}^{N} x_i = l$, $\min x_k \geq 1$, and $x_i \leq x_j$, we want to show that the inductive hypothesis still holds.

We first apply the DP recursion (1.7) to state $x - e_i + e_j$ and x and then compare each pair of components in the equation.

$$W(x - e_i + e_j) = 1 + \frac{1-q}{N} \sum_{k=1}^{N} W(x - e_i + e_j - e_k) + q \max_{k=1,\ldots,N} W(x - e_i + e_j - e_k), \tag{1.10}$$

$$W(x) = 1 + \frac{1-q}{N} \sum_{i=1}^{N} W(x - e_k) + q \max_{i=1,\ldots,N} W(x - e_k). \tag{1.11}$$

The total inventory on-hand in state $x - e_i + e_j - e_k$ and $x - e_k$ are less than l. If $k \neq j$, then $(x - e_k)_i \leq (x - e_k)_j$. According to the inductive hypothesis,

$$W(x - e_i + e_j - e_k) \leq W(x - e_k). \tag{1.12}$$

If $k = j$, then

$$x - e_k = x - e_j,$$

$$x - e_i + e_j - e_k = x - e_i.$$

When $x_i < x_j$, then $(x - e_j)_i \leq (x - e_j)_j$ and (1.12) holds according to the inductive hypothesis. When $x_i = x_j$, then $x - e_j$ and $x - e_i$ are equivalent inventory states (one can exchange the index of item i and j) and (1.12) holds with equality. Therefore, under all cases, $W(x - e_i + e_j - e_k) \leq W(x - e_k)$ for all k. Therefore (1.10) is at most (1.11) implying $W(x - e_i + e_j) \leq W(x)$ for all states x with total inventory on-hand l. This completes the induction.

Now we may apply Eq. (1.9) to the state $x - e_{i^*}$. When $x_j < x_{i^*}$, $(x - e_{i^*})_j \leq (x - e_{i^*})_{i^*}$. Since $x - e_j = x - e_{i^*} - e_j + e_{i^*}$, by Eq. (1.9) we can conclude that $W(x - e_j) \leq W(x - e_{i^*})$, which implies that allocating the maximum inventory item to an opaque customer maximizes $W(x)$. When $x_j = x_{i^*}$, state $x - e_j$ and $x - e_{i^*}$ are the same state and $W(x - e_j) = W(x - e_{i^*})$. Therefore, the balancing policy $\mathcal{M}$ maximizes the expected number of customers to be served between two consecutive replenishments. □

Proof of Lemma 1.3.2. In this proof, we rely on our connection to the balls-into-bins

problem and make use of the following result from Raab and Steger (1998). The result exactly characterizes the tail behavior on the number of balls in the maximum loaded bin when $m \gg N \log N$ are thrown uniformly at random into N bins. We note that Raab and Steger (1998) presented a slightly weaker version of Lemma 1.7.1 as their main result, but they directly proved the slightly stronger version that we present below.

Lemma 1.7.1 (Theorem 1 in Raab and Steger (1998))**.** *Let M be the random variable that counts the maximum number of balls in any bin, if we throw m balls independently and uniformly at random into N bins. Then $\mathbb{P}[M(m) \geq k_\alpha] = o(1)$ if $\alpha > 1$ and $\mathbb{P}[M(m) \geq k_\alpha] = 1 - o(1)$ if $0 < \alpha < 1$, where*

$$k_\alpha = \begin{cases} \frac{m}{N} + \alpha\sqrt{2\frac{m}{N}\log N}, & \text{if } N\log N \ll m \leq \Theta(N(\log N)^3), \\ \frac{m}{N} + \sqrt{\frac{2m\log N}{N}(1 - \frac{1}{\alpha}\frac{\log^{(2)} N}{2\log N})}, & \text{if } m \gg N(\log N)^3. \end{cases} \tag{1.13}$$

(Note that we shall always ensure that $\alpha > \frac{\log^{(2)} N}{2\log N}$ so that the result is well-defined and shall use different α values for the analysis.) Armed with Lemmas 1.3.1 and 1.7.1, we may now proceed to bound $\mathbb{E}[R(0)]$ by conditioning on the event that $R(0)$ exceeds a particular target m_α.

We now carefully define m_α to be the number of balls needed such that $k_\alpha = S$, thus from Eq. (1.13), we have for any $\alpha > \frac{\log^{(2)N}}{2\log N}$,

$$\begin{aligned} m_\alpha &= NS - N\left(\sqrt{2\alpha^2 S\log N + (\alpha^2\log N)^2} - \alpha^2\log N\right) \\ &= NS - \Theta(N\sqrt{S\log N}) \end{aligned}$$

if $\log N \ll S \leq \Theta((\log N)^3)$, and

$$\begin{aligned} m_\alpha &= NS - N\Bigg[\sqrt{2S\log N(1-\frac{1}{\alpha}\frac{\log^{(2)} N}{2\log N}) - \left(\log N\left(1-\frac{1}{\alpha}\frac{\log^{(2)N}}{2\log N}\right)\right)^2} \\ &\quad - \log N\left(1-\frac{1}{\alpha}\frac{\log^{(2)N}}{2\log N}\right)\Bigg] \\ &= NS - \Theta(N\sqrt{S\log N}) \end{aligned}$$

if $S \gg (logN)^3$. Therefore, we have

$$m_\alpha = NS - \Theta(N\sqrt{S\log N}),\ \forall\ \alpha > \frac{\log^{(2)N}}{2\log N}. \tag{1.14}$$

By assumption, under the case $S \gg (\log N)^3$, $m_\alpha \gg N(\log N)^3$ and under the case $\log N \ll S \leq \Theta((\log N)^3)$, $N\log N \ll m_\alpha \leq \Theta(N(\log N)^3)$, which satisfies the condition in Lemma 1.7.1. So we can now reshape the result of Lemma 1.7.1 as the following statement which we shall use for the remainder of the proof: *If we throw m_α balls independently and uniformly at random into N bins, $\mathbb{P}[M(m_\alpha) \geq S] = o(1)$ if $\alpha > 1$ and $\mathbb{P}[M(m_\alpha) \geq S] = 1 - o(1)$ if $0 < \alpha < 1$.*

Next, we derive an upper bound on $\mathbb{E}[R(0)]$ using Lemma 1.7.1 with $\frac{\log^{(2)} N}{2\log N} < \alpha < 1$:

$$\begin{aligned} \mathbb{E}[R(0)] =& \mathbb{E}[R(0)|R(0) \leq m_\alpha]\mathbb{P}[R(0) \leq m_\alpha] + \mathbb{E}[R(0)|R(0) > m_\alpha]\mathbb{P}[R(0) > m_\alpha] \\ (R(0) \leq NS \text{ w.p. } 1)\ \leq& m_\alpha\mathbb{P}[R(0) \leq m_\alpha] + cN\mathbb{P}[R(0) > m_\alpha] \\ =& NS - (NS - m_\alpha)\mathbb{P}[R(0) \leq m_\alpha] \\ \text{(Lemma 1.3.1)} \quad =& NS - (NS - m_\alpha)\mathbb{P}[M(m_\alpha) \geq S] \end{aligned}$$

(Lemma 1.7.1) $= NS - (NS - m_\alpha)(1 - o(1))$

(Eq. (1.14)) $= NS - \Theta(N\sqrt{S \log N})(1 - o(1))$.

Therefore, $\theta_S = NS - \mathbb{E}[R(0)] = \Omega\left(N\sqrt{S \log N}\right)$. □

Proof of Lemma 1.3.3. The lower bound on θ_q is trivial since the largest possible value for $R^{\mathcal{M}}(q, N)$ is $N(S-1)+1$.

Next, we derive the lower bound by again making a connection of our inventory management problem to a balls-into-bins problem, namely the one studied by Peres et al. (2010). Recall that a balls-into-bins process is characterized by a distribution vector $\mathbf{p} = (p_1, \ldots, p_N)$, where p_i is the probability a ball is placed in the i-th most loaded bin. After t balls have been thrown following $\mathbf{p}$, let $y(t)$ be the vector where the i-th component denotes the load of the i-th most loaded bin minus the average load. Let $\Gamma(t)$ be a potential function on the load imbalance, defined formally as

$$\Gamma(t) = \sum_{i=1}^{N} \exp(c_1 \epsilon y_i) + \sum_{i=1}^{N} \exp(-c_1 \epsilon y_i).$$

Here c_1 is some constant. Note that $\Gamma(t)$ is smaller as the loads of the bins are spread more uniformly. In fact, let $Gap(t)$ be defined as the maximum load minus the average load when the total number of balls is t. By definition, $Gap(t) = M(t) - \frac{t}{N}$, where $M(t)$ is the random variable defined in Section 1.3.1 denoting the load of the maximally loaded bin after t balls have been thrown. Applying Lemma 1.3.1, we have the following relationship on $R^{\mathcal{M}}(q, N)$

and $Gap(t)$,

$$\mathbb{P}[R^{\mathcal{M}}(q,N) \leq t] = \mathbb{P}[M(t) \geq S] = \mathbb{P}[Gap(t) \geq S - t/N]. \tag{1.15}$$

By definition, we also have that $Gap(t) = y_1(t)$ and therefore,

$$\Gamma(t) \geq e^{c_1 \epsilon Gap(t)}. \tag{1.16}$$

Peres et al. (2010) provides an upper bound on $\mathbb{E}[\Gamma(t)]$, which is a constant regardless of the number of balls that has been thrown. This shall allow us to bound the tail probabilities of $Gap(t)$, and therefore the CDF of $R^{\mathcal{M}}(q,N)$ from Eq. (1.15). Note that the assumptions we use in Lemma 1.7.2 are slightly weaker than the assumptions in Peres et al. (2010), but their proof only require this weaker version of assumptions and the result still holds.

Lemma 1.7.2 (Theorem 2.2 and 2.3 in Peres et al. (2010))**.** *When the distribution vector* $\mathbf{p}$ *satisfies the assumptions* $p_i \leq p_{i+1}$, $p_{\frac{N}{3}} \leq \frac{1-4\epsilon}{N}$ *and* $p_{\frac{2N}{3}+1} \geq \frac{1+4\epsilon}{N}$, *then there exists constant* c_1, c_2 *such that for any* $t \geq 0$,

$$\mathbb{E}[\Gamma(t)] \leq \frac{c_2}{\epsilon^7} N. \tag{1.17}$$

Equation (1.17) *also holds for any distribution vector* $\mathbf{p}'$ *that majorizes* $\mathbf{p}$.

We will use Lemma 1.7.2 and Eq. (1.16) to show a tail bound on $Gap(t)$ under the

assumptions of Lemma 1.7.2. For any k,

$$\begin{aligned}
\mathbb{P}\left[Gap(t) \geq k\frac{\log N}{c_1\epsilon} + \frac{\log\left(\frac{c_2}{\epsilon^7}\right)}{c_1\epsilon}\right] &= \mathbb{P}\left[e^{c_1\epsilon Gap(t)} \geq e^{c_1\epsilon k\frac{\log N}{c_1\epsilon} + c_1\epsilon\frac{\log\left(\frac{c_2}{\epsilon^7}\right)}{c_1\epsilon}}\right] \\
&\leq \mathbb{P}\left[\Gamma(t) \geq e^{k\log N + \log\left(\frac{c_2}{\epsilon^7}\right)}\right] \\
&= \mathbb{P}\left[\Gamma(t) \geq N^k \cdot \frac{c_2}{\epsilon^7}\right] \\
&\leq \mathbb{P}\left[\Gamma(t) \geq N^{k-1}\mathbb{E}[\Gamma(t)]\right] \\
&\leq N^{1-k}. \qquad (1.18)
\end{aligned}$$

The first inequality follows from Eq. (1.16). The second inequality follows from Lemma 1.7.2 and the last inequality follows from Markov's Inequality. We shall use the tail bound in (1.18) later in computing the lower bound of $\mathbb{E}[R^{\mathcal{M}}(q,N)]$.

The connection to our problem is that the distribution vector $\mathbf{p}'$ characterizing the probability of customers choosing the i-th lowest inventory product under the balancing policy $\mathcal{M}$ majorizes a distribution vector $\mathbf{p}$ which satisfies the condition in Lemma 1.7.2. Specifically, $\mathbf{p}' = (\frac{1-q}{N}, ..., \frac{1-q}{N}, \frac{1-q}{N} + q)$ and $p_i = \frac{1-q}{N} + \frac{2(i-1)q}{N(N-1)}$ for $i = 1, \ldots, N$. For any $q \in (0, 1]$, it is easy to see that $\mathbf{p}' \succeq \mathbf{p}$. $\mathbf{p}$ satisfies the condition in Lemma 1.7.2 when $\epsilon = \frac{q}{12}$. Therefore, we can apply the results in Lemma 1.7.2 and Eq. (1.18) holds for the balls-into-bins process associated with $\mathbf{p}'$.

Now we can derive a lower bound on $\mathbb{E}[R^{\mathcal{M}}(q)]$:

$$\begin{aligned}
\mathbb{E}[R^{\mathcal{M}}(q,N)] &= \sum_{t=0}^{N(S-1)+1} \mathbb{P}[R^{\mathcal{M}}(q,N) \geq t] \\
&= \sum_{t=0}^{N(S-1)+1} (1 - \mathbb{P}[R^{\mathcal{M}}(q,N) < t])
\end{aligned}$$

$$= N(S-1)+2-\sum_{t=S}^{N(S-1)+1}\mathbb{P}[R^{\mathcal{M}}(q,N)<t])$$
$$\geq N(S-1)+2-\sum_{t=S}^{N(S-1)+1}\mathbb{P}[Gap(t)\geq S-t/N]), \tag{1.19}$$

where the last inequality follows from Eq. (1.15).

Now we only need to find an upper bound for $\sum_{t=c}^{N(S-1)+1}\mathbb{P}[Gap(t)\geq S-t/N])$. For simplicity, define $\eta := 2\left\lceil\frac{N\log\frac{c_2N}{\epsilon^7}}{c_1\epsilon}\right\rceil+3$ and $\eta=\Theta\left(\frac{N}{q}\log\frac{N}{q}\right)$. In particular, we want to show that $\theta_q = NS-\mathbb{E}[R^{\mathcal{M}}(q,N)] = O\left(\frac{N}{q}\log\frac{N}{q}\right)$. We only need to show that $\sum_{t=c}^{N(S-1)+1}\mathbb{P}[Gap(t)\geq S-t/N])=O(\eta)$ and then we can yield the final result.

$$\begin{aligned}
\sum_{t=S}^{N(S-1)+1}\mathbb{P}[Gap(t)\geq S-t/N]) &= \sum_{t=S}^{NS-\eta}\mathbb{P}[Gap(t)\geq S-t/N]) + \sum_{t=NS-\eta+1}^{N(S-1)+1}\mathbb{P}[Gap(t)\geq S-t/N]) \\
&\leq \sum_{t=S}^{NS-\eta}\mathbb{P}[Gap(t)\geq S-t/N])+\eta \\
&= \sum_{t=S}^{NS-\eta}\mathbb{P}\left[Gap(t)\geq\left(\frac{S-t/N-\frac{\log\left(\frac{c_2}{\epsilon^7}\right)}{\alpha}}{\log N/(c_1\epsilon)}\right)\frac{\log N}{c_1\epsilon}+\frac{\log\left(\frac{c_2}{\epsilon^7}\right)}{c_1\epsilon}\right]+\eta \\
&\leq \sum_{t=S}^{NS-\eta}N^{1-\frac{S-t/N-\frac{\log\left(\frac{c_2}{\epsilon^7}\right)}{c_1\epsilon}}{\log N/(c_1\epsilon)}}+\eta \\
&\leq N^{1-\frac{c_1\epsilon}{\log N}S+\frac{\log\left(\frac{c_2}{\epsilon^7}\right)}{\log N}}\int_{t=S}^{NS-\eta+1}N^{\frac{c_1\epsilon}{N\log N}t}dt+\eta \\
&= N^{1-\frac{c_1\epsilon}{\log N}S+\frac{\log\left(\frac{c_2}{\epsilon^7}\right)}{\log N}}\left[\frac{N^{\frac{c_1\epsilon}{N\log N}(NS-\eta+1)}-N^{\frac{c_1\epsilon}{N\log N}S}}{\frac{c_1\epsilon}{N\log N}\log N}\right]+\eta \\
&\leq \frac{N}{c_1\epsilon}N^{1+\frac{\log\left(\frac{c_2}{\epsilon^7}\right)}{\log N}-\frac{c_1\epsilon}{N\log N}\eta+\frac{c_1\epsilon}{N\log N}}+\eta \\
&\leq \frac{N}{c_1\epsilon}N^{\frac{N\log N+N\log\left(\frac{c_2}{\epsilon^7}\right)-2N\log\left(\frac{c_2N}{\epsilon^7}\right)+2c_1\epsilon-3c_1\epsilon+c_1\epsilon}{N\log N}}+\eta \\
&= \frac{N}{c_1\epsilon}N^{\frac{N\log N-N\log\left(\frac{c_2}{\epsilon^7}\right)}{N\log N}}+\eta \\
&\leq \frac{N}{c_1\epsilon}+\eta
\end{aligned}$$

$$= \frac{N}{c_1\epsilon} + 2\left\lceil \frac{N \log \frac{c_2 N}{\epsilon^7}}{c_1 \epsilon} \right\rceil + 3$$
$$= \Theta\left(\frac{N}{q} \log \frac{N}{q}\right).$$

The first inequality follows from bounding probabilities by 1, and the second inequality follows from (1.18).

Note that we need to make sure that $\theta_q < NS$ in order to make the lower bound nontrivial. For any $\epsilon \in (0,1)$, if $q > \frac{1}{S^{1-\epsilon}}$, then one can verify that $\frac{N}{q} \log \frac{N}{q} < NS^{1-\epsilon} \log(NS^{1-\epsilon}) < NS$ for S sufficiently large. Therefore, as long as $q > \Omega\left(\frac{1}{S^{1-\epsilon}}\right)$, $\theta_q < NS$ holds. □

Proof of Lemma 1.3.4. In this proof we again use the connection of our inventory allocation problem to the balls-into-bins problem stated in Peres et al. (2010). Specifically, the allocation process of 2-opaque product can be characterized by distribution vector $\mathbf{p}$ with $p_i = \frac{1-q}{N} + q\frac{2(i-1)}{N(N-1)}$. The second term in p_i represents the probability that an opaque customer is allocated the i-th lowest inventory product (under the balancing policy), which only happens when the customer chooses the i-th least inventory product along with another product with lower on-hand inventory.

Let $\epsilon = \frac{q}{12}$, then $\mathbf{p}$ satisfies the condition in Lemma 1.7.2 where $p_{N/3} \leq \frac{1-4\epsilon}{N}$ and $p_{2N/3+1} \geq \frac{1+4\epsilon}{N}$. Therefore, we can apply the same analysis in the proof of Lemma 1.3.3 to the 2-opaque product and get $\bar{\theta}_q = NS - \mathbb{E}[R^{\mathcal{M}}(q,2)] = O\left(\frac{N}{q} \log \frac{N}{q}\right)$. The lower bound on $\bar{\theta}_q$ is trivial since the largest possible value for $R^{\mathcal{M}}(q,2)$ is $N(S-1)+1$. □

Lemma 1.7.3. $\mathbb{E}[R(0)^2] \leq \Theta\left(N^3 S^2 e^{-S/6}\right) + \left(\frac{3NS}{2} - N + 1\right)\mathbb{E}[R(0)] - \frac{NS}{2}((N(S-1)+1)$.

Proof. In this proof we focus on an upper bound of $\mathbb{E}[R(0)^2]$. The random variable $R(0)$

can take any integer value between $[S, N(S-1)+1]$. However, it is quite unlikely for $R(0)$ to be close to S. In order to get a good upper bound on the variance, first we show that $R(0) \in [NS/2, N(S-1)+1]$ with probability $1 - Ne^{-S/6}$.

Let $Y_1(m), \ldots, Y_N(m)$ be the random variable that counts the number of customers who purchase each product when the total number of customers is m, and the initial inventory is $\vec{S}$. This count continues even if replenisments occur before m customers arrive. Under the no opaque selling strategy, $Y_1(m), \ldots, Y_N(m)$ follows a multinomial distribution with probability $\frac{1}{N}$ for each outcome. The marginal distribution of $Y_i(m)$ is binomial, $B(m, \frac{1}{N})$. When $R(0) < NS/2$, then there must be at least one product that was chosen S times in the first $\frac{NS}{2}$ arrivals. This implies that there is an i such that $Y_i(\frac{NS}{2}) \geq S$. Therefore we get the following inequality,

$$\begin{aligned}
\mathbb{P}[R(0) < \frac{NS}{2}] \leq & \mathbb{P}\left[\cup_{i=1}^{N} Y_i\left(\frac{NS}{2}\right) \geq S\right] \\
\leq & N\mathbb{P}\left[B\left(\frac{NS}{2}, \frac{1}{N}\right) \geq (1+1)\frac{S}{2}\right] \\
\leq & Ne^{-\frac{S}{6}}, \qquad (1.20)
\end{aligned}$$

where the second inequality follows from the union bound and the last inequality follows immediately from the Chernoff bound, $\mathbb{P}[B(m, 1/N) \geq (1+\delta)\frac{m}{N}] \leq e^{-\frac{\delta^2 m}{3N}}$.

Now we have that

$$\begin{aligned}
\mathbb{E}[R(0)^2] &= \sum_{i<NS/2} i^2 \mathbb{P}[R(0)=i] + \sum_{i=NS/2}^{N(S-1)+1} i^2 \mathbb{P}[R(0)=i] \\
&\leq \left(\frac{NS}{2}\right)^2 \mathbb{P}\left[R(0) < \frac{NS}{2}\right] + \sum_{i=NS/2}^{N(S-1)+1} i^2 \mathbb{P}[R(0)=i]
\end{aligned}$$

$$
\begin{aligned}
\leq & \Theta(N^3 e^{-S/6} S^2) + \sum_{i=NS/2}^{N(S-1)+1} \left[(N(S-1)+1+\frac{NS}{2})i - (N(S-1)+1)\frac{NS}{2} \right] \mathbb{P}[R(0)=i] \\
= & \Theta(N^3 e^{-S/6} S^2) + \left(N(S-1)+1+\frac{NS}{2} \right) \sum_{i=NS/2}^{N(S-1)+1} i\mathbb{P}[R(0)=i] \\
& - (N(S-1)+1)\frac{NS}{2} \sum_{i=NS/2}^{N(S-1)+1} \mathbb{P}[R(0)=i] \\
\leq & \Theta(N^3 e^{-S/6} S^2) + \left(N(S-1)+1+\frac{NS}{2} \right) \mathbb{E}[R(0)] - (N(S-1)+1)\frac{NS}{2} \mathbb{P}[R(0) \geq NS/2] \\
\leq & \Theta(N^3 e^{-S/6} S^2) + \left(N(S-1)+1+\frac{NS}{2} \right) \mathbb{E}[R(0)] - (N(S-1)+1)\frac{NS}{2} \left(1 - Ne^{-S/6} \right) \\
= & \Theta(N^3 e^{-S/6} S^2) + \left(N(S-1)+1+\frac{NS}{2} \right) \mathbb{E}[R(0)] - (N(S-1)+1)\frac{NS}{2}. \qquad (1.21)
\end{aligned}
$$

The first inequality follows from (1.20) and the second inequality follows from the fact that for $i \in [a,b]$, $i^2 \leq (a+b)i - ab$. The third inequality follows from the fact that $\sum_{i=NS/2}^{N(S-1)+1} i\mathbb{P}[R(0)=i] < \sum_{i=S}^{N(S-1)+1} i\mathbb{P}[R(0)=i] = \mathbb{E}[R(0)]$. The fourth inequality follows from (1.20) again. □

Proof of Theorem 1.3.1. In this proof, we first compute the lower bound on the relative holding cost savings and ordering cost savings separately, then combine them together to get a lower bound on the relative total cost savings.

For N-opaque products, the savings in long run average ordering cost is

$$
\begin{aligned}
\frac{\mathcal{K}(0) - \mathcal{K}^{\mathcal{M}}(q,N)}{\mathcal{K}(0)} = & \frac{K/\mathbb{E}[R(0)] - K/\mathbb{E}[R^{\mathcal{M}}(q,N)]}{K/\mathbb{E}[R(0)]} \\
= & 1 - \frac{\mathbb{E}[R(0)]}{\mathbb{E}[R^{\mathcal{M}}(q,N)]} \\
= & 1 - \frac{NS - \theta_S}{NS - \theta_q} \\
= & \frac{\theta_S - \theta_q}{NS - \theta_q}. \qquad (1.22)
\end{aligned}
$$

The first equality follows from the cost formula in Eq. (1.1). Directly applying results in Lemma 1.3.2 and 1.3.3, we can get

$$\frac{\mathcal{K}(0) - \mathcal{K}^{\mathcal{M}}(q,N)}{\mathcal{K}(0)} = \frac{\Omega\left(N\sqrt{S\log N}\right)}{NS} = \Omega\left(\sqrt{\frac{\log N}{S}}\right).$$

Note that for this equality to hold, we need to make sure that θ_q can not exceed θ_S. For any $\epsilon > 0$, if $q = \Omega\left(\sqrt{\frac{\log N}{S^{1-\epsilon}}}\right)$, then we can show that $\frac{N}{q}\log\frac{N}{q} = o(N\sqrt{S\log N})$ and thus the last equality holds.

Next we derive a lower bound on the savings in long run average holding cost. We will first provide an upper bound on $\mathcal{H}^{\mathcal{M}}(q,N)$.

$$\begin{aligned}
\mathcal{H}^{\mathcal{M}}(q,N) =& \frac{(2NS+1)\mathbb{E}[R^{\mathcal{M}}(q,N)] - \mathbb{E}[R^{\mathcal{M}}(q,N)^2]}{2\lambda\mathbb{E}[R^{\mathcal{M}}(q,N)]}h. \\
\leq& \frac{(2NS+1)\mathbb{E}[R^{\mathcal{M}}(q,N)] - \mathbb{E}[R^{\mathcal{M}}(q,N)]^2}{2\lambda\mathbb{E}[R^{\mathcal{M}}(q,N)]}h \\
=& \frac{2NS+1-\mathbb{E}[R^{\mathcal{M}}(q,N)]}{2\lambda}h \\
=& \frac{2NS+1-NS+\theta_q}{2\lambda}h \\
=& \frac{h}{2\lambda}\left(NS+\theta_q+1\right). \qquad (1.23)
\end{aligned}$$

The first equality follows from Eq. (1.2) and the first inequality follows from the Jensen's Inequality.

We then derive a lower bound on $\mathcal{H}(0)$.

$$\mathcal{H}(0) = \frac{(2NS+1)\mathbb{E}[R(0)] - \mathbb{E}[R(0)^2]}{2\lambda\mathbb{E}[R(0)]}h.$$

$$
\begin{aligned}
\geq & h\frac{1}{2\lambda\mathbb{E}[R(0)]}\Bigg((2NS+1)\mathbb{E}[R(0)]-\Theta(N^3e^{-S/6}S^2) \\
& +(N(S-1)+1)NS/2-(N(S-1)+1+NS/2)\mathbb{E}[R(0)]\Bigg) \\
= & h\frac{NS/2+N}{2\lambda}+h\frac{(N(S-1)+1)NS/2-\Theta\left(N^3e^{-S/6}S^2\right)}{2\lambda\mathbb{E}[R(0)]} \\
= & h\frac{NS/2+N}{2\lambda}+h\frac{(N(S-1)+1)NS/2-\Theta\left(N^3e^{-S/6}S^2\right)}{2\lambda(NS-\theta_S)} \\
= & h\frac{(NS)^2-\frac{NS}{2}\theta_S+\frac{N(N+1)}{2}S-N\theta_S-\Theta\left(N^3e^{-S/6}S^2\right)}{2\lambda(NS-\theta_S)} \\
= & h\frac{(NS)^2-\frac{NS}{2}\theta_S+O(N^2S)}{2\lambda(NS-\theta_S)}.
\end{aligned}
\tag{1.24}
$$

The first inequality follows from the upper bound on $\mathbb{E}[R(0)^2]$ in Lemma 1.7.3.

Then using the bound on holding cost, we derive lower bound on the saving in holding cost.

$$
\begin{aligned}
\frac{\mathcal{H}(0)-\mathcal{H}^{\mathcal{M}}(q,N)}{\mathcal{H}(0)} = & 1-\frac{\mathcal{H}^{\mathcal{M}}(q,N)}{\mathcal{H}^{\mathcal{M}}(0)} \\
\geq & 1-\frac{\frac{h}{2\lambda}\left(NS+\theta_q+1\right)}{h\frac{(NS)^2-\frac{NS}{2}\theta_S+O(N^2S)}{2\lambda(NS-\theta_S)}} \\
= & \frac{(NS)^2-\frac{NS}{2}\theta_S+O(N^2S)-\left(NS-\theta_S\right)\left(NS+\theta_q\right)}{(NS)^2-\frac{NS}{2}\theta_S+O(N^2S)} \\
= & \frac{\frac{NS}{2}\theta_S+O\left(N^2S\right)-\theta_qO\left(NS\right)}{N^2S^2-O\left(N^2S^{\frac{3}{2}}\sqrt{\log N}\right)}.
\end{aligned}
\tag{1.25}
$$

The first inequality follows (1.23) and (1.24), where we plug in the lower bound for $\mathcal{H}(0)$ and the upper bound for $\mathcal{H}^{\mathcal{M}}(q,N)$. Then directly plugging results in Lemma 1.3.2 and 1.3.3

into Eq. (1.25), we can get

$$\frac{\mathcal{H}(0)-\mathcal{H}^{\mathcal{M}}(q,N)}{\mathcal{H}(0)} \geq \frac{\frac{NS}{2}\Omega(N\sqrt{S\log N})}{N^2S^2} = \Omega\left(\sqrt{\frac{\log N}{S}}\right).$$

Note that for this equality to hold, we need to make sure that θ_q does not exceed $\Theta(N\sqrt{S\log N})$. We know that for any $\epsilon > 0$, if $q = \Omega\left(\sqrt{\frac{\log N}{S^{1-\epsilon}}}\right)$, then $\frac{N}{q}\log\frac{N}{q} = o(N\sqrt{S\log N})$ and thus the last equality holds.

Since the lower bound on the savings for holding costs and ordering costs are the same, we can conclude that

$$\frac{\mathcal{H}(0)+\mathcal{K}(0)-\mathcal{H}^{\mathcal{M}}(q,N)-\mathcal{K}^{\mathcal{M}}(q,N)}{\mathcal{H}(0)+\mathcal{K}(0)} = \Omega\left(\sqrt{\frac{\log N}{S}}\right).$$

The proof for the $k = 2$ case is essentially the same due to the similarity of the results in Lemma 1.3.3 and 1.3.4. In all the steps of the proof, we only need to substitute θ_q with $\bar{\theta}_q$, which leads to the same order of magnitude as follows.

$$\frac{\mathcal{H}(0)+\mathcal{K}(0)-\mathcal{H}^{\mathcal{M}}(q,2)-\mathcal{K}^{\mathcal{M}}(q,2)}{\mathcal{H}(0)+\mathcal{K}(0)} = \Omega\left(\sqrt{\frac{\log N}{S}}\right).$$

For any $2 \leq k \leq N$, the relative cost savings will be bounded between the $k = 2$ case and the $k = N$ case, with the same q value, which leads to the result in Theorem 1.3.1. □

Proof of Theorem 1.3.2. The goal is to show that both $\frac{\mathcal{K}(0)-\mathcal{K}^{\mathcal{M}}(q,k)}{\mathcal{K}(0)}$ and $\frac{\mathcal{H}(0)-\mathcal{H}^{\mathcal{M}}(q,k)}{\mathcal{H}(0)}$ are on the order of $\frac{\theta_S}{NS}$. For any $k \geq 2$ and $q = \Omega\left(\sqrt{\frac{\log N}{S^{1-\epsilon}}}\right)$, we have the following bounds from

the proof of Theorem 1.3.1 (Eq. (1.22) and (1.25)).

$$\frac{\mathcal{K}(0) - \mathcal{K}^{\mathcal{M}}(q,k)}{\mathcal{K}(0)} = \Theta\left(\frac{\theta_S}{NS}\right)$$
$$\frac{\mathcal{H}(0) - \mathcal{H}^{\mathcal{M}}(q,k)}{\mathcal{H}(0)} \geq \Theta\left(\frac{\theta_S}{NS}\right).$$

Thus, in order to conclude that both cost savings are equal to $\Theta\left(\frac{\theta_S}{NS}\right)$, we only need to upper bound $\frac{\mathcal{H}(0)-\mathcal{H}^{\mathcal{M}}(q,N)}{\mathcal{H}(0)}$ by the same order of magnitude.

Note that the maximum possible savings in holding costs is achieved in the fully flexible scheme with an allocation policy that minimizes the long run holding costs. Next, we will show that in the fully flexible scheme $(1, N)$, the balancing policy indeed minimizes long run holding costs, and then provide an upper bound on the relative holding cost savings.

The long run holding costs for the fully flexible scheme is

$$\mathcal{H}^{\pi}(1,N) = \frac{(2NS+1)\mathbb{E}[R^{\pi}(1,N)] - \mathbb{E}[R^{\pi}(1,N)^2]}{2\lambda\mathbb{E}[R^{\pi}(1,N)]}h.$$

To show that the balancing policy $\mathcal{M}$ minimizes the long run holding costs, we only need to prove that it maximizes $\frac{\mathbb{E}[R^{\pi}(1,N)^2]}{\mathbb{E}[R^{\pi}(1,N)]}$. Note that the random variable $R^{\pi}(1,N)$ is non-negative and upper bounded by $N(S-1)+1$, we can derive a trivial upper bound which is $\frac{\mathbb{E}[R^{\pi}(1,N)^2]}{\mathbb{E}[R^{\pi}(1,N)]} \leq N(S-1)+1$. Also note that when the retailer applies the balancing policy $\mathcal{M}$, $R^{\mathcal{M}}(1,N) = N(S-1)+1$ with probability 1. Therefore, $\frac{\mathbb{E}[R^{\mathcal{M}}(1,N)^2]}{\mathbb{E}[R^{\mathcal{M}}(1,N)]} = N(S-1)+1$, which achieves the maximum possible value of $\frac{\mathbb{E}[R^{\pi}(1,N)^2]}{\mathbb{E}[R^{\pi}(1,N)]}$.

Therefore, we can upper bound the relative holding cost savings:

$$\begin{aligned}
\frac{\mathcal{H}(0)-\mathcal{H}^{\mathcal{M}}(q,k)}{\mathcal{H}(0)} \leq & \frac{\mathcal{H}(0)-\mathcal{H}^{\mathcal{M}}(1,N)}{\mathcal{H}(0)} \\
= & 1-\frac{\mathcal{H}^{\mathcal{M}}(1,N)}{\mathcal{H}^{\mathcal{M}}(0)} \\
\leq & 1-\frac{\frac{(2NS+1)\mathbb{E}[R^{\mathcal{M}}(1,N)]-\mathbb{E}[R^{\mathcal{M}}(1,N)^2]}{2\lambda\mathbb{E}[R^{\mathcal{M}}(1,N)]}h}{\frac{2NS+1-\mathbb{E}[R(0)]}{2\lambda}h} \\
= & 1-\frac{NS+N}{NS+\theta_S+1} \\
= & \frac{\theta_S-N+1}{NS+\theta_S+1} \\
= & \Theta\left(\frac{\theta_S}{NS}\right). \qquad (1.26)
\end{aligned}$$

The first equality follows from the cost formula in Eq. (1.2) and the second inequality follows from Eq. (1.3).

To sum up, we have $\frac{\mathcal{K}(0)-\mathcal{K}^{\mathcal{M}}(q,k)}{\mathcal{K}(0)}=\Theta\left(\frac{\theta_S}{NS}\right)$ and $\frac{\mathcal{H}(0)-\mathcal{H}^{\mathcal{M}}(q,k)}{\mathcal{H}(0)}=\Theta\left(\frac{\theta_S}{NS}\right)$ for any $k\geq 2$ and $q=\Omega\left(\sqrt{\frac{\log N}{S^{1-\epsilon}}}\right)$. This means that both the relative savings in ordering costs and holding costs are on the same order. □

Proof of Theorem 1.3.3. The proof of Theorem 1.3.3 utilizes the intermediate results from the proof of Theorem 1.3.1 and 1.3.2. We have shown that the relative total cost savings for a limited flexible scheme $(q,2)$ and the relative holding cost savings for the fully flexible scheme $(1,N)$ are on the same order, which is $\Theta\left(\frac{\theta_S}{NS}\right)$. The ordering cost savings for the fully flexible scheme is also on the same order.

$$\frac{\mathcal{K}(0)-\mathcal{K}^{\mathcal{M}}(1,N)}{\mathcal{K}(0)} = \frac{K/\mathbb{E}[R(0)]-K/\mathbb{E}[R^{\mathcal{M}}(1,N)]}{K/\mathbb{E}[R(0)]}$$

$$
\begin{aligned}
=&1-\frac{\mathbb{E}[R(0)]}{\mathbb{E}[R^{\mathcal{M}}(1,N)]}\\
=&1-\frac{NS-\theta_S}{N(S-1)+1}\\
=&\frac{\theta_S-N+1}{NS-N+1}\\
=&\Theta\left(\frac{\theta_S}{NS}\right). \qquad (1.27)
\end{aligned}
$$

Therefore, considering the relative savings in total costs, the order of magnitude is still $\Theta\left(\frac{\theta_S}{NS}\right)$, either for a minimal degree of opacity $(q,2)$, or the fully flexible scheme $(1,N)$. $\square$

Proof of Lemma 1.4.1. First, we provide an upper bound for γ^*. Using the formula of long run average cost per unit sold in (1.1) and (1.2), we have

$$
\begin{aligned}
\gamma^* &= h\frac{(2NS+1)\mathbb{E}[R^{\pi^*}(q,N)]-\mathbb{E}[(R^{\pi^*}(q,N))^2]}{2\lambda\mathbb{E}[R^{\pi^*}(q,N)]}+\frac{K}{\mathbb{E}[R^{\pi^*}(q,N)]}\\
&\le h\frac{(2NS+1)\mathbb{E}[R^{\mathcal{M}}(q,N)]-\mathbb{E}[R^{\mathcal{M}}(q,N)^2]}{2\lambda\mathbb{E}[R^{\mathcal{M}}(q,N)]}+\frac{K}{\mathbb{E}[R^{\mathcal{M}}(q,N)]}\\
&\le \frac{h}{2\lambda}(2NS+1-\mathbb{E}[R^{\mathcal{M}}(q,N)])+\frac{K}{\mathbb{E}[R^{\mathcal{M}}(q,N)]}\\
&= \frac{h}{2\lambda}\left(NS+\theta_q\right)+\frac{K}{NS-\theta_q}.
\end{aligned}
$$

The first inequality follows from the fact that γ^* is a lower bound on the cost of any feasible policy, including $\mathcal{M}$. The second inequality follows from Jensen's inequality. The last equality follows from Lemma 1.3.3.

Next, we provide a lower bound for γ^*. The largest possible value of $R^{\pi}(q,N)$ is achieved when the system enters state $(1,\ldots,1)$, and after serving one more customer, the replenishment is triggered. Thus, $R^{\pi}(q,N)\le N(S-1)+1$ with probability 1, no matter what policy

is used. Therefore, $R^{\pi^*}(q,N)$ is at most $N(S-1)+1$ with probability 1. Therefore,

$$\begin{aligned}
\gamma^* &= h\frac{(2NS+1)\mathbb{E}[R^{\pi^*}(q,N)] - \mathbb{E}[(R^{\pi^*}(q,N))^2]}{2\lambda\mathbb{E}[R^{\pi^*}(q,N)]} + \frac{K}{\mathbb{E}[R^{\pi^*}(q,N)]} \\
&\geq h\frac{(2NS+1)\mathbb{E}[R^{\pi^*}(q,N)] - (N(S-1)+1)\mathbb{E}[R^{\pi^*}(q,N)]}{2\lambda\mathbb{E}[R^{\pi^*}(q,N)]} + \frac{K}{N(S-1)+1} \\
&\geq \frac{h}{2\lambda}(NS+N) + \frac{K}{N(S-1)+1} \\
&\geq \frac{h}{2\lambda}(NS+N) + \frac{K}{NS},
\end{aligned}$$

where the first inequality follows from the previously derived upper bound on $R^{\pi}(q)$. □

Proof of Theorem 1.4.1. (a) Let x and y be two vectors with length N. We say x majorizes y, or $x \succ y$, if after sorting x and y in ascending order, then $\sum_{i=j}^{N} x_i \geq \sum_{i=j}^{N} y_i$ for $j = 1, \cdots, N$ and $\sum_{i=1}^{N} x_i = \sum_{i=1}^{N} y_i$. We say an inventory state x is in level k if $\sum_{i=1}^{N} x_i = k$. Recall that $J(x)$ is the relative cost-to-go under the optimal policy until the next replenishment, beginning at inventory state x.

First, we inductively show a hypothesis that for any two states x and y in any level at most $\lfloor \frac{NS}{2} \rfloor$ such that $x \succ y$, we have that $0 > J(x) \geq J(y)$. Without loss of generality, we assume that inventory states x, y are sorted in ascending order. The lowest possible inventory level is N with only one possible inventory state, which is $(1, \cdots, 1)$. From Eq. (1.5), we have that $J(1, \ldots, 1) = \frac{h}{\lambda}N - \gamma^*$. From Lemma 1.4.1, we know that $\gamma^* > \frac{h}{2\lambda}NS$. Hence, we have $J(1, \ldots, 1) < 0$. This completes the base case.

Now suppose that the inductive hypothesis holds for any inventory level strictly less than k. Now consider any two states $x \succ y$ in level k. There are 2 possible cases. Define i_x to be 0 if $x_1 \geq 2$ and $i_x := \max\{j : x_j = 1\}$ otherwise. Define i_y analogously to i_x. Since $x \succ y$,

then $i_x \geq i_y$. From our dynamic programming formulation we have that

$$\begin{aligned}
J(x) &= \frac{hk}{\lambda} - \gamma^* + \frac{1-q}{N} i_x J(\vec{S}) + \frac{1-q}{N} \sum_{i=i_x+1}^{N} J(x-e_i) + q \min\{0, J(x-e_{i_x+1}), \ldots, J(x-e_N)\} \\
&= \frac{hk}{\lambda} - \gamma^* + \frac{1-q}{N} i_x J(\vec{S}) + \frac{1-q}{N} \sum_{i=i_x+1}^{N} J(x-e_i) + q J(x-e_N) \\
&\geq \frac{hk}{\lambda} - \gamma^* + \frac{1-q}{N} i_y J(\vec{S}) + \frac{1-q}{N} \sum_{i=i_y+1}^{N} J(y-e_i) + q J(y-e_N) \\
&= \frac{hk}{\lambda} - \gamma^* + \frac{1-q}{N} i_y J(\vec{S}) + \frac{1-q}{N} \sum_{i=i_y+1}^{N} J(y-e_i) + q \min\{0, J(y-e_{i_y+1}), \ldots, J(y-e_N))\} \\
&= J(y).
\end{aligned}$$

The first equation follows Eq. (1.5) since (i) $x \neq \vec{S}$ implying $g(x) = \frac{hk}{\lambda}$, (ii) $J(\vec{S}) = 0$ by definition, and (ii) $J(x-e_{i_x+1}), \ldots, J(x-e_N)$ are all less than 0 by the inductive hypothesis. The second equation follows from the fact that $J(x-e_{i_x+1}) \geq \ldots \geq J(x-e_N)$ since $x - e_{i_x+1} \succ \ldots \succ x - e_N$ from the inductive hypothesis. The inequality follows from the facts that (i) $i_x \geq i_y$, (ii) $J(\vec{S}) = 0$ by definition, (iii) $J(x-e_i) \geq J(y-e_i)$ when $x_i > 1$ since $x - e_i \succ y - e_i$ by the inductive hypothesis, and (iv) $J(y-e_{i_y+1}), \ldots, J(y-e_N)$ are all less than 0 by the inductive hypothesis.

From Lemma 1.4.1, we know that $\gamma^* > \frac{h}{2\lambda} NS$, which implies $J(x) < 0$ since it can be expressed as the sum of non-positive terms. This completes the induction proof. The proof of the result now follows immediately since for any state x with level at most $\lfloor \frac{NS}{2} \rfloor$, the cost of $J(x-e_N) < 0$ and less than $J(x-e_i)$ for any i such that $x_i > 1$. This implies that the balancing action is optimal when the level is at most $\lfloor \frac{NS}{2} \rfloor$.

(b) Without loss of generality, we assume the state x is sorted in ascending order, and

thus $x_1 = 1$. State $\vec{S}$ is a possible state to enter from x if the next customer purchases product 1. We know that $J(\vec{S}) = 0$. Thus, if we show that $J(x - e_N) > 0 = J(\vec{S})$, then the balancing policy is suboptimal compared to choosing product 1 by Eq. (1.5). Next, we provide a sufficient condition for $J(x - e_N)$ to be greater than 0.

From Eq. (1.6), we can write $J(x - e_N)$ as

$$J(x - e_N) = \mathbb{E}\left[R^{\pi^*}(q, N; x - e_N)\right]\left[\frac{h}{2\lambda}\left[\left(2\left(\sum_{i=1}^{N} x_i - 1\right) + 1\right) - \frac{\mathbb{E}[R^{\pi^*}(q, N; x - e_N)^2]}{\mathbb{E}[R^{\pi^*}(q, N; x - e_N)]}\right] - \gamma^*\right]$$

Since $\mathbb{E}\left[R^{\pi^*}(q, N; x - e_N)\right]$ is always positive, in order to show $J(x - e_N) > 0$, we only need to show

$$\frac{h}{2\lambda}\left[\left(2\left(\sum_{i=1}^{N} x_i - 1\right) + 1\right) - \frac{\mathbb{E}[R^{\pi^*}(q, N; x - e_N)^2]}{\mathbb{E}[R^{\pi^*}(q, N; x - e_N)]}\right] - \gamma^* > 0,$$

which is equivalent to

$$\sum_{i=1}^{N} x_i > \frac{\lambda}{h}\gamma^* + \frac{1}{2} + \frac{\mathbb{E}[R^{\pi^*}(q, N; x - e_N)^2]}{2\mathbb{E}[R^{\pi^*}(q, N; x - e_N)]}.$$

To complete the proof, we upper bound the RHS of the above inequality to give a looser sufficient condition on $\sum_{i=1}^{N} x_i$. Specifically, observe that upper bounding γ^* using Lemma 1.4.1 and upper bounding $\frac{\mathbb{E}[R^{\pi^*}(q,N;x-e_N)^2]}{2\mathbb{E}[R^{\pi^*}(q,N;x-e_N)]}$ by $\frac{2N^2-(1-q)N}{2(1-q)^2}$ yields the final result.

Thus, all that remains is to show that $\frac{\mathbb{E}[R^{\pi^*}(q,N;x-e_N)^2]}{2\mathbb{E}[R^{\pi^*}(q,N;x-e_N)]}$ can be upper bounded by $\frac{2N^2-(1-q)N}{2(1-q)^2}$. First, we lower bound $\mathbb{E}[R^{\pi^*}(q, N; x - e_N)]$ by 1. We upper bound the second moment by considering a state $y = (1, \infty, \ldots, \infty)$, and observe that $R^{\pi}(q, N; y)$ stochastically dominates $R^{\pi^*}(q, N; x - e_N)$ for any policy π. Next, we observe that the Geometric random

variable counting for the first success occurrence with success probability $\frac{1-q}{N}$ stochastically dominates $R^{\pi}(q,N;y)$. Thus, $\mathbb{E}[R^{\pi^*}(q,N;x-e_N)^2] \leq E[Geo(\frac{1-q}{N})^2] = \frac{2N^2-(1-q)N}{(1-q)^2}$, which completes the proof. $\square$

Proof of Theorem 1.4.2. Let $\gamma^{\mathcal{M}}$ denote the long run average cost under the balancing policy. We want to show $\frac{\gamma^{\mathcal{M}}}{\gamma^*} \leq 1 + \Theta\left(\frac{N}{S}\right)$, which is derived as

$$\begin{aligned}
\frac{\gamma^{\mathcal{M}}}{\gamma^*} &= \frac{\frac{K}{\mathbb{E}[R^{\mathcal{M}}(q,N)]} + \frac{h((2NS+1)\mathbb{E}[R^{\mathcal{M}}(q,N)]-\mathbb{E}[R^{\mathcal{M}}(q,N)^2])}{2\lambda\mathbb{E}[R^{\mathcal{M}}(q,N)]}}{\gamma^*} \\
&\leq \frac{\frac{K}{\mathbb{E}[R^{\mathcal{M}}(q,N)]} + \frac{h(2NS+1-\mathbb{E}[R^{\mathcal{M}}(q,N)])}{2\lambda}}{\frac{h}{2\lambda}(NS+N)+\frac{K}{NS}} \\
&\leq \max\left\{\frac{\frac{K}{\mathbb{E}[R^{\mathcal{M}}(q,N)]}}{\frac{K}{NS}}, \frac{\frac{h(2NS+1-\mathbb{E}[R^{\mathcal{M}}(q,N)])}{2\lambda}}{\frac{h}{2\lambda}(NS+N)}\right\} \\
&= \max\left\{\frac{\frac{K}{NS-\theta_q}}{\frac{K}{NS}}, \frac{NS+1+\theta_q}{NS+N}\right\} \\
&\leq \max\left\{1+\frac{\theta_q}{NS-\theta_q}, 1+\frac{\theta_q}{NS+1}\right\} \\
&\leq 1+\frac{\theta_q}{NS-\theta_q}.
\end{aligned}$$

The first equality follows from the exact formula for the long run average cost $\gamma^{\mathcal{M}}$ in Eqs. (1.1) and (1.2). The first inequality follows from Jensen's inequality and plugging in lower bound of γ^* in Lemma 1.4.1. The second inequality follows from the fact that $\frac{a_1+b_1}{a_2+b_2} \leq \max\left\{\frac{a_1}{b_1}, \frac{a_2}{b_2}\right\}$ for any positive $a_1, a_2, b_1, b_2 \in \mathbb{R}$. In the third inequality, we plug in the lower bound of $\mathbb{E}[R^{\mathcal{M}}(q)]$ in Lemma 1.3.3. $\square$

1.8 Distinctions between Consumer Flexibility and Existing Literature

In this section, we compare the role of consumer flexibility in opaque selling with other notion of flexibility in the literature. Specifically, we consider process flexibility (Jordan and Graves (1995)), flexible online resource allocation (Asadpour et al. (2019)), and flexible queueing systems (Tsitsiklis and Xu (2012), Tsitsiklis and Xu (2017)). Our goal is to demonstrate that the demand-side flexibility can not be captured by existing models of supply-side flexibility.

Jordan and Graves (1995) demonstrates that even just a little flexibility, if configured in the right way, can be extremely effective in coping with demand uncertainty. This stream of work, also known as process flexibility for manufacturers, consider an offline decision making problem where the demands are realized all at once in each period. In online retail demands arrive sequentially, and thus our model necessarily considers dynamic allocation decisions. Moreover, the manufacturer in process flexibility has full control in allocating production capacity to demands, since the demand itself does not have a preference of where it would be fulfilled from. In contrast, the retailer in our model can only allocate a product to an opaque customer within their selected k options.

Asadpour et al. (2019) consider an online resource allocation problem for multiple resource types, each with a fixed initial inventory, and no replenishment is allowed. The performance measure is the expected total number of lost sales after a finite time horizon. Asadpour et al. (2019) show that the long chain design in this setting can effectively balance the depletion of the inventory of different resources. In their setting, all requests are flexible,

i.e., it is indifferent for the request to be served by either the primary resource or a secondary resource. This is similar to the case $q = 1$ in our model, and with an opaque product design configured as a long chain. However, for general $q < 1$ and k-opaque products, the benefit cannot be captured by Asadpour et al. (2019). Despite that both models consider the benefit from inventory balancing in an online fashion, it is obvious that the performance metric and inventory replenishment policy we use are quite different than the online resource allocation problem. Xu et al. (2020) can be thought as an extension to Asadpour et al. (2019) to more general settings. Shi et al. (2019) illustrates that the concept of chaining can be extended to much more general settings in the multi period MTO environment when demand can be backlogged.

In flexible queueing systems, there is a multi-server queueing system where each server is capable of serving some demand types. Incoming requests are assigned to servers according to a predetermined allocation policy. The performance metric is the expected waiting time or the average queue length. Tsitsiklis and Xu (2012) considers n demand types and n dedicated servers for each demand type with service rate $1 - p$, and one fully flexible server with service rate np. We can flip the supply and demand in our model and try to fit into the flexible queueing framework, and ultimately show this is not possible. Consider the customers in our model as the servers, and the interarrival time of customers as a service rate. With $q = p$, the N-opaque customers can be viewed as the fully flexible server. The inventory in our setting becomes the demand and forms n queues at each dedicated server. Our inventory balancing policy can be interpreted as the following allocation rule: when a fully flexible server becomes idle, the first unit of inventory in the queue with the longest length is allocated to the fully flexible server. The holding cost is analogous to the

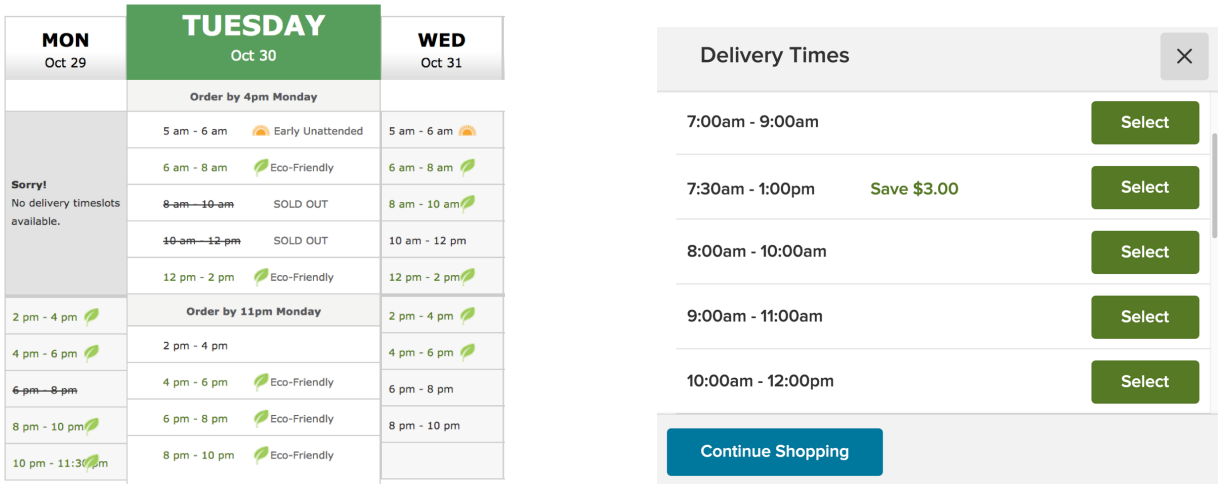
average waiting time. When a dedicated queue has zero length, an inventory replenishemnt is triggered, and a batch of inventory arrives to the queueing system where the arrival size depends on the inventory on-hand for each queue. The idea of inventory replenishment fails to fit into this queueing model since the arrival of inventory occurs in batches. In fact, the batch arrival has state dependent size and state dependent interarrival time, which cannot be handled by the assumptions in Tsitsiklis and Xu (2012) and related papers.

To summarize, the literature in flexibility cannot address many aspects of the model we use to study consumer flexibility, and thus a new modeling and analysis framework is needed.

Chapter 2

The Value of Consumer Flexibility in Scheduled Service Systems

In this paper, we study the benefit of leveraging consumer flexibility into scheduled service systems with time window choices. Consumer flexibility is the fundamental principle that instead of one specific option, consumers may be willing to receive one of multiple options in exchange for a reward. In the context of online booking systems for scheduled services, a customer makes an appointment by choosing one option from a list of regular time windows and consumer flexibility can be realized through large time windows, which are composed of multiple regular time windows. Service providers can benefit from capacity pooling and our goal is to understand how to design large time windows and how large this impact can be. The main insight is that just creating large time windows that is composed of two consecutive regular time windows and with just a small proportion of the customers willing to choose large time windows can significantly improve the capacity utilization. We demonstrate that there are diminishing returns to the increased fraction of customers being flexible. Moreover, the benefit of the limited flexible large time window design can capture most of the total capacity pooling benefit.



(a) (b)

Figure 2.1: The time window design from (a) FreshDirect.com and (b) Peapod.com

2.1 Introduction

Due to the rapid development of mobile and web-based technologies, online booking system for scheduled services becomes increasingly popular. Examples include online grocery delivery, home maintenance, beauty services, health-care appointments, and restaurant reservations. In all of these examples, customers select a specific time window to receive the service from a menu of time windows presented by the service provider. In practice, there are various ways to design time windows. Figure 2.1(a) shows an example from FreshDirect.com that uses non-overlapping time windows and Figure 2.1(b) shows an example from Peapod.com that uses overlapping windows and large windows.

The capability to fulfill demand on time is critical in keeping consumer satisfaction and the compatibility of service providers. Time windows with short duration are more attractive to customers. However, due to demand randomness, it is possible that the capacity in some time windows can not satisfy demands, while at the same time there are other time windows

with remaining capacity. The goal of the firm is to avoid such situation and to satisfy as much demand as possible with their current capacity.

A classical approach to improve capacity utilization for a system with multiple class of demands and dedicated capacities is the capacity pooling. The key idea is to let the capacity be able to serve not just one type of demands. The flexibility of allocating any capacity to any type of demand helps the firm to improve service quality. However, the traditional capacity pooling approach can not be adapted to the service booking system. We focus on a setting where all customers want the same service and the demand types are differentiated only by the time when a customer get served. There is no flexible capacity – a unit of capacity at 8am can not be used to serve a customer at 9am.

Motivated by the fact that some consumers are willing to sacrifice some of their choices in exchange for discounts, the firm can offer large time windows (LTWs) at a lower price in addition to the small windows to better utilize capacities. This consumer behavior falls within the concept of *consumer flexibility*, or demand-side flexibility, which refers to a consumer's willingness to receive one of multiple options that are offered to them instead of a specific option, in exchange for a reward or discount. Although the classical capacity pooling approach is not applicable here, the firm can use small incentives to attract flexible customers to be willing to get served in multiple time windows and achieve the capacity pooling benefit. The goal of this paper is to make progress toward the better understanding in the capacity pooling benefit through consumer flexibility and the key design principles of LTWs.

In order to measure the benefit, we first model the customers' choice process. We assume that there are n non-overlapping regular time windows (small time windows) in a day. We

propose a simple LTW design where large time windows are defined as the union of two consecutive regular time windows, $1\&2, 2\&3, \ldots, n-1\&n$, along with a special large time window $n\&1$ that combines the first and last regular time window. We assume consumers' preferences for regular time windows are symmetric and in addition, they have i.i.d. willingness to extend their initial choice to one of the large time windows that contains the initial choice. The probability a consumer being *flexible* is denoted by q. Implicitly, q depends on the discount, although we do not explicitly model this. After a cut-off time, demands are allocated to each regular time window. We assume that the capacity in any regular time window is a constant. Our allocation decision is made based on an optimization problem that maximizes the total demand fulfilled subject to the capacity constraints. We evaluate the expected performance of various LTW designs and compare with the design without LTWs.

2.1.1 Summary of Main Contributions

First, our model reveals the power of limited flexibility in consumer flexibility. With just a small fraction of customers willing to choose a LTW, a limited flexible LTW design can achieve most of the total potential capacity pooling benefit. We show that the expected fulfilled demand is an increasing concave function in q for a two regular time window system (Thorem 2.3.1), which explains why a small q is enough to capture most of the benefit. We also show that a limited flexible design with just a few LTWs, for example, a non-overlapping LTW design, is enough to capture most of the potential benefits.

Our analysis also demonstrates the fundamental difference between demand-side flexi-

bility and supply-side flexibility. Our large time window design closely resembles the 'long chain' design in the process flexibility literature (Jordan and Graves 1995). In the long chain design, it is well-known that incrementally adding flexibility has increasing returns (supermodularity), and as a byproduct 'closing the loop (from n to 1)' provides the most benefit (Simchi-Levi and Wei 2012). We find that the supermodularity property does not hold in our model, and that 'closing the loop' typically provides the least value in time window design. In fact, we prove that the benefit of closing the loop can be arbitrarily small. This highlights a fundamental difference between consumer flexibility and process flexibility – prioritizing coverage with large time windows is more important than having time windows configured as a chain.

Finally, we conduct extensive numerical experiments which yield several further insights. First, the diminishing return in the increased proportion of flexible customers is observed for general number of time windows. Furthermore, we examine the benefit of LTWs in a dynamic capacity allocation setting. By comparing with a dynamic pricing policy which dynamically providing discounts to the least popular time window, we show that offering LTWs from the beginning of the selling horizon can improve more in the capacity utilization while the total discounts being offered is lower. Finally, we apply the non-overlapping LTW design to an online grocery delivery problem and consider the travel time constraint rather than the capacity constraint. Offering LTWs can improve the number of requests fulfilled given the same travel time limit.

2.1.2 Literature Review

In this section, we briefly review literature that is most relevant to our work. Our research is closely related with consumer flexibility, process flexibility and time window management.

Innovative strategies that leverage consumer flexibility rapidly emerges in the online marketplace. Considering revenue management for airline ticket selling, Gallego and Phillips (2004) derive conditions and algorithms for the management of two perishable products and a flexible product. For the online car-sharing booking service, Ströhle et al. (2018) study the benefits of spatial and temporal customer flexibility using a real-world data set, where in this application, customers give up the exact knowledge of the pick-up and drop-off time and location. In the context of the online retailing industry, consumer flexibility can be realized through opaque selling, where some specific attributes such as color are not revealed to the customer until after purchase. Elmachtoub et al. (2015) study the opaque selling strategy with two substitutable products and demonstrate the benefit in inventory cost reduction. Elmachtoub et al. (2019) further extended the opaque selling model to the case with multiple types of products and proposed a limited flexible opaque selling strategy, which can reduce the inventory costs by the same order of magnitude as the fully flexible design, even with just a small fraction of customers being flexible. Our analysis also demonstrates the power of limited flexibility in both the fraction of customers being flexible and the design of flexible options. The focus of this paper is on the scheduled service systems, which is different from other applications of consumer flexibility in the aspect of performance metrics and methodologies.

The benefit of LTW designs fundamentally comes from the capacity pooling effect. Eppen

(1979) was the first to demonstrate the value of pooling effect on cost by aggregating all of the i.i.d. normal demands and introduced the well-known "Square-Root Law". Although the performance metric in the capacity pooling literature is different from the performance metric in our paper, the benefit all comes from the fact that pooling capacity together can hedge demand uncertainty. The literature of flexibility in manufacturing, service, and supply chain management demonstrates the capacity pooling effect on limited flexible systems (see Wang et al. (2019) for a recent survey). Jordan and Graves (1995) demonstrated that a partial flexible structure, the long chain design, can accrue most of the benefits achieved by a fully flexible system. The effectiveness of the long chain and other designs with limited flexibility has been investigated theoretically in many recent works (e.g. Simchi-Levi and Wei (2012), Wang and Zhang (2015), Désir et al. (2016)). We find that the notion of "a little flexibility goes a long way" still applies to consumer flexibility. However, our results show that consumer flexibility is intrinsically different from the previous notions of flexibility that comes from the supply side, and some key insights from process flexibility no longer drive the decision making.

Time window management has been extensively studied for scheduled services, especially in attended home delivery, where the requests for delivery are associated with a specific time window. Campbell and Savelsbergh (2005) propose an algorithm to accept or reject consumers' requests based on opportunity cost and demonstrate that expanding a one-hour time window to two hours can increase profits by more than 6% and can be increased an additional 5% if further expanded to three hours. Campbell and Savelsbergh (2006) study the model where service providers may offer incentives for the selection of certain time windows and show that incentive schemes can substantially reduce delivery costs, since customers

may accept wider service time windows supporting a more efficient combination of requests in vehicle routing and scheduling operations. Yang et al. (2014) propose approximating opportunity costs based on the insertion heuristic while also incorporating information about predicted demand. Agatz et al. (2011) study the problem that decides which time slots to offer for each zip code, so as to minimize expected delivery costs. Recently, Köhler et al. (2020) introduce the idea of flexible time window management. In the booking process, their approach first offers long delivery time windows, while shorter time windows are offered to certain customers after careful consideration. Our large time window design does not display long time windows as the only option to some customers. Instead, we propose to show the customers both short and long time windows at the same time and let the customers make the choice. Besides attended home delivery, time window management has also been studied for general scheduled service systems. Liu et al. (2019) consider the online appointment booking problem where the service provider offers assortments of time slots to a customer in multiple stages. If a patient is offered an assortment that includes time slots she is interested in, then she chooses among them uniformly. The authors characterize the optimal sequence of time slots to offer. The distinguishing feature of our research from all previous work is that we leverage consumer flexibility in time window design and evaluate the capacity pooling benefit introduced by large time windows.

The remainder of the paper is organized as follows. Section 2.2 introduces the model of the consumers' behavior and the scheduled service system. Section 2.3 shows the effectiveness of the limited flexible LTW design and concave property in q. Section 2.4 discuss the connection and distinction between the time window design and the long chain design. In

Section 2.5, we conduct extensive numerical experiments. In Section 2.6, we specifically design a numerical experiment including vehicle routing decisions and validate the benefit of large time windows in scheduled delivery systems. Finally we conclude in Section 2.7.

2.2 Model and Notation

We consider a service provider using an online booking system to schedule the service in a day in the future. The service provider creates a menu of time windows for the customers to choose from, and the service is committed to be completed during the selected time window. There is a cutoff time for each day where the booking system will be closed after the cutoff time and then the service provider will allocate the capacities in a way that best utilizes its own capacity. The demands that can not be fulfilled will be lost or redirected to a third party service provider. The goal of the service provider is to maximize the number of jobs that can be served with its own capacity.

Next, we formally define the design of time windows we focus on in this paper. There are n non-overlapping regular time windows in a day, denoted as $1, 2, \ldots, n$. Each regular time window has the same duration. For example, the two-hour time windows in Figure 2.1 are regular time windows in our setting. The service capacity in each regular time window is a constant C.

Additional to the regular time windows the service provider has $2^n - n - 1$ possible ways to combine regular time windows into large time windows (LTWs). However, too many options displayed to customers may be confusing and hard to implement in reality. Thus, we focus on LTWs that are the union of two consecutive regular time windows, i.e.,

$1\&2, 2\&3, \ldots, n-1\&n$, along with a special LTW $n\&1$. A customer who chooses LTW $i\&j$ will be served in either time window i or j.

We focus on the performance of time window designs that are composed of all regular time windows and a subset of the LTWs. Let L_n^i denote the set of time windows that includes all regular time windows and the first i large time windows.

$$L_n^0 := \{1, \ldots, n\},$$

$$L_n^i := \{1, 2, \ldots, n\} \cup \{1\&2, \ldots, i\&i+1\}, i = 1, \ldots, n-1,$$

$$L_n^n := \{1, 2, \ldots, n\} \cup \{1\&2, \ldots, n-1\&n, n\&1\}.$$

Note that L_n^0 is the time window design with only regular time windows.

2.2.1 Demand model

We model the customer's selection process as the following. We assume that there is an underlying initial choice t for a customer and a customer who is *flexible* is willing to extend to a LTW that contains his or her initial choice. Whether a customer is flexible or not is an independent Bernoulli event and we denote the probability of a customer being flexible as q, for a given discount level. Implicitly, q depends on the discount, although we do not explicitly model the relationship between q and discount level here. We only make a mild assumption here that is q is non-decreasing in the discount level.

There are two LTWs that contains a regular time window t. We assume the preference is symmetric. Then a customer with initial choice t is willing to extend to one of the LTWs that contains t, which is either $t-1\&t$ or $t\&t+1$, each with probability $\frac{q}{2}$. The customer

selects the desired LTW if it is included in the time window design. Otherwise, this customer will keep the initial choice, which is the regular time window t.

Let X_t denote the demand for regular time window t in design L_n^0, i.e., when there is no LTW. X_t is composed of consumers who are not flexible, consumers who are willing to choose LTW $t-1\&t$ and consumers who are willing to choose LTW $t\&t+1$. Let ξ_t^1 be the number of customers with initial choice t who are not flexible, ξ_t^2 be the number of customers with initial choice t who are willing to extend to LTW $t-1\&t$, and ξ_t^3 be the number of customers with initial choice t who are willing to extend to LTW $t\&t+1$. $X_t \stackrel{d}{=} \xi_t^1 + \xi_t^2 + \xi_t^3$. Given $X_t = x_t$, $\xi_t^1, \xi_t^2, \xi_t^3$ follows multinomial distribution with parameter $x_t, (1-q, q/2, q/2)$.

For LTW design L_n^i with $i < n$, the demand vector $\mathbf{D}$ can be constructed as follows.

$$
\begin{aligned}
D_1 &= \xi_1^1 + \xi_1^2 \\
D_t &= \xi_t^1, \quad \text{for } t = 2, \ldots, i, \\
D_{i+1} &= \xi_{i+1}^1 + \xi_{i+1}^2 \\
D_t &= \xi_t^1 + \xi_t^2 + \xi_t^3, \quad \text{for } t = i+1, \ldots, n, \\
D_{t\&t+1} &= \xi_t^3 + \xi_{t+1}^2, \quad \text{for } t = 1, \ldots, i.
\end{aligned}
$$

For LTW design L_n^n, the demand vector $\mathbf{D}$ can be constructed as follows.

$$
\begin{aligned}
D_t &= \xi_t^1, \quad \text{for } t = 1, \ldots, n, \\
D_{t\&t+1} &= \xi_t^3 + \xi_{t+1}^2, \quad \text{for } t = 1, \ldots, n.
\end{aligned}
$$

2.2.2 Performance Metric

In this subsection, we define the comparison metric of different time window designs. The service provider wants to maximize the demands it can serve within the capacity constraint. The metric we focus on in this paper is the expected demands that can be served for a given time window design.

For a given demand realization, the problem can be formulated as a max flow problem on a bipartite graph. For time window design L_n^i, we define graph G_n^i as a bipartite graph where the left-hand-side nodes are $\{w_t : t \in L_n^i\}$, representing all the time windows offered in L_n^i, and the right-hand-side nodes are $\{c_1, \ldots, c_n\}$, representing the actual capacity in each regular time window. We use E_n^i to represent the corresponding edge set in graph G_n^i. The direction of flow is from the left-hand-side nodes to the right-hand-side nodes. Figure 2.2 shows the bipartite graph for design L_4^3 and L_4^4.

For design L_n^i, a demand realization is represented by vector $\mathbf{d} := (d_t)_{t \in L_n^i}$. The inflow of node w_t can not exceed the demand realization d_t and the outflow of the node c_i can not exceed the capacity level C. Let f_{tj} denote the flow from node w_t to c_j, which is the capacity allocated from regular time window j to time window k. For design L_n^i and a q value, the max flow problem can be formulated as (2.1).

$$
\begin{aligned}
P(\mathbf{d}; L_n^i) = \max \quad & \sum_{(t,j) \in E_n^i} f_{tj} \\
s.t. \quad & \sum_{(t,j) \in E_n^i} f_{tj} \leq C, \quad \forall j = 1, \ldots, n, \\
& \sum_{(t,j) \in E_n^i} f_{tj} \leq d_t, \quad \forall t \in L_n^i,
\end{aligned}
\tag{2.1}
$$

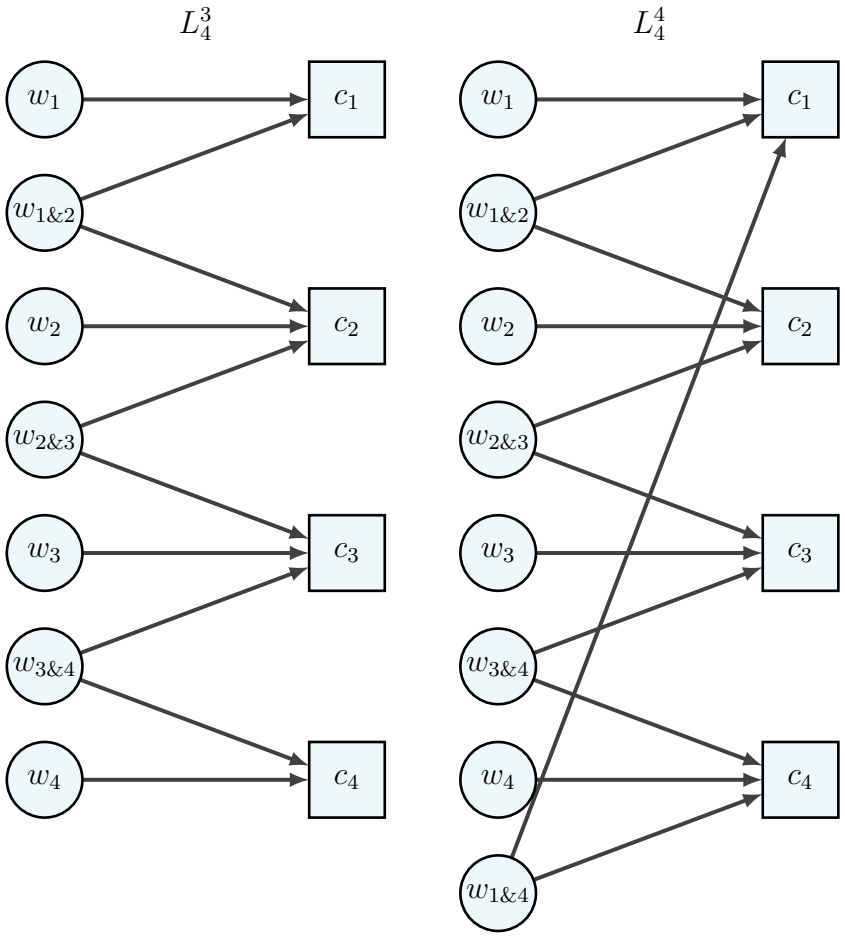


Figure 2.2: Graph representation of time window designs

$$f_{tj} \geq 0, \quad \forall (t,j) \in E_n^i.$$

Next we introduce Algorithm 1, a greedy algorithm that finds the optimal solution of problem (2.1) when there is no loop in the graph associates with design L_n^i. The main idea is first allocating the maximum amount of capacity to the customers who are not flexible, and then using the remaining capacity to satisfy demand in LTW $t\&t+1$ starting from $t = 1$ to i. We formally state the algorithm below.The correctness of Algorithm 1 is omitted and it is easy to verify that there is no augmenting path in the graph.

The performance metric we focused on is the expected number of jobs the firm can serve.

Algorithm 1: Greedy algorithm for optimal flow in $L_n^i, 0 \le i \le n-1$

for $j = 1, \ldots, n$ **do**
  $f_{jj}^* = \min\{C, d_j\}$
$f_{1\&2,1}^* = \min\{C - f_{11}^*, d_{1\&2}\}$;
$f_{1\&2,2}^* = \min\{C - f_{22}^*, d_{1\&2} - f_{1\&2,1}^*\}$;
for $j = 2, \ldots, i$ **do**
  $f_{j\&j+1,j}^* = \min\{C - f_{jj}^* - f_{j-1\&j,j}^*, d_{j\&j+1}\}$;
  $x_{j\&j+1,j+1}^* = \min\{C - x_{j+1,j+1}^*, d_{j\&j+1} - x_{j\&j+1,j}^*\}$;

Let $F(L_n^i; q)$ denote the expected max flow value with LTW design L_n^i for a given q,

$$F(L_n^i, q) = \mathbb{E}_{\mathbf{D}}[P(\mathbf{D}; L_n^i)].$$

The following Lemmas shows that the performance is always better with more flexible customers and with more LTWs.

Lemma 2.2.1. *For* $0 \le i \le j \le n$ *and* $0 \le q_1 \le q_2 \le 1$, $F(L_n^i, q_1) \le F(L_n^j, q_2)$.

Proof. Given a demand scenario $\mathbf{d}$ and n multinomial random variable realization, the optimal solution of problem $P(\mathbf{d}, L_n^i)$ is always a feasible solution to problem $P(\mathbf{d}, L_n^{i+1})$. Therefore, the max flow with design L_n^{i+1} is at least as large as L_n^i and so does the expected performance.

Using the same argument, for a fixed design and with different q values, the result follows directly from the stochastic dominance of binomial random variables. □

In particular, we denote the performance of the no LTW design as $F(L_n^0)$, which drops the dependence on q. $F(L_n^0) = \sum_{i=1}^n \mathbb{E}[\min\{X_i, C\}] = n\mathbb{E}[X_1 | X_1 \le C]$. In this paper, we focus on the benefit of adding LTWs comparing with the no LTW design. The benefit then can be written as $F(L_n^i, q) - F(L_n^0)$.

We finally define a performance benchmark in this paper to be the fully flexible system $\mathcal{F}_n$. In a fully flexible system, any demands can be fulfilled in any time window. The expected performance of $\mathcal{F}_n$ can be written as

$$F(\mathcal{F}_n) = \mathbb{E}[\min\{\sum_{i=1}^{n} X_i, nC\}].$$

The assumption of system $\mathcal{F}_n$ is unrealistic under our setting. However, the performance difference between the fully flexible system and the no flexible system is the maximum potential benefit of capacity pooling, which is naturally an upper bound of the performance of any LTW design. For example, assume X_i's follow Normal distribution $N(\mu, \sigma)$ with $\mu = C$. Then an upper bound of the performance of any LTW design can be written as:

$$\begin{aligned} F(L_n^i, q) - F(L_0) &\leq F(\mathcal{F}_n) - F(L_0) \\ &= \mathbb{E}[\min\{\sum_{i=1}^{n} X_i, nC\}] - n\mathbb{E}[\min\{X_1, C\}] \\ &= n\mu - \sqrt{n}\sigma\frac{\phi(0)}{\Phi(0)} - n\left(\mu - \sigma\frac{\phi(0)}{\Phi(0)}\right) \\ &= (n - \sqrt{n})\sigma\sqrt{\frac{2}{\pi}}. \end{aligned} \tag{2.2}$$

The example in Equation (2.2) resembles the well-known "Square-Root Law" by Eppen (1979) in the risk pooling literature. The benefit of LTWs fundamentally comes from the potential to pool capacities. The question is, how much capacity pooling benefit can a limited flexible LTW design L_n^i capture with a limited flexible level q. We seek to answer this question in the following sections.

2.3 Capacity Pooling Benefit from LTWs

In this section, we present and prove some properties and insights of the capacity pooling effect through LTWs.

We start with a motivating numerical example. Assume a firm provides 8 regular time windows in a day and the demand for each regular time window follows uniform distribution between 50 and 150. We simulate demands and compute the average performance for various LTW designs. Two comparison bench marks are $F(L_n^0) = 698.02$ and $F(\mathcal{F}_n) = 765.12$, where the first one is the worst performance among all designs and the second one is the best possible performance. We are interested in how much benefit can limited flexible designs capture and how large q is required for a significant improvement. We first focus on the performance with $q = 1$ and then shrink the q value.

Table 2.1: $F(L_n^i, 1)$ for $i = 1, \ldots, 8$

L_8^1	L_8^2	L_8^3	L_8^4	L_8^5	L_8^6	L_8^7	L_8^8
706.37	715.19	724.66	734.17	743.80	753.34	763.02	765.09

In Table 2.1, we list $F(L_n^i, 1)$ for $i = 1, \ldots, 8$. The first observation is that the performance of design L_n^n is very close to the performance of the fully flexible system. Also, the benefit of adding the last LTW 1&8 is significantly smaller than the benefit of adding any other LTW. Adding any other LTW can improve the expected fulfilled demand by around 9 units, while the benefit of adding the last LTW is only 2. Indeed, $F(L_8^7, 1)$ is already very close to $F(\mathcal{F}_n)$. Therefore, with $q = 1$, the limited flexible design L_n^{n-1} is almost as effective as the fully flexible system.

Next, we shrink the proportion of customers who are willing to choose a LTW. When $q = 20\%$, the expected performance of design L_n^{n-1} is 731.84. Considering the fraction of total benefit this design captures,

$$\frac{F(L_8^7, 0.2) - F(L_0)}{F(\mathcal{F}_8) - F(L_0)} = \frac{731.84 - 698.02}{765.12 - 698.02} \approx 50.4\%.$$

With only 20% of flexible consumers, the firm can gain half of the capacity pooling benefit. The insights from this numerical example is that the firm can guarantee a significant proportion of improvements from a simple LTW design with a small discount level (which implies a small q value) and the LTW 1&n has limited marginal value.

In the following subsections, we first explain the stunning performance of small q by proving the concave property in q when $n = 2$ and then formally show that LTW 1&n is not necessary to the system.

2.3.1 Concavity in q

In this subsection, we focus on the concave behavior in q.

Theorem 2.3.1. *$F(L_2^2, q)$ is concave in q.*

Note that when $n = 2$, LTW 1&2 and 1&n are the same and it is the only possible LTW. This is a degenerate case. Therefore, design L_2^2 is essentially regular time window 1,2 and a LTW 1&2. Let X_1, X_2 be the random demand for regular time window 1 and 2 in design L_2^0. Since a customer is willing to choose LTW 1&2 with probability q, given X_1, X_2, the number of customers who chooses LTW 1&2 is the summation of two Binomial random variables,

$Bin(X_1, q) + Bin(X_2, q)$.

Proof of Theorem 2.3.1. The goal is to show that for any $0 \leq q_1 < q_2 < q_3 \leq 1$ such that $q_3 - q_2 = q_2 - q_1$, $F(L_2^1, q_3) - F(L_2^1, q_2) \leq F(L_2^1, q_2) - F(L_2^1, q_1)$.

We first analyze the max flow for a deterministic demand and then adapt the submodularity result in Gale and Politof (1981).

Let x_1, x_2 be arbitrary demand realizations for X_1, X_2. Let $Y_1 \stackrel{d}{=} Bin(x_1, q)$ denote the number of flexible customers who initially choose regular time window 1 and $Y_2 \stackrel{d}{=} Bin(x_2, q)$ denote the number of flexible customers who initially choose regular time window 2. The total number of customers choosing the LTW is $Y_1 + Y_2$. The corresponding demand that remains in regular time window i is $x_i - Y_i \stackrel{d}{=} Bin(x_i, 1-q)$. For $i = 1, 2$, $(Y_i, x_i - Y_i)$ follows the multinomial distribution with parameter $(x_i, q, 1-q)$. Figure 2.3 represents the demand shifting behavior for the $n = 2$ case. Node $w_{1\&2}^i$ is the demand for LTW splited from regular time window i. Opening the LTW is equivalent to add two edges $(w_{1\&2}^1, c_2)$ and $(w_{1\&2}^2, c_1)$ to the graph.

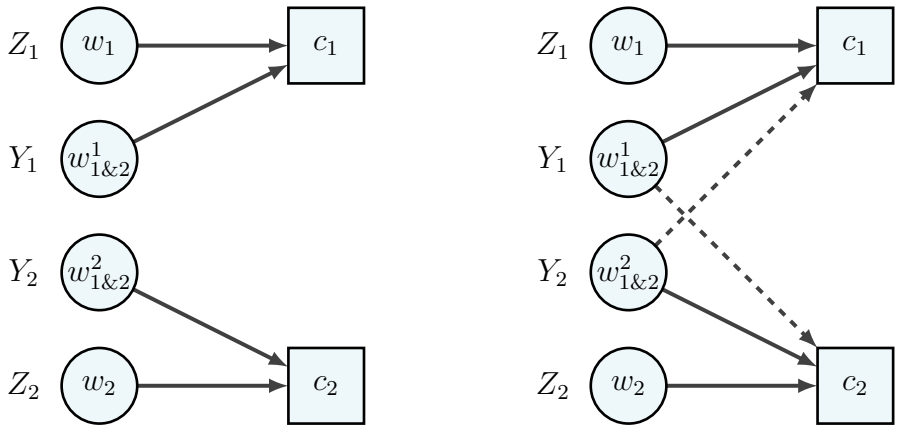


Figure 2.3: Equivalent representation for L_2^0 and L_2^1

We next claim that the expected benefit of adding LTW 1&2 are contributed equally from

the demand split from X_1 and X_2. Let $M(a, b, c)$ be the max flow on the bipartite graph in Figure 2.4 where the a, b, c are the upper bound of the total inflow of node $w_1, w^1_{1\&2}, w_2$. $M(x_1 - y_1, y_1, x_2)$ represents the case where only customers from regular time window 1 will shift to the LTW when the service provider opens the LTW option.

Lemma 2.3.1. $F(L^1_2, q) - F(L^0_2) = 2[\mathbb{E}[M(X_1 - Y_1, Y_1, X_2)] - F(L^0_2, q)]$

The proof of Lemma 2.3.1 is provided in Appendix 2.8.1. With Lemma 2.3.1, we can only focus on the max flow in Figure 2.4.

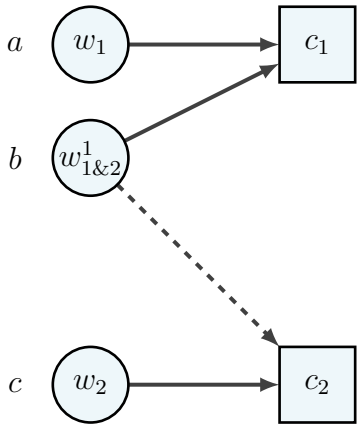


Figure 2.4: Max flow problem $M(a, b, c)$

Next, we consider the change in Figure 2.4 with increased q values in Figure 2.5. Consider a demand instance x_1, x_2, a bipartite graph in Figure 2.5 and two specific arcs $\alpha = (c, c_2)$ and $\beta = (d, c_2)$ with given nonnegative capacity upper bound u_α and u_β. Let E denote the edge set. Let $\mathbf{Z}$ be a multinomial random vector with parameter $(x_1, 1-q_3, q_1, q_3-q_2, q_2-q_1)$ and $\mathbf{z}$ be a instance of $\mathbf{Z}$. Define

$$P_{\alpha,\beta}(u_\alpha, u_\beta; \mathbf{z}, x_2) = \max \sum_{(i,j)\in E} f_{ij}$$

$$
\begin{aligned}
s.t. \quad & \sum_{(a,j)\in E} f_{aj} \le z_1, \quad \sum_{(b,j)\in E} f_{bj} \le z_2, \\
& \sum_{(c,j)\in E} f_{cj} \le z_3, \quad \sum_{(d,j)\in E} f_{dj} \le z_4, \\
& \sum_{(i,c_1)\in E} f_{ic_1} \le C, \quad \sum_{(i,c_2)\in E} f_{ic_2} \le C, \\
& f_\alpha \le u_\alpha, \quad f_\beta \le u_\beta, \\
& f_{w_2c_2} \le x_2, \quad f_{ij} \ge 0, \quad \forall (i,j) \in E.
\end{aligned}
$$

We first prove that $P_{\alpha,\beta}(u_\alpha, u_\beta; \mathbf{z}, x_2)$ is submodular in u_α and u_β. We apply the main theorem in Gale and Politof (1981), which shows that if two arcs in the graph are in parallel, then the max flow solution is submodular with respect to the capacity of both arc. In a directed graph, two arcs are said to be in parallel, if for any cycle containing both arcs, they have the opposite direction when we fix an orientation of the cycle. Note that α and β has the same head and therefore they are in parallel and we have that $P_{\alpha,\beta}(u_\alpha, u_\beta; \mathbf{z}, x_2)$ is submodular in u_α and u_β. Therefore, we have

$$
P_{\alpha,\beta}(C, C; \mathbf{z}, x_2) + P_{\alpha,\beta}(0, 0; \mathbf{z}, x_2) \le P_{\alpha,\beta}(C, 0; \mathbf{z}, x_2) + P_{\alpha,\beta}(0, C; \mathbf{z}, x_2). \tag{2.3}
$$

Next, we construct the connection between $P_{\alpha,\beta}$ and the expected max flow $F(L_2^2, q)$. Note that the following equations hold.

$$
\begin{aligned}
P_{\alpha,\beta}(C, C; \mathbf{z}, x_2) &= M(z_1, z_2 + z_3 + z_4, x_2), \\
P_{\alpha,\beta}(0, 0; \mathbf{z}, x_2) &= M(z_1 + z_3 + z_4, z_2, x_2),
\end{aligned}
$$

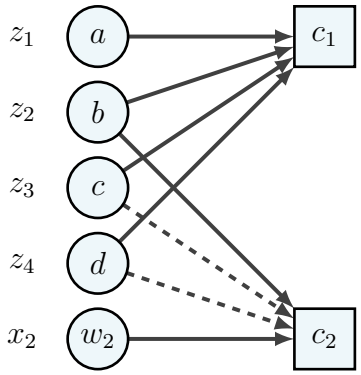


Figure 2.5: Illustration for the proof of Theorem 2.3.1

$$P_{\alpha,\beta}(C, 0; \mathbf{z}, x_2) = M(z_1 + z_3, z_2 + z_4, x_2),$$

$$P_{\alpha,\beta}(0, C; \mathbf{z}, x_2) = M(z_1 + z_4, z_2 + z_3, x_2).$$

By definition of a multinomial random vector (Z_1, Z_2, Z_3, Z_4) with parameter $(x_1, 1-q_3, q_1, q_3-q_2, q_2-q_1)$, we can think $Z_i + Z_j$ as the number of event i or j happens, which implies that $(Z_1 + Z_3 + Z_4, Z_2)$ follows multinomial distribution with parameter $(x_1, 1-q_1, q_1)$, $(Z_1, Z_2 + Z_3 + Z_4)$ follows multinomial distribution with parameter $(x_1, 1-q_3, q_3)$, and $(Z_1 + Z_3, Z_2 + Z_4)$ follows multinomial distribution with parameter $(x_1, 1-q_2, q_2)$. Finally note that $q_3 - q_2 = q_2 - q_1$, $(Z_1 + Z_4, Z_2 + Z_3)$ also follows multinomial distribution with parameter $(x_1, 1-q_2, q_2)$. Combining with Lemma 2.3.1, we have the following equations.

$$\mathbb{E}[P_{\alpha,\beta}(C, C; \mathbf{Z}, X_2)] = \mathbb{E}[M(Z_1, Z_2 + Z_3 + Z_4, X_2)] = \frac{1}{2}(F(L_2^2, q_3) - F(L_2^0, q_3)) + F(L_2^0, q_3),$$

$$\mathbb{E}[P_{\alpha,\beta}(0, 0; \mathbf{Z}, X_2)] = \mathbb{E}[M(Z_1 + Z_3 + Z_4, Z_2, X_2))] = \frac{1}{2}(F(L_2^2, q_1) - F(L_2^0, q_1)) + F(L_2^0, q_1),$$

$$\mathbb{E}[P_{\alpha,\beta}(C, 0; \mathbf{Z}, X_2)] = \mathbb{E}[M(Z_1 + Z_3, Z_2 + Z_4, X_2))] = \frac{1}{2}(F(L_2^2, q_2) - F(L_2^0, q_2)) + F(L_2^0, q_2),$$

$$\mathbb{E}[P_{\alpha,\beta}(0, C; \mathbf{Z}, X_2)] = \mathbb{E}[M(Z_1 + Z_4, Z_2 + Z_3, X_2))] = \frac{1}{2}(F(L_2^2, q_2) - F(L_2^0, q_2)) + F(L_2^0, q_2).$$

Plugging into inequality (2.3), we can get

$$F(L_2^2, q_3) - F(L_2^2, q_2) \leq F(L_2^2, q_2) - F(L_2^2, q_1),$$

which completes the proof. □

For general n, unfortunately we can not show the concave property. Nevertheless, the concave behavior can be observed in extensive numerical results in Section 2.5.

2.3.2 Unnecessity of LTW $1\&n$

From the numerical example, we observe that the benefit of LTW $1\&n$ is rather limited comparing to other LTWs. Moreover, the LTW $1\&n$ is hard to implement in reality. In Theorem 2.3.2, we show that the benefit of LTW $1\&n$ can be arbitrarily small comparing to other LTWs $t\&t+1$.

Theorem 2.3.2. *For $n = 3$ and any $\epsilon > 0$, there exists a demand distribution where the benefit of closing the loop is less than ϵ times the benefit of adding any other LTW.*

The proof of Theorem 2.3.2 is provided in Appendix 2.8.1. Theorem 2.3.2 implies that the firm should just considering use design L_n^{n-1} in reality for its simplicity and efficiency. Note that from Algorithm 1 we know that the optimal solution for design L_n^{n-1} can be found in $\Theta(n)$ time, while finding the optimal solution for design L_n^n requires solving problem (2.1).

2.4 Connection to the Long Chain Design

In this section, we state the connection between our model and the long chain design in the literature of process flexibility. The long chain design is a special configuration of a production system. There are n different product types and n different production plants. Each plant produces exactly two products and each product is produced in exactly two plants, in a way that "chains" all the products and plants.

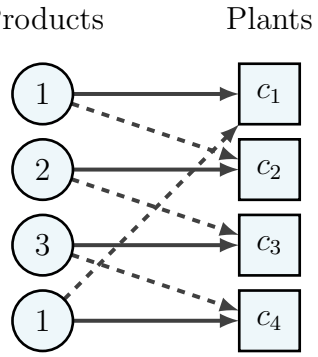


Figure 2.6: Long chain design

First, we formally define a long chain design and the optimization problem associated with a long chain. Let graph $\tilde{G} = (\tilde{V}, \tilde{E})$ denote the graph of a long chain, which is a bipartite graph. $\tilde{V}$ is composed of two set of disjoint nodes. $\tilde{V} = \tilde{L} \cup \tilde{R}$, where $\tilde{R} = \{1, 2, \ldots, n\}$ denoting n plants and $\tilde{L} = \{1, 2, \ldots, n\}$ denoting n products. Demand for product t can be fulfilled from two production plants t and $t+1$. (When $t = n$, the demand can be fulfilled from plant n and 1). The set of edges are $\tilde{E} = \{(t, t), (t, t+1), t = 1, \ldots, n\}$, which connects $\tilde{L}$ with $\tilde{R}$. To maximize the number of jobs can be served in graph $\tilde{G}$ with demand vector

$\tilde{\mathbf{d}}$ and capacity C, the optimization problem can be written as

$$\begin{aligned} \tilde{P}(\tilde{\mathbf{d}}, \tilde{G}) = \max & \sum_{(i,j)\in\tilde{E}} f_{ij} \\ s.t. & \sum_{(i,j)\in E} f_{ij} \le C, \forall j \in \tilde{R} \\ & \sum_{(i,j)\in E} f_{ij} \le d_i, \forall i \in \tilde{L} \\ & f_{ij} \ge 0, \forall (i,j) \in E \end{aligned} \tag{2.4}$$

Back to the optimization problem in (2.1). If removing nodes $1, \ldots, n$ from L_n and all the edges associate with them, then graph G_n is the same as the graph of a long chain. When $d_t = 0$, for all $t = 1, \ldots, n$, the constraints in problem $P(\mathbf{d}, G)$ associated with node $t = 1, \ldots, n \in L$ degenerates to $x_{tt} = 0$, for $t = 1, \ldots, n$. The solution of problem $P(\mathbf{d}, G)$ remains the same after removing variable x_{tt}'s. Then problem $P(\mathbf{d}, G)$ and $\tilde{P}(\tilde{\mathbf{d}}, \tilde{G})$ have the same constraints and objective function. Note that if the demand follows i.i.d. Poisson distribution, then not only the graphs are the same, but also the expected performance will be exactly the same for LTW design L_n^n $(q = 1)$ and the long chain design. For general distributions, we can construct the demand for a long chain design to make the expected performance identical in these two systems.

Proposition 2.4.1. *When $q = 1$, the expected performance $F(L_n^n, 1)$ is equal to the performance of a long chain design with demand for each product follows $\tilde{D}_t \stackrel{d}{=} \xi_t^3 + \xi_{t+1}^2$.*

Proposition 2.4.1 directly connects the performance of a LTW design with the long chain design. In the literature of process flexibility, the good performance of long chain has been

demonstrated in various metrics. Especially, its capability to perform almost as good as a fully flexible system has been observed both numerically and theoretically. Therefore, we can conclude that the performance of design L_n^n with all consumers being flexible can also achieve the majority of capacity utilization improvement.

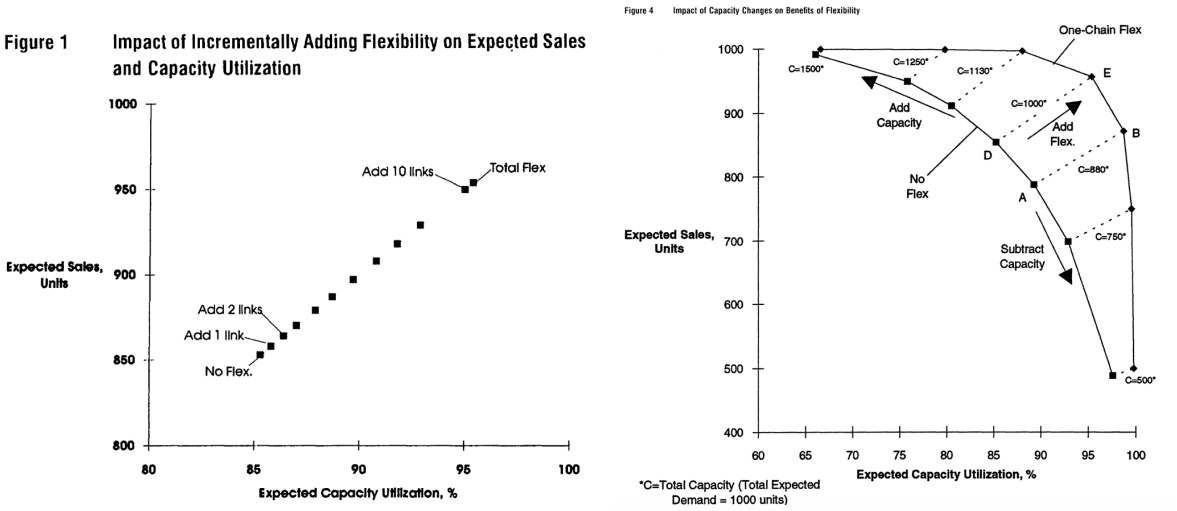
2.4.1 Numerical Comparison

We conduct a numerical experiment with the same settings in Jordan and Graves (1995) to see whether properties in long chain design also hold in our model.

The setting in Jordan and Graves (1995) is a system with 10 products and 10 plants, i.e. $n = 10$. The demand distributions are i.i.d. truncated normal with mean 100 and standard deviation 40, upper and lower bounded by 180 and 20. The capacity for each plant is $C = 100$. We assume that in design L_n^0, the demand for each regular time window follows the same truncated normal distribution and apply the multinomial split to each demand realization. Figure 2.7 are the original graphs in Jordan and Graves (1995). We use the same representation of metrics and plot the results in Figure 2.8.

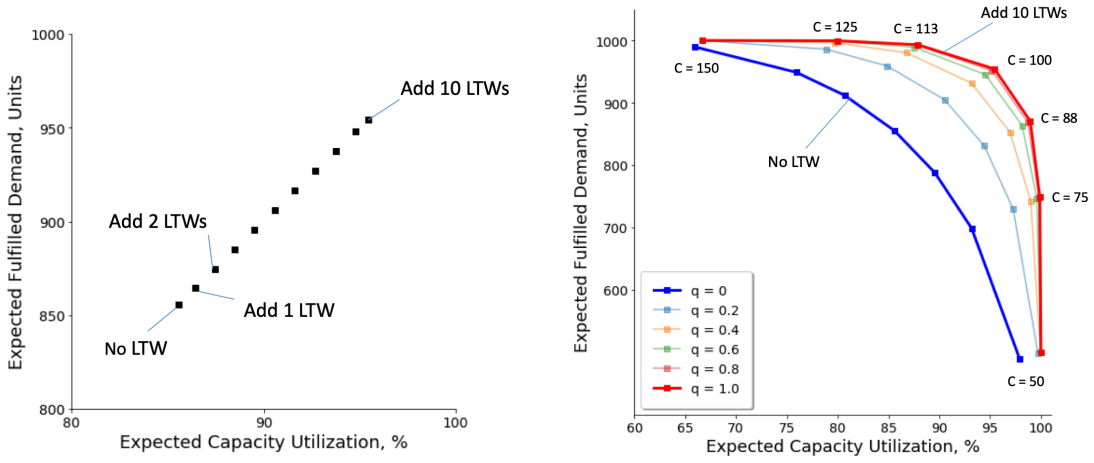
Figure 2.7(a) shows the extra benefit of adding each flexible link to the system. There are two key observations from Figure 2.7(a) First, adding each increment of flexibility yields increasingly greater sales and utilization benefit when constructing the long chain. Second, the performance of the long chain design is close to the fully flexible system. Figure 2.7(b) shows that adding flexibility can significantly improve capacity utilization and expected fulfilled demand at the same time for a variety of capacity levels.

Comparing Figure 2.7 and Figure 2.8, we can see both similarity and distinction between



(a) (b)

Figure 2.7: Numerical Results from Jordan and Graves (1995)



(a) (b)

Figure 2.8: Numerical results using the setting from Jordan and Graves (1995)

the LTW design and long chain design. We can observe similar behavior from Figure 2.8(b) such that for all q values, introducing a limited flexible LTW design can significantly improve the capacity utilization and expected fulfilled demand at the same time, no matter the capacity level is high or low. Moreover, we can observe the concave behavior in q, which is consistent with our finding for $n = 2$ in Theorem 2.3.1.

In Figure 2.8(a), although the performance of L_n^n is very close to the fully flexible design, adding each LTW does not yield increasing benefit. We provide a theoretical explanation of why the proof of the increasing benefit in Simchi-Levi and Wei (2012) does not go through in the time window design problem in Appendix 2.8.2.

We can conclude two important guidelines for LTW design from the comparison from Figure 2.7 and 2.8:

- It is not necessary to construct a closed loop, i.e. L_n^{n-1} can be effective enough;
- Even a chain structure is not necessary and non-overlapping LTWs can be very effective.

2.4.2 The Efficiency of Non-overlapping LTWs

The lack of supermodularity in our setting is rather a fortunate outcome, as a high utilization can be achieved without offering too many choices. Clearly, the window $n\&1$ would be difficult to implement in practice anyhow. In this section, we will investigate the performance of using just non-overlapping large time windows. In particular, we will focus on the case where n is a even number. Let $N_n = \{1, 2, \ldots, n, 1\&2, 3\&4, \ldots, n-1\&n\}$, denoting the design with $\frac{n}{2}$ non-overlapping LTWs.

Numerical examples suggest that using just a half number of LTWs roughly achieves a half of the total benefit.

Table 2.2: The incremental benefit from constructing L_n^n and the proportion of improvement captured by N_n when $n = 8$ and $q = 1$

Demand	1	2	3	4	5	6	7	8	$\frac{F(N_n,1)-F(L_n^0)}{F(L_n^n,1)-F(L_n^0)}$
Two point 50, 150	24.87	12.58	21.37	15.75	19.81	17.12	19.13	11.63	0.699
Two point 20, 180	40.01	20.10	30.80	24.98	28.20	26.51	27.42	17.42	0.743
Poisson(100)	2.33	2.73	2.91	3.03	3.11	3.21	3.26	0.00063	0.452
Uniform[50,150)	8.35	8.82	9.47	9.51	9.63	9.54	9.68	2.07	0.498
Normal(100,50)	10.87	12.15	12.55	12.51	12.58	12.51	12.71	7.49	0.466

For general n, we have the concavity in q for design N_n.

Corollary 2.4.1. *$F(N_n, q)$ is concave in q.*

Proof. With balanced demand across regular time windows, the benefit of adding a non-overlapping LTWs is equivalent to the benefit of adding and only adding the first LTW 1&2. This directly follows from the fact that N_n can be decomposed to $n/2$ disconnected subgraphs and the improvement in each subgraph is equal to the benefit of adding LTW 1&2. Therefore, we have $F(N_n, q) - F(L_n^0) = \frac{n}{2}(F(L_n^1, q) - F(L_n^0))$. Using the same proof as Theorem 2.3.1, we can show that $F(L_n^1, q) - F(L_n^0)$ is concave in q. □

2.5 Numerical Experiments

In this section, we use extensive numerical experiments to show the effectiveness of limited flexibility in both the time window design and proportion of customers choosing LTW options. We first focus on balanced systems and then extend to unbalanced systems with more general demand and capacity assumptions.

2.5.1 Concavity in q

We first conduct a simulation example with uniform demands. Assume there are 10 regular time windows in a day, each with i.i.d. uniformly distributed demand with mean 100. The capacity level for each regular time window is also equla to 100. For 5 correlation of variation (CV) levels $(0.1, 0.2, \ldots, 0.5)$, we compute the relative improvement in number of fulfilled demand for LTW design L_n^{n-1} and L_n^n, which is

$$\frac{F(L_n^{n-1}, q) - F(L_n^0, q)}{F(L_n^0, q)} \text{ and } \frac{F(L_n^n, q) - F(L_n^0, q)}{F(L_n^0, q)}.$$

In Figure 2.9, we first can observe that across all CV levels and LTW designs, the relative improvement has a concave shape in q. The second observation is that the relative improvement is higher in a larger CV system. This implies that when the variability of demand is higher, the capacity pooling effect of LTW design has more value in dealing with demand fluctuations. However, when CV is higher, the performance difference between design L_n^{n-1} and L_n^n is more significant. In the case of CV $= 0.1$, whether closing the loop or not does not make much difference, while in the case of CV $= 0.5$, the performance difference is greater than 1% for $q \geq 0.3$. Even though, design L_n^{n-1} has relatively good performance and is an effective design in increasing capacity utilization.

2.5.2 The Value of Closing the Loop

In Table 2.3, we present more simulation results to demonstrate the effectiveness of limited flexible design L_n^{n-1}. We again use the demand setting in Jordan and Graves (1995), which

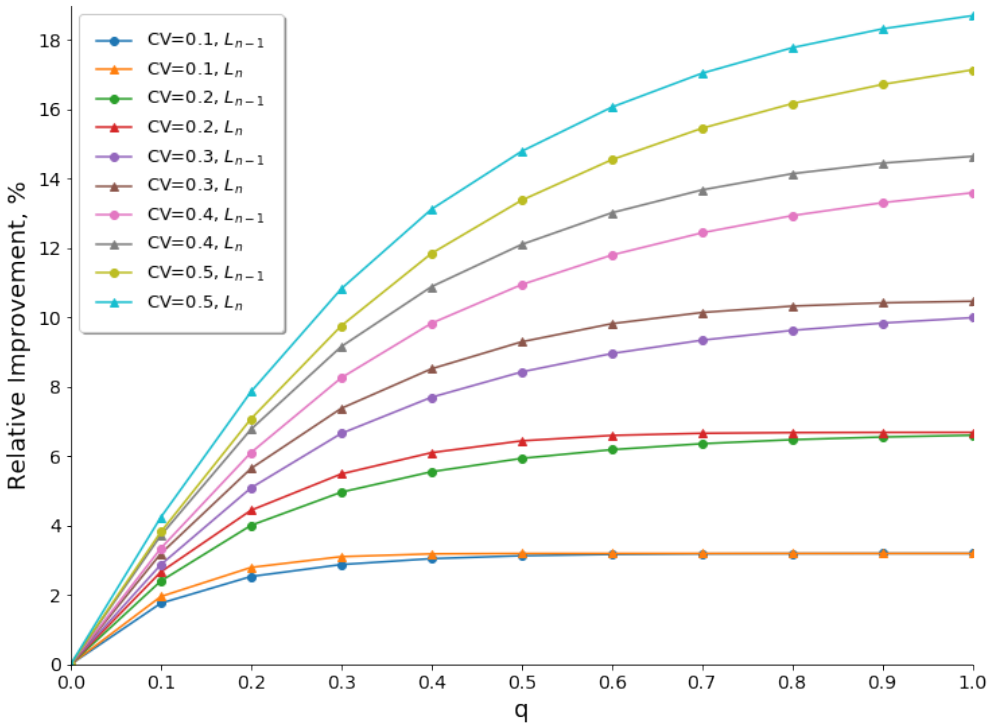


Figure 2.9: Performance of L_n^{n-1} and L_n^n with differenct CV and q
Note: $n = 10$, X_t follows Uniform distribution with mean 100

is i.i.d. truncated ($\pm 2\sigma$) normal distribution with mean $\mu = 100$ and standard deviation $\sigma = 40$. The capacity is equal to μ. We try 3 n values and 5 q values and compute the relative improvement $\frac{F(L_n^{n-1},q)-F(L_n^0,q)}{F(L_n^0,q)}$ and $\frac{F(L_n^n,q)-F(L_n^0,q)}{F(L_n^0,q)}$. 20% of consumers willing to choose LTWs can provide nearly a half of the benefit comparing with the case of $q = 100\%$. Also, for all q values and n values, the performance of design L_n^{n-1} and L_n^n are close to each other, both guarantee significant improvements. We can also observe that when n increases, the benefit increases to a small degree.

To evaluate the effectiveness of capacity pooling, we compute the performance of the fully flexible system. In the experiment with $n = 8$, the fully flexible design provides 11.06% of relative improvement. When $n = 16$, 12.81% and when $n = 24$, the maximum improvement is 13.55%. From Table 2.3 we can see that the proportion of maximum improvement

Table 2.3: Relative improvement comparing with L_n^0

q	$n = 8$		$n = 16$		$n = 24$	
	L_n^{n-1}	L_n^n	L_n^{n-1}	L_n^n	L_n^{n-1}	L_n^n
20%	5.07	5.78	5.43	5.79	5.54	5.78
40%	7.79	8.88	8.36	8.92	8.54	8.91
60%	9.21	10.33	9.92	10.62	10.14	10.61
80%	9.98	10.87	10.84	11.62	11.10	11.66
100%	10.42	11.02	11.41	12.20	11.71	12.34

Note: Demand follows i.i.d. truncated ($\pm 2\sigma$) normal distribution with $\mu = 100$ and standard $\sigma = 40$.

captured by design L_n^{n-1} is already very high even for small q values. This demonstrates the effectiveness of limited flexible design with limited flexible levels.

2.5.3 Increasing Flexible Level vs. Increasing Capacity

In order to increase the number of demand that can be fulfilled, the firm can either seek for consumer flexibility or simply increase the capacity levels. We design a simulation experiment that targets a 5% improvement in the expected number of fulfilled demand and search for the required flexible level q while keeping the capacity the same, and search for the required capacity increment with no LTW added. We simulate a system with 8 regular time windows and i.i.d. truncated ($\pm 2\sigma$) normal distribution distribution with mean 200 and standard deviation from 40 to 100. The LTW design we choose here is the open chain L_n^{n-1} with $q = 0.2$ since the effectiveness of limited flexibility has been demonstrated by previous results. The required flexible level and capacity increments are summarized in Table 2.4.

When the variability of demand is relatively small, e.g., when the standard deviation is 40, the flexible level requirement is as high as 72%. In a service system, it may be hard to attract such high proportion of customers to choose a LTW. Indeed, in this case, the total

Table 2.4: The required q and capacity increments to achieve a 5% improvement

Demand	q	C
$N(200, 40)$	72%	25%
$N(200, 60)$	26%	10.5%
$N(200, 80)$	20%	9.5%
$N(200, 100)$	17%	9%

potential benefit from capacity pooling just exceeds 5%. However, with higher variability, capacity pooling shows its power. When the standard deviation is 100, with only 17% of customers being flexible, the improvement target can be achieved, while a 9% capacity increase is required, which can be considered as 9% increase in the cost of services. Given the fact that the discount provided to LTWs are usually much smaller than the labor cost and profit from completing a service, it may be more beneficial in terms of total profit to utilize the capacity pooling effect through LTWs.

2.5.4 Unbalanced Systems

In this subsection, we numerically show the effectiveness of adding LTWs when demands are not identical across the day. It is common that a service platform faces one or two demand peaks in the middle of the day.

First, we presents an simulation example based on a real data set from a service platform. The service time window in a day is composed of 9 non-overlapping regular time windows. The average demand in each regular time window are $41, 60, 57, 48, 35, 33, 50, 43, 26$ units, respectively. This example represents the typical demand pattern which has one peak in the morning and one peak in the afternoon. We assume that demands are independent truncated (1 to 103) normally distributed random variables with a standard deviation equal

to 50% of expected demand. We consider all large time window designs L_n^i for $i = 0, \ldots, 9$ and various flexible level q between 0 and 1. With these data, we simulate 100000 days of demand realization and evaluate the expected number of fulfilled demand for different time window designs and the results are summarized in Figure 2.10.

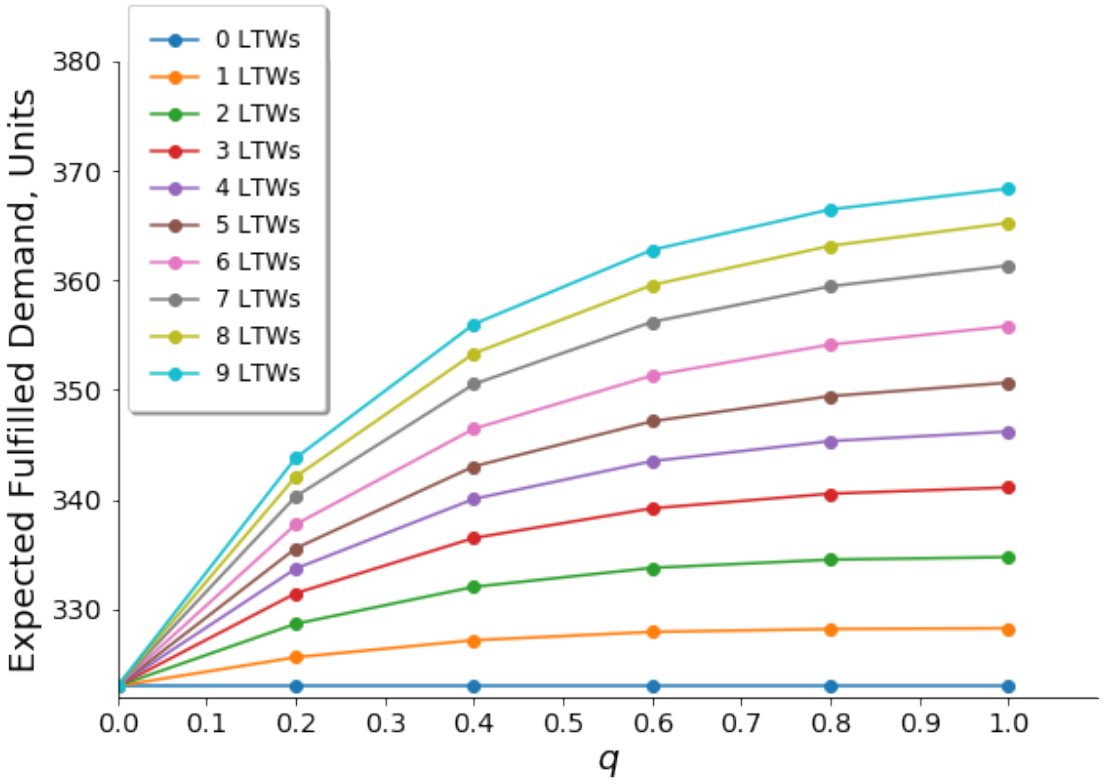


Figure 2.10: Diminishing return to increased flexible level with unbalanced demand

Figure 2.10 shows the diminishing return in expected fulfilled demand of adding flexible level. We can observe the concavity behavior when increasing the q value and it is consistent among all large time window designs. For a fixed q value, the vertical distance between two consecutive curves represents the benefit of adding an extra LTW. From Figure 2.10 we can observe that adding the 9th LTW which closes the loop creates relatively small benefit comparing with the benefit of adding other LTWs.

Next, to further validate the effectiveness of limited flexible LTW design, we design

Table 2.5: Percentage of improvement captured by limited flexible designs v.s. fully flexible system

Capacity	Average Performance				Worst-case Performance			
	$q = 20\%$		$q = 100\%$		$q = 20\%$		$q = 100\%$	
	L_n^{n-1}	L_n^n	L_n^{n-1}	L_n^n	L_n^{n-1}	L_n^n	L_n^{n-1}	L_n^n
Ave. Demand	40.3%	45.4%	85.7%	95.0%	33.0%	37.2%	74.7%	86.0%
70% SL	54.6%	61.1%	94.3%	99.0%	45.4%	51.9%	85.0%	95.6%
80% SL	69.4%	76.48%	98.2%	99.8%	57.6%	64.2%	93.3%	98.7%
90% SL	85.2%	90.6%	99.8%	99.996%	74.3%	80.2%	96.3%	99.8%

a simulation experiment with more general settings. We again use the real data set as a guidance for parameter choices. We fix $n = 9$ and use independent truncated normal distributions. For the demand distribution, we randomly generated 500 set of means and standard deviations (each set contains the parameter for 9 regular time windows), where the mean of a regular time window is uniformly generated between 20 and 80 and the coefficient of variation is randomly generated between 0.2 and 0.8. For each set of randomly generated means and standard deviations, we evaluate the performance of design L_n^{n-1} and L_n^n by averaging the number of fulfilled demands from 10000 simulated days. We compare the improvement with the maximum potential benefit of capacity pooling, which is the so-called fully flexible system. In a fully flexible system, any demands can be fulfilled in any time window. Therefore, the performance difference between the fully flexible system and the no flexible system is the maximum potential benefit of capacity pooling. We compute the proportion of maximum potential benefit captured by design L_n^{n-1} and L_n^n and list in Table 2.5. Besides the average performance, we also compute the worst-case performance among the 500 different demand distributions. We try different capacity levels . In reality, capacity levels are often determined by service level targets and we set the capacity level to meet the $70\%, 80\%$ and 90% service level constraints.

Table 2.5 shows the proportion of maximum benefit captured by design L_n^{n-1} and L_n^n. The performance gets better when the capacity level increases. However, when service level target is high, the total potential benefit from capacity pooling is indeed very small. Nevertheless, limited flexible design with small q value can capture a large proportion of the total potential benefit, both on average and in worst case.

2.5.5 Where to Add LTWs in Unbalanced Systems

In this subsection, we design 4 demand patterns and seek to answer the question that which LTWs are more important than others in unbalanced systems. We consider a system with 6 regular time windows and 5 candidate large time windows 1&2, 2&3, 3&4, 4&5 and 5&6. Demands are independent truncated ($\pm 2\sigma$) normal distribution distribution with $CV = 0.5$. Typically there will be one or two demand peaks in a day and the demand rate in the begining and end of the day would be relatively lower. Therefore, we design 4 demand patterns:

- Two peak 1:$[100, 180, 120, 200, 120, 100]$,
- Two peak 2: $[100, 180, 120, 80, 200, 100]$,
- One peak 1: $[100, 120, 200, 140, 120, 80]$,
- One peak 2: $[80, 120, 180, 180, 120, 100]$.

When only i LTWs are allowed in the design, we find the best combination among all 5 choose i possible set of LTWs that guarantees the most improvements. The results are shown in Table 2.6. The best LTWs tends to connect regular time windows that have the most variability in total. Therefore, the highest gain tends to appear around the peak demand by

Table 2.6: The best design with certain number of LTWs

Demand	1 LTW	2 LTWs	3 LTWS	4 LTWs
Two peak 1	3&4	3&4, 4&5	2&3, 3&4, 4&5	1&2, 2&3, 3&4, 4&5
Two peak 2	2&3	2&3, 5&6	1&2, 2&3, 5&6	1&2, 2&3, 4&5, 5&6
One peak 1	3&4	2&3, 3&4	2&3, 3&4, 4&5	1&2, 2&3, 3&4, 4&5
One peak 2	3&4	2&3, 3&4	2&3, 3&4, 4&5	2&3, 3&4, 4&5, 5&6

our design. Note that in demand pattern "Two peak 2", the two demand peaks are rather far away. Therefore, the best 2 LTWs, 3 LTWs and 4 LTWs are not a connected chain. In contrast, other demand patterns has either one peak or two peaks that are close enough, and the best design with 4 LTWs forms an open chain.

2.5.6 Dynamic Capacity Allocation

In many applications, the capacity will be dynamically allocated upon customers' arrival. After the capacity in a time window is sold out, that time window will be no longer available to customers and some customers may be lost due to this reason. In this situation, adopting LTWs can delay the final allocation for the flexible customers and help balance demands. The firm offers time window design L_{n-1}, where the choice of a regular time window refers to an immediate commitment between the customer and the firm, and the choice of a large time window refers to a delayed decision. The final decision will be sent to the customer after the selling horizon (usually before the service date). Our benchmark is the no-discount strategy. The firm only provides n regular time windows with same prices and does not offer discounts.

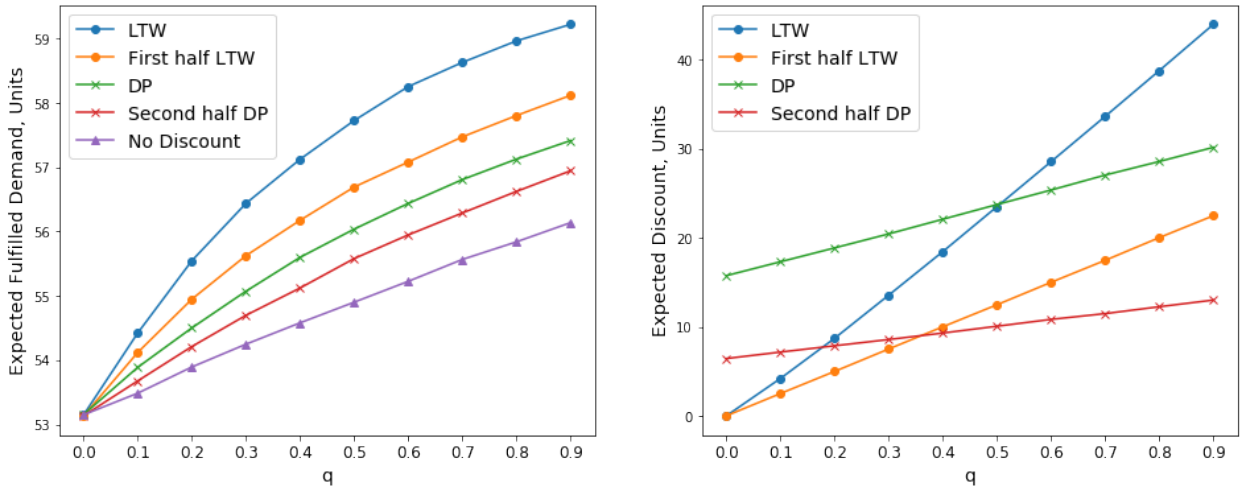
Another common strategy to balance demands is to dynamically offer discount to some time windows. Specifically, we analyze a strategy that only offer one discount level and only

to the time windows that have the highest capacity level left. If all time windows have the same capacity level, then no discount is needed.

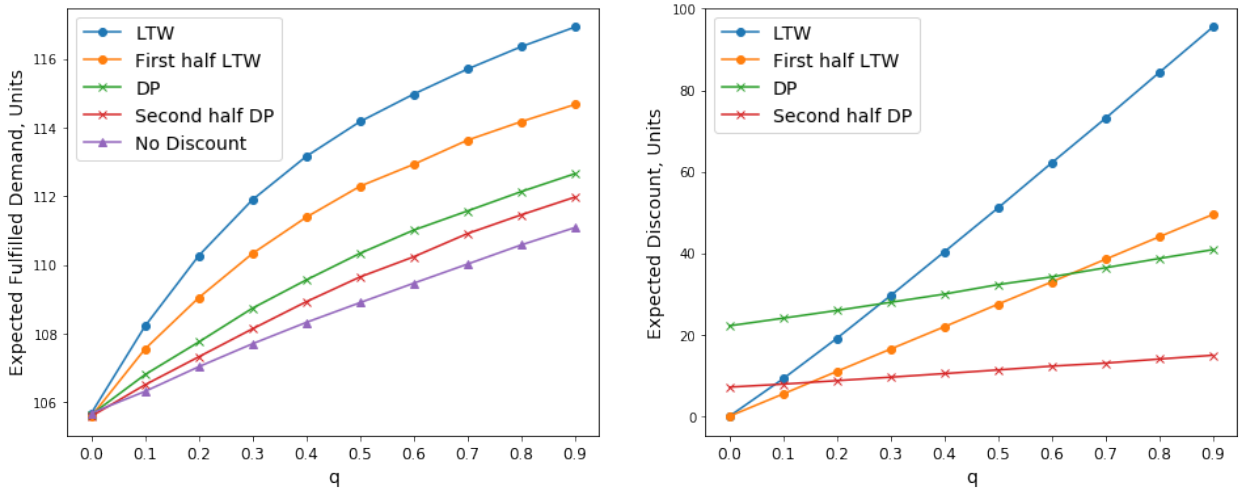
We assume the customers are either flexible or not. A flexible customer has a first choice and a second choice. A non-flexible customer only has one preferred time window. In the no-discount strategy, a flexible customer will choose the second choice only if the first choice is sold out and the second choice is available. In the dynamic pricing strategy, a flexible customer will choose the second choice in two cases: i) if the second choice has lower price than the first choice, and ii) if the first choice is sold out. In the LTW strategy, flexible customers will choose the LTW if available. Note that a LTW will be offered if and only if both regular time windows correspond to the LTW is available.

We want to compare the number of customers the firm can serve in a finite horizon under three different strategies. In order to make a fair comparison, we focus on three strategies with the same proportion of flexible customers and with the same selling horizon $T = nC$. We also count the number of discounts that are eventually given out to the customers. Can some strategy boost the capacity utilization with less number of discounts offered? In Figure 2.11, we plot the results for two scenarios $n = 6$ and $n = 12$. Besides offering LTWs throughout the selling horizon, we also try a strategy that offers LTWs in the first half of the selling horizon and then closes the LTW options. Also for the dynamic pricing strategy, we try a strategy that only offers discounts in the second half of the selling horizon. We aim to see the trade-off between the improvement in the expected fulfilled demand and the number of discount offered.

First, we can notice that just offering LTW in the first half of the selling horizon can achieve more than half of the improvement in the expected fulfilled demands. This implies



(a) $n = 6, C = 10$



(b) $n = 12, C = 10$

Figure 2.11: Large Time Windows v.s. Dynamic Pricing

that the LTW strategy at the end of the selling horizon is not as effective as the beginning of the selling horizon. On the other hand, offering LTW in the first half of the selling horizon only gives out half number of discounts. It suggests that the firm should always consider offering LTWs at the beginning of the selling horizon, not wait until the demand unbalance appears.

Second, it is clear that with the same percentage of customers being flexible, the LTW strategy provides larger benefit in the expected fulfilled demands than the DP strategy.

When the fraction of flexible customers is small, the LTW strategy gives less number of discounts. It means that the LTW strategy may achieve better improvement with less discounts. As q increases, the number of discounts offered to the customers increases. After some threshold, the LTW strategy will apply much more discounts than applying the DP strategy. However, typically the fraction of flexible customers would not be very large.

Even just providing LTWs in the first half of the selling horizon, the performance is apparently better than dynamic pricing. Note that in the $n = 6$ case, the number of discounts offered in the LTW strategy in the first half selling horizon is always lower than that in the dynamic pricing strategy, where the improvement is always higher.

2.6 Online Grocery Delivery with LTWs

In previous sections, we characterize the value of LTWs in scheduled service systems with constant capacity. Large time window design also appears in the online grocery delivery platforms where our model can not perfectly fit in. In this section, we numerically examine the performance of LTWs in an online grocery delivery problem.

We assume that customers arrive sequentially and select a time window from a list of time windows. The firm accepts customers until a specific cut-off time that is before the day of delivery. In this experiment, we stop accepting new customers after T customer arrivals. The consumer acceptance process is almost identical to our dynamic capacity allocation experiment, except that the firm does not have a constant capacity in each regular time window. Instead, the availability is based on the overall utilization of the total available travel time for a time window in a service region.

Specifically, we assume that a service region is a 20 by 20 block where the customers are located uniformly at random in every intersection. There is a depot located at the center of the service region. In each regular time window, a truck departs from the depot and travels through all the locations of the accepted requests and finally returns to the depot. Assume the truck can only travel along the streets and therefore the distance between two locations is the Manhattan distance. We compute the shortest route by solving a traveling sales man problem. We assume a constant travel time of 1 minute per block and the time span of a regular time window is 60 minutes.

We make the same assumption on consumers' behavior as the experiment on dynamic capacity allocation. Consumers' preferences are symmetric among all time windows. The customers are either flexible or not. A flexible customer has a first choice and a second choice. A non-flexible customer only has one preferred time window. We want to examine the number of demand accepted with LTWs in comparison with a no flexibility setting. When there is no LTW offered to the customer, a flexible customer will choose the second choice only if the first choice is not available while the second choice is available. A LTW will be offered to a specific customer if and only if both regular time windows correspond to the LTW is available to that customer. The offering of a particular time window to a customer is only allowed if the travel time among all accepted customers can still be guaranteed within the length of the time window.

Due to the complexity in computation, we focus on the non-overlapping LTW design N_n. The performance metric is still the number of customers that can be accepted by the system. We also compute the expected fill rate, which is the percentage of demand that can be fulfilled. The benchmark strategy only offers regular time windows and does not

adjust prices through out the selling horizon. We report the numerical results in Table 2.7 and Appendix 2.8.3 by randomly generate 1000 demand scenarios, each contains T customer arrivals.

Table 2.7: Performance of adding two non-overlapping LTWs when $N = 4$

q	Fill Rate	Relative Improvement	Percentage of Scenario	
			Performs better	Performs worse
0.2	83.3%	1.3%	26.1%	3.8%
0.4	86.4%	2.7%	46.9%	4.6%
0.6	89.3%	3.3%	54.3%	9.3%
0.8	92.6%	4.7%	67.6%	6.5%

Note: The fill rate of the benchmark policy with $q = 0$ is 80.32%. $T = 30$.

In Table 2.7, we present the performance of various q levels, from 0.2 to 0.8. We compute the average fill rate, the relative improvement comparing with the benchmark policy with the same q value, as well as the percentage of demand scenarios where adding LTWs performs strictly better than the benchmark policy, and the percentage of demand scenarios where adding LTWs performs strictly worse.

When offering large time windows, although the average fill rate increases, we do observe some demand sequences where adding LTWs performs worse then the benchmark policy. After investigation on this counter-intuitive behavior, we find that it often happens due to a demand located near the boundary of the service region. The acceptance of such requests can greatly restrict the ability to accommodate future requests. While the benchmark policy usually can not accept such demands, the system with LTWs may accept a new customer near the boundary due to the flexibility provided by other customers, which will highly restrict the potential of accepting future demands. However, this does not mean that flexible design can ruin the performance. First of all, in the experiments, we observe that the percentage

of scenarios in which adding LTWs performs worse is much lower than the chance that it performs strictly better. Secondly, if LTW design performs worse more often, it simply suggests that the service region is too large for the travel time constraint such that locations near the boundary should not be accepted at all. Therefore, the firm should reconsider the design of service region and delivery time constraint.

Soeffker et al. (2017) study the issue of fairness with regards to the customer accepting and pricing mechanisms in vehicle routing problems. They show that a policy that can decline certain customer requests, which under serves areas distant from the depot, can overall accept more requests. In our experiments, flexibility increases the possibility a customer at the boundary of the service region to be accepted. It implies that consumer flexibility can improve fairness without sacrificing on the ability to serve more customers on average. Generally, the relationship between flexibility and fairness can be an interesting future direction.

2.7 Discussion

In this paper, we model and evaluate the value of introducing consumer flexibility into scheduled service systems through the design of large time windows. We demonstrate the power of limited flexibility in this context. On one hand, we theoretically and numerically show the diminishing return in increased flexible level q. On the other hand, we connect the LTW design with the long chain design in the process flexibility literature and construct a long chain that is equivalent to the performance of an extreme case of the LTW design. As the performance of long chain design has been thoroughly studied in the literature,

including average performance, asymptotic performance and robustness, we can conclude that the LTW design we proposed which combines two consecutive regular time windows is effective in capacity utilization improvement. Therefore, we demonstrate that limited flexible design with limited flexible level can capture most of the capacity pooling benefits.

We also find the distinctions between the LTW design and long chain design due to the intrinsic differences in the fundamental mathematical model. The open chain L_n^{n-1} performances surprisingly good and closing the loop does not provide much value here. Through numerical studies, we find that to better utilize the LTW design, the firm should prioritize the coverage of all regular time windows in a balanced system, and the coverage of high variability time windows in an unbalanced system.

2.8 Additional Proofs and Results

2.8.1 Additional Proofs

Proof of Lemma 2.3.1. Note that the necessary and sufficient condition for a positive increment to occur when adding the LTW is $(x_1 - C)(x_2 - C) < 0$, i.e. demand in one regular time window exceeds C while the other one does not reach the capacity level. The benefit of adding a LTW can be written as the value of the flow on the augmenting paths when adding edges $(w_{1\&2}^1, c_2)$ and $(w_{1\&2}^2, c_1)$.

An augmenting path must include edge $(w_{1\&2}^1, c_2)$ or $(w_{1\&2}^2, c_1)$. The augmenting flow on edge $(w_{1\&2}^1, c_2)$ can be positive only when $x_1 > C$ and $x_2 < C$. Similarly, the augmenting flow on edge $(w_{1\&2}^2, c_1)$ can be positive only when $x_1 < C$ and $x_2 > C$. These two events

can not happen at the same time. Therefore, the augmenting flow on edge $(w^1_{1\&2}, c_2)$ and $(w^2_{1\&2}, c_1)$ can not be positive at the same time. Thus, the benefit of adding the LTW can be written as the benefit of only adding edge $(w^1_{1\&2}, c_2)$, plus the benefit of only adding edge $(w^2_{1\&2}, c_1)$. Next note that we assume demand is symmetric. Thus the expected benefit of only adding edge $(w^1_{1\&2}, c_2)$ is equal to the expected benefit of only adding edge $(w^2_{1\&2}, c_1)$.

We can conclude that the benefit of adding LTW 1&2 is equal to

$$2\left[\mathbb{E}[M(X_1 - Y_1, Y_1, X_2)] - F(L_2^0)\right],$$

which completes the proof. □

Proof of Theorem 2.3.2. We construct a discrete distribution with $n = 3$ that satisfy the inequality in Theorem 2.3.2.

Let X_i follows a discrete distribution on two values l, h with equal probability, $l < h$ and $C = (l + h)/2$. Assume $q = 1$. The goal is to quantify the benefit of adding each LTW when $n = 3$ and show that adding LTW 1&3 is much smaller than the other two LTWs.

The demand X_1, X_2, X_3 has 8 possible realizations and for (l, l, l) and (h, h, h), all the system have the same performance. The performance of other scenarios are summarized in Table 2.8.

In Table 2.8, a and b are defined as the following. Let $Z \overset{d}{=} Y \overset{d}{=} Bin(h, 0.5)$, Z, Y independent.

$$a = \mathbb{P}[Z < \frac{h-l}{2}]\mathbb{E}[Z|Z < \frac{h-l}{2}] + \mathbb{P}[Z \geq \frac{h-l}{2}]\frac{h-l}{2} \leq \frac{h-l}{2}, \tag{2.5}$$

$$b = \mathbb{P}[Y + Z < \frac{h-l}{2}]\mathbb{E}[Y + Z|Y + Z < \frac{h-l}{2}] + \mathbb{P}[Y + Z \geq \frac{h-l}{2}]\frac{h-l}{2} \leq \frac{h-l}{2}. \tag{2.6}$$

Table 2.8: $P(d; L_n^i)$ for different realizations d

	L_n^0	L_n^1	L_n^2	L_n^3
(l, l, h)	$2l + C$	$2l + C$	$2l + C + a$	$2l + h$
(h, h, l)	$2C + l$	$2C + l$	$2C + l + a$	$2C + l + b$
(l, h, l)	$2l + C$	$2l + C + a$	$2l + h$	$2l + h$
(l, h, h)	$2C + l$	$2C + l + a$	$2C + l + a$	$3C$
(h, l, l)	$2l + C$	$2l + C + a$	$2l + C + a$	$2l + h$
(h, l, h)	$2C + l$	$2C + l + a$	$2C + l + b$	$2C + l + b$

We first provide a lower bound on a. Let $l = \alpha h$, and use the tail bound of binomial distribution, we can get

$$\begin{aligned}\mathbb{P}[Z < \frac{h-l}{2}] &\leq \exp(-2(\frac{(\frac{h}{2} - \frac{h-l}{2})^2}{h})) \\ &\leq \exp(-\frac{l^2}{2h}) \\ &= \exp(-\frac{\alpha^2}{2}h).\end{aligned}$$

Then a can be lower bounded by

$$\begin{aligned}a &= \mathbb{P}[Z < \frac{h-l}{2}]\mathbb{E}[Z|Z < \frac{h-l}{2}] + \mathbb{P}[Z \geq \frac{h-l}{2}]\frac{h-l}{2} \\ &\geq \frac{h-l}{2}\mathbb{P}[Z \geq \frac{h-l}{2}] \\ &\geq \frac{h-l}{2}(1 - \exp(-\frac{\alpha^2}{2}h)).\end{aligned}$$

Thus, the benefit of adding the first and second LTW can be written as

$$\begin{aligned}F(L_3^1, 1) - F(L_3^0) &= \frac{4}{8}a = a/2, \\ F(L_3^2, 1) - F(L_3^1, 1) &\geq \frac{2}{8}a = a/4.\end{aligned}$$

Next we compute the benefit of adding the third LTW, which is also the last LTW when $n = 3$. From Table 2.8, we have

$$\begin{aligned} F(L_3^3) - F(L_3^2) &\leq \frac{4}{8}(\frac{h-l}{2} - a) \\ &= \frac{1}{2}\frac{h-l}{2}\exp(-\frac{\alpha^2}{2}h) \end{aligned}$$

Then we can compute the ratio between the benefit of adding the last LTW and the benefit of adding the first (second) LTW.

$$\frac{F(L_3^3) - F(L_3^2)}{F(L_3^1, 1) - F(L_3^0)} \leq \frac{F(L_3^3) - F(L_3^2)}{F(L_3^2, 1) - F(L_3^1, 1)} \leq \frac{h-l}{2a}\exp(-\frac{\alpha^2}{2}h) \leq \frac{\exp(-\frac{\alpha^2}{2}h)}{1 - \exp(-\frac{\alpha^2}{2}h)}.$$

For any l, h satisfying $\frac{l^2}{h} \geq 2\log\left(\frac{1+\epsilon}{\epsilon}\right)$, we have $\exp(-\frac{\alpha^2}{2}h) \leq \frac{\epsilon}{1+\epsilon}$. Therefore, we have

$$\frac{\exp(-\frac{\alpha^2}{2}h)}{1 - \exp(-\frac{\alpha^2}{2}h)} \leq \epsilon,$$

which completes the proof. □

2.8.2 Explanation of why supermodularity does not hold

Jordan and Graves (1995) observe that the marginal benefit increases as the long chain is constructed, and the largest benefit is always achieved when the chain is closed by adding the tenth arc to the system. This important insight was later verified in Simchi-Levi and Wei (2012) by proving a fundamental property of long chains, supermodularity. Here we provide a theoretical explanation of why the proof of supermodularity in long chain design (Simchi-Levi and Wei (2012)) fails in our setting.

The proof in Simchi-Levi and Wei (2012) is based on two facts: i) the max flow problem can be formulated as a max circulation problem, ii) all the arcs added in long chain are in series. Then the supermodularity result directly follows from the main theorem in Gale and Politof (1981). In Figure 2.12, the dashed lines represent the arcs added, which are the flexible links. The proof fails for the time window design problem because of the fact that not all the arcs added are in series. In the time window design problem, adding a LTW $t\&t+1$ is equivalent to adding an arc from the demands initially pointing to t, but are willing to extend to $t\&t+1$, which we denote as ξ_t^3 in Section 2.2, and at the same time, adding an arc from the demands initially pointing to $t+1$, but are willing to extend to $t\&t+1$, which we denote as ξ_{t+1}^2. In Figure 2.12, we use $\alpha_t, \beta_t, \gamma_t$ to denote the demand node for ξ_t^2, ξ_t^1 and ξ_t^3, respectively. We can note that arc (γ_1, c_2) and (α_3, c_2) have the same head. By definition, two arcs that share the same head or tail are in parallel. According to Gale and Politof (1981), the max flow is submodular in arcs that are in parallel. Therefore, although there exists some arcs that are in series in Figure 2.12, which may provide increasing benefit, adding LTWs overall can not guarantee increasing marginal benefits.

2.8.3 More Experiments in the Online Grocery Delivery Setting

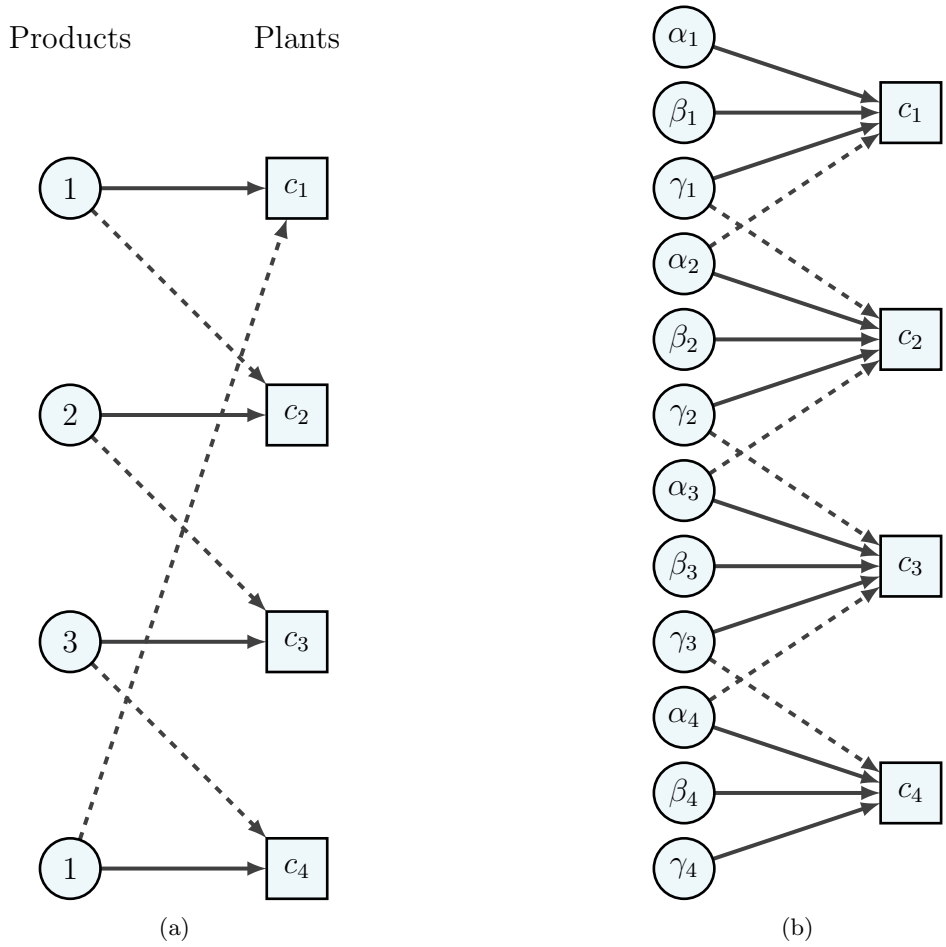


Figure 2.12: Comparison the arcs added in long chain design and LTW design

Table 2.9: Performance of Adding a LTW when $N = 2$

q	Fill Rate	Relative Improvement	Percentage of Scenario	
			Performs better	Performs worse
0.2	85.44%	2.78%	27.70%	3.20%
0.4	89.99%	5.68%	50.00%	3.50%
0.6	94.53%	7.77%	64.70%	1.30%
0.8	98.31%	9.37%	73.30%	0.50%

Note: The fill rate of the benchmark policy with $q = 0$ is 81.05%. $T = 15$.

Table 2.10: Performance of Adding three non-overlapping LTWs when $N = 6$

q	Fill Rate	Relative Improvement	Percentage of Scenario	
			Performs better	Performs worse
0.2	83.26%	1.31%	37.20%	5.60%
0.4	86.44%	2.68%	58.90%	8.10%
0.6	89.24%	3.37%	69.90%	7.80%
0.8	92.43%	4.17%	76.60%	7.50%

Note: The fill rate of the benchmark policy with $q = 0$ is 80.04%. $T = 45$.

Chapter 3

Queuing Safely for Elevator Systems amidst a Pandemic

The requirement of social distancing during the COVID-19 pandemic has presented significant challenges for high-rise buildings, which heavily rely on elevators for vertical transportation. In particular, the need for social distancing has reduced elevator capacity by at least half and as much as two-thirds the normal amount. This reduction is a serious concern, as reduced elevator capacities cause large queues to build up in lobbies, which makes social distancing a challenge. The objective of this study is to safely manage the elevator queues by proposing simple, technology-free interventions that drastically reduce the waiting time and length of lobby queues. We use mathematical modeling, epidemiological principles, and simulation to design and evaluate our interventions. The key idea is to explicitly or implicitly group passengers that are going to the same floor into the same elevator as much as possible. In the *Cohorting* intervention, we attempt to find passengers going to the same floor as the first person in the queue. In the *Queue Splitting* intervention, we create a different queue for different groups of floors. Based on simulation and analytical studies, Cohorting and Queue Splitting can significantly reduce queue length and wait time, while also maintaining safety from viral transmission in otherwise crowded elevators, building lobbies, and entrances. The interventions we propose do not require programming the elevators, and rely on only using

signage and/or a queue manager to guide passengers.

3.1 Introduction

The COVID-19 pandemic has made it imperative to design interventions for people to stay safe in potentially crowded areas. For high-rise buildings, social distancing reduces the capacity of elevators, cutting the number of passengers per elevator by at least half and as much as two-thirds (Weber 2020). Reduced elevator capacity can cause large lobby queues and long wait times, resulting in crowding and reduced social distancing (Weber 2020, Wilson 2020, Smith 2020, van Rijn et al. 2020). With no interventions and reduced capacity on elevators, the increased waiting times and queue lengths in the lobby could pose significant safety risks. Thus, an intervention to the public health problem of safely managing queues for elevator systems amidst a pandemic is needed. *This project was directly requested by the NYC Mayor's Office, which had continuous input into our work throughout the process.*

One can broadly consider two major forms of interventions, based on (i) changing passenger behavior and (ii) elevator artificial intelligence. A variety of technological innovations from elevator companies and building management have been considered during the pandemic (Wilson 2020). In many elevator systems, especially in older ones, changing the algorithms and technology of how the elevators navigate through the building is challenging, and would require long-term planning and expensive modifications. Thus, in this work (in collaboration with an epidemiologist) we focus on *technology-free* interventions that safely manage how passengers use and board elevators, which should be more accessible and practical for an overwhelming majority of buildings with elevators.

Currently, many elevator systems take a hands-off approach to managing the flow of people to elevators, resulting in something that resembles first-come first-serve (Fujino et al. 1997). Our simulations, using data calibrated from a large government New York City building, show that such a hands-off approach will lead to large and unsafe queues if building occupancy returns to pre-pandemic levels while elevator capacities are still reduced due to social distancing. Thus, it is imperative that we design interventions that use the elevators more efficiently – getting passengers to their destination at a faster aggregate rate (higher throughput) – by more carefully managing who uses which elevator when. For instance, we shall consider interventions where we try to get passengers going to the same floor to cohort and ride an elevator together as well as interventions where passengers are encouraged to walk up or down a floor after riding the elevator. Although researchers have studied algorithms for managing elevator systems (Barney and Dos Santos 1975, Barney and Al-Sharif 2015, Lee et al. 2009, Fujino et al. 1997, Pepyne and Cassandras 1997, Al-Sharif et al. 2012), to the best of our knowledge, there is not much literature on designing elevator systems with pandemic safety considerations.

Our simulation model is based on a discipline of applied mathematics known as queuing theory. Previous work that utilizes queuing theory in elevator systems rely on assumptions that may not hold in practice (Alexandris 1977, Barney and Al-Sharif 2015, Finschi 2010), thus we utilize a detailed simulation to estimate mean wait times and queue lengths. A discrete event simulation (Ross 2013) models the operation of a system as a sequence of events in time. Simulation has previously been used in elevator traffic studies (Al Sukkar et al. 2017, Hakonen and Siikonen 2008). For example, a recent paper in the context of the COVID-19 pandemic (Swinarski 2020) models and predicts elevator traffic in an university

classroom building when passengers mainly travel in the short time period between two classes. They discuss an intervention of pairing passengers going to same destinations.

We propose a more general model with many other interventions, along with open source code. The social distancing requirement during a pandemic may lead to unsafe queues in the lobby with no interventions. Using mathematical modeling, epidemiological principles, and simulation we design and evaluate simple interventions to load passengers in elevators that can drastically reduce the length of lobby queues amidst a pandemic. The proposed interventions increase efficiency of the elevator system, and are effective beyond the constraints imposed by a pandemic. Therefore, the interventions are useful even after the pandemic to manage lobby queues. Our interventions do not require programming the elevators, and rely on using only signage and/or a queue manager (QM) to guide passengers. We propose an intervention we call *Cohorting*, which we attempt to find passengers going to the same floor as the first person in the queue. Simulations show Cohorting reduces waiting time for passengers and the number of people in the lobby (queue length) significantly. In limited lobby spaces, we recommend the *Cohorting with Pairing* intervention, where we pair passengers going to the same floors. With less communication from a QM, we also propose the *Queue Splitting* intervention where we create a different queue for different groups of floors. Queue Splitting even with a small number of floors achieves comparable performance to Cohorting. Using data calibrated from a large government building in New York City that is planning for re-opening and is in need of managing elevator traffic amidst the COVID-19 pandemic, we perform simulations of these interventions, explore the impact of some passengers willing to walk up or down one floor from their destination, as well as examine considerations in applying the proposed interventions in practice.

We also analytically investigate the reason behind the good performance of Cohorting and Queue Splitting using a technique from queueing theory known as stability analysis. Specifically, we characterize the system parameters required for each intervention to ensure that the queues do not increase in length over time, i.e., the queues are stable. Our theoretical analysis reveals that these interventions can effectively reduce the average distance traveled and the number of stops for the elevator trips.

The paper is organized as the following. We introduce the simulation model of the elevator system in Section 3.2 and present the simulation results in Section 3.3. Section 3.4 analyzes the stability condition for each intervention we propose and compares the results to FCFS. In Section 3.5, we discuss potential the practical issues and solutions for the Cohorting intervention. Finally, we conclude and discuss ideas for future work in Section 3.6.

3.2 Simulation Model

In this section, we describe our modeling framework. In particular, the model considers moving passengers upwards through a building from a lobby, which presents the biggest challenge for social distancing in a high-rise building. We study low-tech solutions (requiring no programming of elevators and no knowledge of internal elevator algorithms) and describe interventions to manage the queue of passengers in the lobby. We focus on analyzing solutions that work for high volume periods, e.g. morning rush hour, lunchtime, etc. where social distancing is a challenge. These busy periods are referred to as *uppeak* (Barney and Dos Santos 1975) and typically an elevator system working efficiently during the morning uppeak can handle interfloor traffic and downpeaks without any issues (Barney

and Al-Sharif 2015). Below we describe the model we use in the simulation. We simplify some of the assumptions when deriving the analytical results in Section 3.4.

We model a building as having m floors denoted $1, \ldots, m$ and N elevators denoted $1, \ldots, N$. We assume passengers wanting to go to floor j at time t arrive according to a non-stationary Poisson process with arrival rate $\lambda_j(t)$. The Poisson assumption for individual arrivals is considered a good approximation to the arrival process (Barney and Al-Sharif 2015, Alexandris 1977). Each of the N elevators have a capacity of C, the number of people that the elevator can safely transport while ensuring social distancing. In many high-rise buildings, elevators are constrained to certain floors so we let $S(n)$ denote the floors that elevator n can serve. If there is no restriction on the service range of the elevator n, then $S(n) = \{1, 2, \ldots, m\}$. We assume the elevator speed ν is constant and the (de)boarding times of the elevator is a function of the number of people k that are (de)boarding, denoted by $BoardingTime(k)$. The (de)boarding time $BoardingTime(k)$ is a constant time ω to open and close the elevator door, and additional time depending on the number of passengers k entering (exiting). The travel time to start at floor j_1 and stop at floor j_2 is $T(j_1; j_2) = \nu(j_2 - j_1)$. If k ($\leq C$) passengers with destinations $d_1 \leq d_2 \ldots \leq d_k$ board an elevator n at the lobby, we can create a count $\vec{F} = \{F_2, \ldots, F_m\}$ of the number of passengers deboarding at each floor. Note that $\sum_{j=2}^{m} F_j = k$, the number of passengers boarding at the lobby. The floor $H := d_k$ is typically referred to as the highest reversal floor in the literature (Lee et al. 2009) and is useful in calculating the round trip time. We also need to approximate inter-floor traffic (including down traffic), which we do by using an estimated multiplier β (e.g., $\beta = 1.3$ which means it takes 30% longer down) from the time the elevator takes to drop off the last passenger. Then the ascent time $AscentTime(\vec{F})$ without accounting for stops,

is given by $AscentTime(\vec{F}) = T(1; d_k)$, the time spent making stops is $StopTime(\vec{F}) = \sum_{j=2}^{m} BoardingTime(F_j)$ and the descent time from when the last passenger has deboarded is $DescentTime(\vec{F}) = \beta \times AscentTime(\vec{F}) = \beta T(1; d_k)$. The boarding time for k passengers to board at the lobby is $BoardingTime(\vec{F}) = BoardingTime(\sum_{j=2}^{m} F_j)$. Thus the total round trip $RoundTripTime(\vec{F})$ time of the elevator n is

$$RoundTripTime(\vec{F}) = BoardingTime(\vec{F}) + AscentTime(\vec{F}) + StopTime(\vec{F}) + DescentTime(\vec{F}).$$

To measure the performance of different interventions in the simulation, we consider the following metrics: average waiting time of a passenger at the lobby, average number of passengers at the lobby (queue length), average number of passengers (load) in the elevator, average time spent by a passenger in the elevator, and the average number of stops (buttons pressed) in elevator trips. We also discuss other secondary considerations like human and material resources needed, ease of understanding for managers and passengers, and perceived inequity (Larson 1987, Berry et al. 2002) (e.g., when an intervention lets some passengers skip ahead of others).

Specifically, we want to mathematically characterize the performance of the system. Let W_i denote the wait time for passenger i. Let $N(t)$ denote the number of people waiting in the lobby at time t. In a classic service system, the goal is to typically minimize the total (average) expected wait time, i.e., $\mathbb{E}[\sum_i W_i]$. However, the primary objective in the context of a pandemic is to maximize safety, which corresponds to minimizing the number of people in the lobby that are waiting for an elevator. Metrics of interest could be the expected number of passengers in the lobby in a time horizon where we analyze the system $[0, T]$,

$\frac{1}{T}\int_0^T \mathbb{E}[N(t)]dt$, or the maximum queue length, $\max_t N(t)$.

Finally, we describe our system dynamics in the simulation. Passengers arrive at the (1st floor) lobby and queue in a line or multiple lines, depending on the intervention being implemented. When an elevator is available at the lobby (either there is a free elevator already or passengers wait for an elevator to arrive), it is loaded according to the rules of the intervention, up to the capacity limit C of the elevator. At constant intervals (Δt seconds), we update the system by loading available elevators in the lobby with passengers already in line(s) using the rules of the intervention.

3.2.1 Elevator Capacity

The capacity C of the elevators should be set based on the physical dimensions of each elevator. Social distancing needs to be taken into account to put floor markers for passengers to stand inside an elevator, e.g. opposite corners of a diagonal for loading two people or all corners for loading four people. In general, there is a fundamental trade-off between setting a lower elevator capacity and increased queues in the lobby.

3.2.2 Interventions

The standard way most elevator systems operate is akin to first-come first-serve (FCFS). However, moving towards safe interventions requires moving away from FCFS, which means that some people may be allowed to "cut in line" in order to have a more efficient movement of passengers. Next, we elaborate on FCFS and several interventions that relieve the potential congestion in the elevator lobby and shorten the waiting time for each passenger. The main

motivation behind each intervention is to reduce the average travel time for each elevator trip while serving as many passengers as possible. Our interventions will rely on a queue manager (QM) for implementation, where the QM can be thought of as a personnel or a device with a screen.

First, we discuss the status quo - *FCFS* - where the passengers who arrive at the lobby first will enter an elevator first. FCFS for elevator loading follows the standard social norm of queuing. There are obvious advantages for using the status quo, as it ensures fairness and requires no management of the queue. However, even pre-COVID, especially during rush hours such as morning and lunchtime, the lobby may be crowded with passengers waiting to get to their floor. Meanwhile, elevators are fully loaded and can make many stops during the trip. With a physical distancing rule during a pandemic such as COVID-19, the dramatically reduced elevator capacity could cause congestion in the lobby and thus increase the risk of disease spread.

Next, we propose the intervention which we call *Cohorting*, which seeks to group together passengers going to the same floor. In this intervention, passengers line up in a queue in order of arrival. When an elevator arrives, the first passenger boards. Then, the QM asks if anyone in the queue going to the same floor as the first passenger and they board as well (according to their arrival order), in order to create a cohort going to the same floor (such passengers are allowed to "cut in line"). If there is still capacity in the elevator, then the passenger at the front of the queue enters and the QM again allows passengers going to the same floor to board the elevator. This process is repeated until the elevator is full or the queue is empty. Cohorting is the best-performing intervention to improve efficiency (as seen in the Results section), but requires a QM to interact effectively with the queue to learn

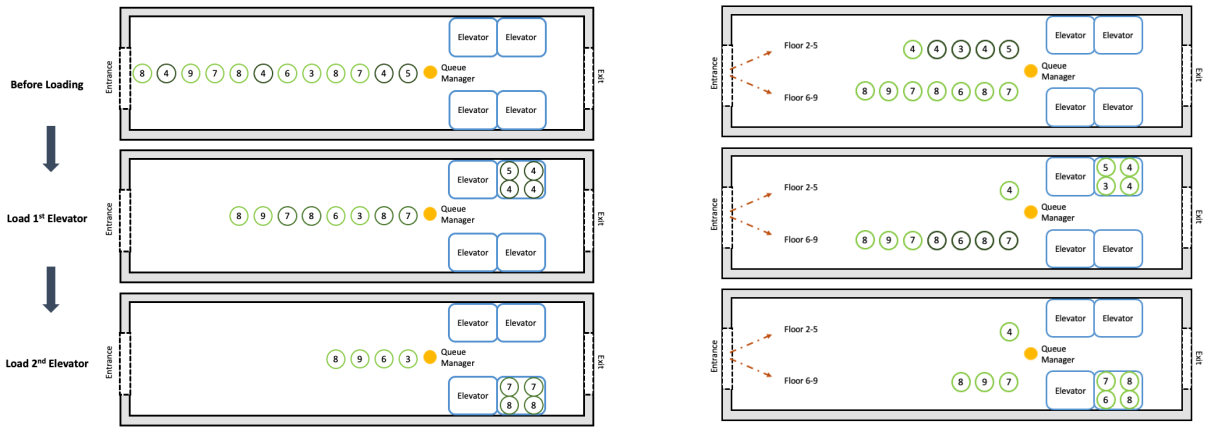
where passengers are going. It may be difficult for the QM to know the destinations of passengers that are far back in the queue. This consideration is discussed in Section 3.5 and is called Pairing.

Other considerations include the ease of understanding for the passengers and perceived inequity when passengers cut the line. Next, we consider two other interventions which may be easier to implement, even though they do not perform as well as Cohorting.

The next intervention we propose is *Queue Splitting*, where we form a separate queue for disjoint groups of floors. In Queue Splitting, floors are assigned to different groups, where each group consists of consecutive floors, e.g., $1-8$ and $9-16$. We create a queue corresponding to each floor group. Arriving passengers join a queue corresponding to their floor group, and elevators are boarded from the queues in a round robin fashion (possibly with the help of a QM). For instance, there can be 4 queues, each corresponding to 6 floors. When an elevator arrives, one of the queues sends the first C passengers in line to it. If there are less than C in the queue, then the next queue sends passengers and so on. The queues are chosen in a round-robin fashion (or in a way to dynamically balance the length of each queue). By creating queues for every floor group, the travel time of elevators is naturally reduced since passengers are likely to be going to the same or nearby floors, which achieves an effect similar to Cohorting. This intervention does not require any programming of the elevator system, but does require organizing the lobby space. A schematic showing the implementation of Cohorting and Queue Splitting is shown in Figure 3.1.

In the *Allocation* intervention, each elevator is assigned to only go to predetermined floors. This intervention can be accomplished by changing the elevator control system, or simply by adding signs on each elevator door. We propose several floor allocation interventions,

Figure 3.1: An illustration of floor layout with a QM to implement the Cohorting intervention and Queue Splitting intervention.



(a) Cohorting (b) Queue Splitting

Note. Passengers enter from the left and are guided by the Queue Manager to the elevators. Those exiting the elevator leave the building on the right, to ensure social distancing from entering passengers. In this example, the QM is loading an elevator in the lobby. We indicate the passengers who will board the next elevator using dark green circles. Under the Cohorting intervention, the first elevator will stop at two floors, which are the destinations of the first and second passenger. Under the Queue Splitting intervention, the QM is first loading from the queue for floor 2-5, and thus the elevator will stop at 3 floors.

including partitioning into ranges of floors, or splitting into odd and even floors. For instance, one building in our case study has 14 elevators and 24 floors to serve. We can split the 14 elevators into 2 groups of 7, where each group goes to 12 floors. Another possibility is to split into odd and even floors which may encourage people to use one level of stairs to reach their final destination. The key intuition behind the allocation intervention is that each elevator, or each set of elevators is only serving a small range of floors. By doing so, the chances of two people in the same group going to the same floor becomes relatively high compared to the case in which everyone is going to an arbitrary floor. Thus Allocation has an effect that resembles Cohorting, leading to a reduction in travel and deboarding time. However, there are two potential issues for Allocation intervention. First, the allocation of floors into different ranges needs careful design. On one hand, the traffic to different floors may vary a lot. On the other hand, it is clear that higher floors will need more elevators allocated,

as the travel time to higher floors are naturally longer than other floors. An allocation decision needs to be based on solving an optimization problem that may be difficult to solve. Different allocation designs may result in significantly different performance, and the imbalance introduced by a mistaken allocation decision could introduce more congestion in the lobby. The second issue is that it may be difficult to reprogram the elevators in some buildings which have outdated elevator systems. If the floor allocation is purely done by signage, then passengers can ignore such signs which may cause problems. Given these practical concerns, we shall only discuss the performance of the Allocation intervention in Appendix Section 3.7.6.

3.2.3 Discrete Event Simulation

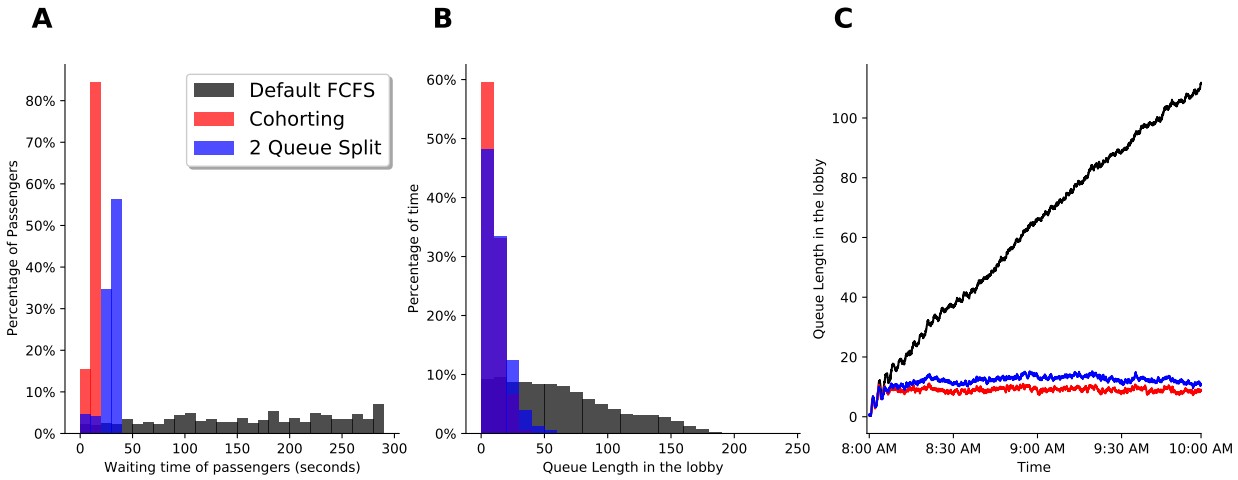
The first major step in the simulation is to generate the arrivals of passengers for each floor according to the Poisson process $\lambda_j(t)$ defined above. We also generate a binary variable with probability WtW for each customer to see if they are willing to walk one flight. We refer to a *scenario* as the sequence of passenger arrival time and information during one simulated rush hour morning. The second major step is the implementation of the proposed interventions. The logic of each intervention can be found in Subsection 3.2.2. We record all quantitative metrics listed in Section 3.2. We simulate 100 independent random scenarios and report the average performance. The code for the simulation is publicly available online.

3.3 Results

We simulate the results from three examples corresponding to a small, medium, and large building. The discussions from hereon are centered around the large building. Results about the other two buildings are in Appendix Section 3.7.3. The large building is calibrated using data from a large government building in New York City that is planning for re-opening and is in urgent need to manage elevator traffic amidst the COVID-19 pandemic. In particular, it is a historical building with a legacy elevator system, so that only technology-free solutions can be implemented. Moreover, this building is heavily used and had more than 5500 people (staff and visitors) accessing it on a pre-pandemic day during the rush hour. The building has $m = 25$ floors and the $N = 28$ elevators are split into two elevator banks (North and South). Without loss of generality, we consider the South bank, where 14 elevators serve about 2750 visitors during the morning rush hour (8-10 AM). In the two-hour period, we assume the arrival process is a stationary Poisson Process with an arrival rate 2750/7200 passengers per second. Based on the physical dimensions of the elevators, the capacity is $C = 4$. Every elevator n serves all floors, i.e., $S(n) = \{1, 2, \ldots, 25\}$. It take 15s for one passenger to (de)board, and an additional 2s per extra passenger. Thus $BoardingTime(k) = 15 + 2(k-1)$ seconds for k passengers in an elevator. The elevators have a constant speed of 1.4 seconds/floor, hence the time to travel from floor j_1 to floor j_2 is $T(j_1; j_2) = 1.4(j_2 - j_1)$ seconds. The speed multiplier β to account for inter-floor traffic is $\beta = 1.3$. Parameters used for the simulations are described in Appendix Section 3.7.2. Note that the queues will all diminish after the end of rush hour because the passenger traffic

goes down, but we do not simulate this. In the figures below, we report the results when we stop the simulation just at the end of rush hour (peak) and hence the queue decline after this time is not shown.

Figure 3.2: Comparison of interventions for our large building case study.



Note. We run 100 independent random scenarios and report the average performance. **A)** Plot of percentage of passengers experiencing different waiting times in the lobby across interventions. **B)** Plot of percentage of time different queue lengths in the lobby occur (measured every 1 second) across interventions. **C)** Plot of queue length in the lobby from beginning to end of the busy period across interventions.

The results for FCFS, Cohorting, and Queue Splitting (2 queues with floor ranges $2-13$ and $14-25$) on the large building are presented in Figure 3.2. One can observe that for FCFS, the number of people in the lobby grows linearly during the rush hour period we simulate. In fact, by the end of the rush hour, there can be up to 100 people in the queue and wait times can reach almost five minutes. Thus, an intervention is absolutely necessary to avoid this unsafe buildup of passengers. We see that Cohorting has a much lower range of queue lengths and waiting times compared to FCFS. Cohorting has a maximum queue length of around 12, which is over a factor of eight times smaller than the maximum queue length of FCFS. In other words, a passenger arriving at any point in the rush hour is likely to experience a queue of at most 12 people with the Cohorting intervention.

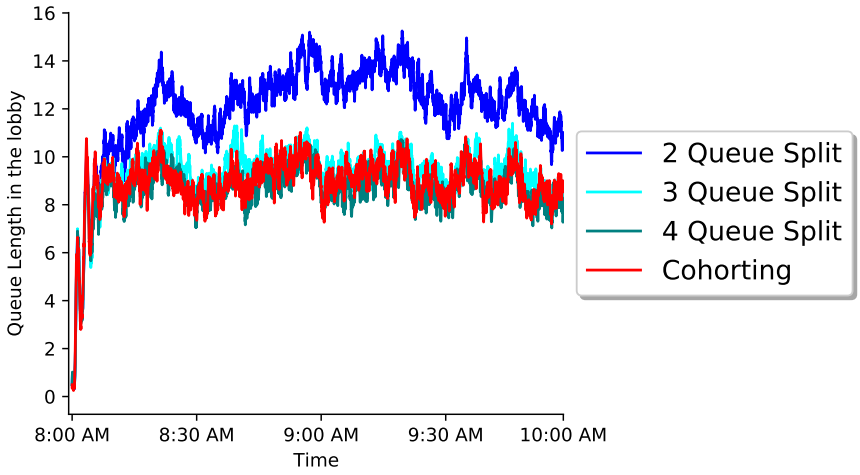
In the Queue Splitting intervention, we do not allocate any elevators but rather form a queue for every group of floors. For instance, there can be 2 queues that each operate as FCFS, where each queue corresponds to 12 floors. Some technology in addition to signage may be needed to allocate queues to arriving elevators. In Figure 3.2, we see that Queue Splitting (into 2 queues) has a much lower range of queue lengths and waiting times compared to FCFS. Similar to Cohorting, the queue length has in this intervention also has a maximum queue length of around 15. In other words, a passenger arriving at any point in the rush hour is likely to experience a queue of less than 15 people in the 2 Queue Split intervention. This is over a factor of five times smaller than the maximum queue length of FCFS. One can see in Figure 3.2 that the maximum wait times and total queue length are relatively stable over time for this intervention, and with average reductions of over 80% compared to the default FCFS. Thus a 2 Queue Split achieves comparable performance to Cohorting, the best intervention.

3.3.1 Number of Queues for Queue Splitting

Figure 3.3 shows the performance of queue splitting into different number of queues for the large building. The floor ranges for each queue are $\{(2-13),(14-25)\}$ in the 2 Queue Split, $\{(2-9),(10-17),(18-25)\}$ in the 3 Queue Split, and $\{(2-7),(8-13),(14-19),(20-25)\}$ in the 4 Queue Split. In this building, 4 queue split works better than 2 and 3 queue splits (higher number of queue splits achieves an effect similar to Cohorting). There is a marked improvement (Figure 3.3) from 2 to 3 queue split and only a marginal return on 4 queue split (which has almost the same queue length performance as Cohorting) instead of 3 queue

split. In fact, the more queues we create, the more efficient the system becomes. However, the tradeoff is that more queues requires a more complex operation and more space in the lobby, especially for horizontal separation between the queues. We find that simply splitting into 2-4 queues already recovers most of the benefit from the Cohorting intervention.

Figure 3.3: Impact of Cohorting and Queue Splitting intervention into $2, 3$ and 4 queues for the large building case study.



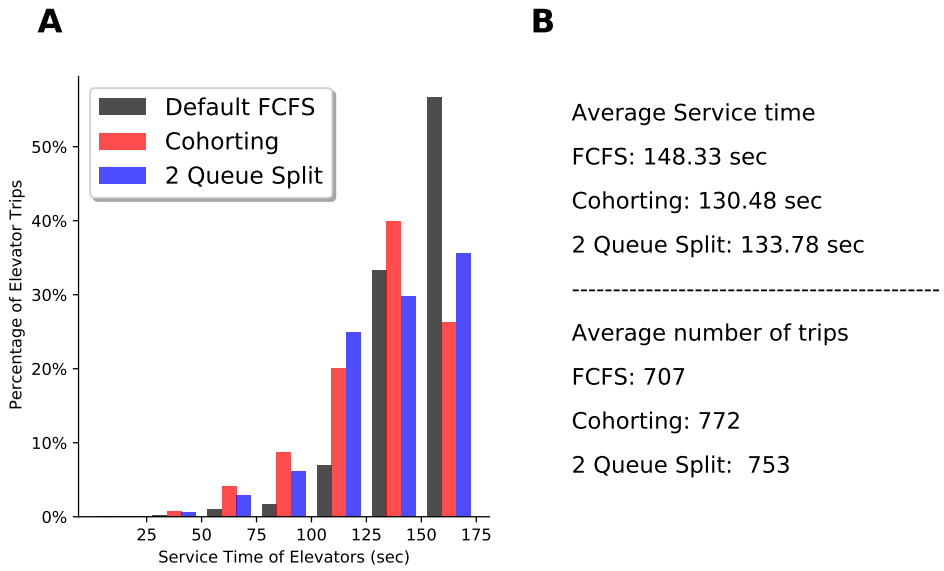
Note. We plot the queue length in the lobby (measured every 1 second) throughout rush hour. We run 100 independent random scenarios and report the average performance.

3.3.2 Difference in Round Trip Time

The round trip time of an elevator trip determines the efficiency of the elevator system. The shorter the round trip time is, the faster the elevator can come back and serve more people. We record the service time profile for FCFS, Cohorting, and 2 Queue Split in the simulation in Figures 3.4.

In Figure 3.4, Cohorting and Queue Splitting have a lower average service time (130s and 133s respectively) than FCFS (148s). In terms of number of trips it can complete in the given time period, Cohorting and 2 Queue Split is much better than FCFS, indicating that

Figure 3.4: Comparison of interventions using service time for our large building case study



Note. We run 100 independent random scenarios and report the average performance. **A)** Plot of percentage of elevator trips with different round trip times (service times) across interventions **B)** Reporting average service time and average number of trips for all interventions.

our proposed interventions indeed make the elevator trips more efficient. Note that our main indicator of system performance, queue length in such a service system is inversely related to average service time ($\propto 1/(\mu - \lambda)$ in classic queueing textbooks, e.g. Ross et al. (1996)) and even a seemingly small improvement in the round trip time as in Figure 3.4 has a big impact on system performance.

From the simulation results, we conclude that Cohorting and Queue Splitting reduce the round trip time for the elevator trips and serve more passengers in a time period. The excellent numerical results motivate us to think about the reason why the two interventions perform similarly well while managing the queue completely differently. In the next section, we offer theoretical support for our proposed interventions and dive into the distribution of round trip time.

3.4 Stability Analysis

In this section, we investigate the theory behind the good performance of the proposed intervention Cohorting and Queue Splitting. As observed in Figure 3.2, the queue length does not grow over time under the Cohorting and Queue Splitting intervention, while under FCFS it keeps increasing. In other words, by using the Cohorting and Queue Splitting intervention, we manage to transfer an unstable queuing network into a stable one under the simulation setting in Section 3.3. In this section, we aim to establish stability conditions for each intervention and explain why the proposed interventions work. In Section 3.4.3 we calculate the stability condition for a special case and in Section 3.4.4 we extend our analysis to general settings. We find that our proposed interventions can lead to a stable queue with higher arrival rate, not only because it reduces the number of stops in comparison to FCFS, but also due to the fact that the elevator trips tend to make less trips to high floors, which is supported by a stochastic dominance result for the highest reversal floor distribution.

3.4.1 Stability Condition for the Queuing Network

In this section, we describe the definition and results in the literature on stability of multiclass queuing networks with different operations rules. The model we focus on has I buffers (or arrival classes), each corresponding to a service type, and K resources. The arrival process for each buffer i is a Poisson Process with rate λ_i, and the service for type i requires a random time with mean t_i. Each resource k is a pool of b_k identical servers. $b = (b_1, \ldots, b_K)$. Each job class is processed by servers from a single specified pool, and each such service is

accomplished by a single server from the pool. The set of buffers that resource k can serve is defined as $\mathcal{I}(k)$. The load vector $\rho = (\rho_1, \ldots, \rho_K)$ is defined as $\rho_k = \sum_{i \in \mathcal{I}(k)} \lambda_i t_i$.

We see below that the stability condition

$$\rho < b \tag{3.1}$$

is a sufficient and necessary condition for the queuing network we study in this paper. For some queuing systems such as $M/M/1$ queue, it is well-known that the system is stable if and only if the load vector ρ is less than 1. However, it is not always true that Eq. (3.1) is a sufficient condition for a general queuing network (Bramson et al. 1994). In Lemma 3.4.1 below, we establish the stability condition for a feedforward queuing network under non-idling control policy. Fortunately, the queuing models corresponding to the elevator interventions all falls within the category of a *feedforward queuing network*, which is defined as a queuing network in which the resources can be numbered in such a way that the arrival jobs never move from higher numbered servers to lower numbered ones. Also, the interventions we propose are *non-idling dynamic control policies*, which is defined as a control policy if no server remains idle while there is a job waiting in any of the buffers that are processed by the server.

Lemma 3.4.1. *Under any non-idling policy, a feedforward queuing network is stable if and only if the stability condition Eq.* (3.1) *holds.*

The proof of Lemma 3.4.1 is provided in Appendix 3.7.7, where we combine multiple results in Dai and Harrison (2020) to establish this key lemma of our analysis.

In the following subsections, we will specify the structure of the queuing network for each intervention, and derive the stability condition according to Eq. (3.1).

3.4.2 Assumptions and Justification

We first simplify the assumptions in order to better deliver the analysis. We formulate the elevator system as a queuing network, in which the m elevators are the servers and the round trip time of an elevator trip is the service time. As discussed in Section 3.2, ν is the time it takes an elevator to travel one floor and we simplify the stop time of an elevator for one stop to ω. From the count $\vec{F}$, we also define the number of stops $S = \sum_{j=2}^{m} \mathbb{1}\{F_j > 0\}$ made in an elevator trip. The round trip time for an elevator trip $\tau(H, S)$ is a random variable with finite support, where its mean value depends on the random variables H and S. We also simplify the ascent time and descent time to be the same value νH, and omit the boarding time at the lobby, which is the same for all interventions. The conditional average round trip time is defined as

$$\mathbb{E}[\tau(H, S)|H, S] := 2\nu H + \omega S. \tag{3.2}$$

The distributions of H and S are determined by the random arrival process and the intervention, and are used only in Eq. (3.2) to determine the expected round trip time. We do not make any further assumption on the distribution of the round trip time.

Next, we describe the queuing networks that represent the dynamic of the elevator queues under different interventions. Our goal is to derive the stability condition, under which the queue does not grow over time under an intervention. Thus, we only consider such a condition in a system with extremely long queues. We assume that from now on, we treat

a set of C passengers as one arrival job to the system. This can be interpreted as grouping C passengers outside the building and letting them take the same elevator. For example, an arrival job under FCFS will be C random passengers with independent and uniform distributed destinations, while an arrival job under the Queue Splitting intervention will be C random passengers who are going to the same subset of floors.

We make the assumptions above in order to provide theoretical insights for the simulation results, allowing us to use tools from the literature of processing networks. Dai and Li (2003) provide stability analysis for batch service systems, and the queue under the operation rule corresponding to the Cohorting and Queue Splitting intervention is rate stable under the stability condition (3.1). Nevertheless, in the following subsections, we focus on the analysis of the stability condition (3.1) for different interventions.

3.4.3 1 Elevator and 2 Floors

We first focus on the simplest setting where the building only has two destination floors 1 and 2 and one elevator with capacity $C = 2$. (To avoid confusion in the notation, we assume the lobby is on the ground floor.) The passengers arrive at the building according to a Poisson Process with rate λ, and go to floor 1 or 2 with equal probability. Equivalently, the arrival process for passengers who go to floor 1 (2) is a Poisson Process with rate $\frac{\lambda}{2}$.

The elevator system under FCFS is an $M/G/1$ queue, where the new jobs arrive according to a Poisson process with rate $\lambda/2$, and the elevator is the server with the time to complete a round trip being the service time. In an $M/G/1$ queue, the stability condition Eq. (3.1) is well-known and can be found in the literature and the textbooks (e.g. Ross et al. (1996)

Example 4.3(a)). It can also be derived from Lemma 3.4.1 since an $M/G/1$ queue is a special case of a feedforward queuing network, and FCFS is apparently a non-idling policy.

For the Cohorting and Queue Splitting intervention, we can think of the following multi-class queuing network with two queuing buffers and one server. The passengers who go to floor 1 form a queue in buffer 1, and the passengers who go to floor 2 form a separate queue in buffer 2. The passengers are also assigned into pairs, and a pair of passengers is considered as a job that is waiting to be served. This network with two buffers and one server satisfies the condition of a feedforward queuing network. In the Queue Splitting intervention, the server will choose a buffer to serve in a round-robin fashion once it finishes the previous job. In the Cohorting intervention, the server will choose a buffer whose head of queue arrives earlier to the system. In the literature, this is called a FCFS control policy (Bramson et al. 1994), where at each server the jobs are processed according to their arrival time to the system. Both the Queue Splitting and Cohorting intervention are non-idling policies. Therefore, we can again use Lemma 3.4.1 to derive the stability condition.

Theorem 3.4.1. *(a) The elevator queue is stable under FCFS if and only if the arrival rate*

$$\lambda < \eta_{FCFS} := \frac{4}{7\nu + 3\omega}.$$

(b) The queuing network is stable under the Cohorting and Queue Splitting intervention if and only if

$$\lambda < \frac{4}{6\nu + 2\omega}.$$

Proof. (a) In the $M/G/1$ queue corresponding to FCFS, the service time distribution is a

mixture of 3 different distributions. With probability 0.25, the highest reversal floor H is 1 and the total number of stops S is 1. Similarly, with probability 0.25, the elevator only stops once and H is 2. With probability 0.5, H is 2 and S is 2. Therefore, the expected service time is

$$\begin{aligned} 0.25\left(\mathbb{E}[\tau(1,1)] + 2\mathbb{E}[\tau(2,2)] + \mathbb{E}[\tau(2,1)]\right) &= \frac{2\nu + \omega + 4\nu + 2\omega + 2(4\nu + 2\omega)}{4} \\ &= \frac{7\nu + 3\omega}{2}. \end{aligned}$$

According to the stability condition Eq. (3.1), we can plug in $\lambda/2$ as the arrival rate of each pair of passengers and get $\frac{\lambda}{2} \cdot \frac{14\nu+6\omega}{4} < 1$. Therefore,

$$\lambda < \frac{4}{7\nu + 3\omega}.$$

(b) The queuing network we defined for Queue Splitting and Cohorting intervention satisfies the condition of a feedforward queuing network with non-idling service policy. Therefore, we only need to specify the average service time for each buffer. For buffer 1, all passengers are going to floor 1, so the expected service time is $\mathbb{E}[\tau(1,1)]$. Similarly, for buffer 2, the expected service time is $\mathbb{E}[\tau(2,1)]$. Then we plug in Eq. (3.1) and get

$$\lambda/4 \cdot \mathbb{E}[\tau(1,1)] + \lambda/4 \cdot \mathbb{E}[\tau(2,1)] < 1.$$

Therefore, the stability condition becomes $\lambda < \frac{4}{\mathbb{E}[\tau(1,1)]+\mathbb{E}[\tau(2,1)]} < \frac{4}{6\nu+2\omega}$. □

Define the stability threshold $\eta_{Cohort} = \eta_{QS} := \frac{4}{6\nu+2\omega}$. Since the stability threshold is an

upper bound on the total arrival rate of passengers, the higher the threshold is, the better the system can deal with rush hour traffic. In the following proposition, we establish by how much the proposed interventions can improve the stability threshold.

Proposition 3.4.1. *In a building with 1 elevator and two floors, if the capacity of the elevator is 2, the Cohorting and Queue Splitting intervention can increase the stability threshold by at least* 16.67%*, and at most* 50%*.*

Proof. In this proof, we want to bound the ratio $\frac{\eta_{QS}}{\eta_{FCFS}}$. Plugging in the threshold we get from Theorem 3.4.1, we have

$$\frac{\eta_{QS}}{\eta_{FCFS}} = \frac{7\nu + 3\omega}{6\nu + 2\omega}.$$

Since ν and ω are positive real number, $\frac{7}{6} < \frac{7\nu+3\omega}{6\nu+2\omega} < \frac{3}{2}$, which yields the final result. □

Proposition 3.4.1 provides a clean explanation to the phenomenon we observe from the simulation: when facing the same passenger arrival pattern, the stability condition for FCFS is violated while the arrival rate is still below the threshold for Queue Splitting and Cohorting. Thus the key driver for the good performance of the proposed interventions is that the expected service time is much shorter, thanks to the fact that both the highest reversal floor and the number of stops have smaller values. When we use the FCFS intervention, $H = 2$ with probability 0.75, despite the fact that only half of the passengers go to floor 2. Using Cohorting and Queue Splitting intervention, we can make sure that only 50% of the time the elevator will go to the higher floor. For the number of stops per elevator trip, using Cohorting and Queue Splitting intervention, we can ensure that the elevator only makes one

stop in each elevator trip, while in FCFS, 50% of the time the elevator will make 2 stops. Our proposed interventions can simultaneously reduce the stop time and the travel time of the elevator, and make the elevator trips more efficient. The analysis in this subsection is a toy example for illustration given the fact that we can compute the expected service time exactly. In the next subsection, we will extend the analysis to a building with multiple elevators and floors.

3.4.4 General Case

In this subsection, we focus on a general building with m floors and N identical elevators that can serve all the floors. Though the expected service time is complicated to compute exactly in the general setting, we can focus on the distribution of H and S and show that the stability threshold for Queue Splitting and Cohorting is much higher than the one for FCFS.

In this subsection, we again treat C passengers as one new job to the queuing system.

Lemma 3.4.2. *For intervention $\pi \in \{FCFS, Cohorting, QS\}$, the queue is stable if and only if the arrival rate λ is less than $\eta_\pi := \frac{NC}{2\nu\mathbb{E}[H_\pi]+\omega\mathbb{E}[S_\pi]}$.*

Proof. We first start with FCFS. By Eq. (3.1), the stability condition is

$$\frac{\lambda}{C}\mathbb{E}[\tau(H_{FCFS}, S_{FCFS})] < N,$$

which is equivalent to

$$\frac{\lambda}{C}\left(2\nu\mathbb{E}[H_{FCFS}]+\omega\mathbb{E}[S_{FCFS}]\right)<N. \tag{3.3}$$

Under the Cohorting intervention, the queuing network consists of m independent buffers for m floors, and an idle server will choose to serve the buffer with the earliest arrival time. By Eq. (3.1), we need to specify the average round trip time for each buffer. The highest reversal floor for buffer i is simply i, and every trip only has one stop. Therefore, the stability condition (3.1) becomes $\sum_{i=1}^{m}\lambda_i\mathbb{E}[\tau(H,1)|H=i]<N$, where $\lambda_i=\frac{\lambda}{Cm}$. Note that the highest reversal floor H_{Cohort} follows a uniform distribution, we can rewrite the left hand side of the stability condition into

$$\begin{aligned}\sum_{i=1}^{m}\lambda_i\mathbb{E}[\tau(H,1)|H=i] &= \frac{\lambda}{C}\sum_{i=1}^{m}\frac{1}{m}\mathbb{E}[\tau(H,1)|H=i]\\ &= \frac{\lambda}{C}\left(2\nu\mathbb{E}[H_{Cohort}]+\omega\right).\end{aligned} \tag{3.4}$$

Under the Queue Splitting intervention, we assume that the m floors are divided into l groups, where each group is a separate buffer and consists of k consecutive floors. $m=lk$ and $l,k\geq 2$ are integers. The stability condition becomes $\sum_{i=1}^{l}\lambda_i\mathbb{E}[\tau(H,S)|H\in \text{ group } i]<N$, where $\lambda_i=\frac{\lambda}{Cl}$ in this case. Since the jobs are processed in a round robin fashion, the probability for a new job to be in group i is $\frac{1}{l}$. Then we can rewrite the left hand side of the stability condition as the following.

$$\sum_{i=1}^{l}\lambda_i\mathbb{E}[\tau(H,S)|H\in \text{ group } i]=\frac{\lambda}{C}\sum_{i=1}^{l}\frac{1}{l}\mathbb{E}[\tau(H,S)|H\in \text{ group } i]$$

$$= \frac{\lambda}{C} \left(2\nu \mathbb{E}[H_{QS}] + \omega \mathbb{E}[S_{QS}]\right). \tag{3.5}$$

Therefore, following the result in Lemma 3.4.1, for FCFS, Cohorting, and Queue Splitting intervention, the queue is stable if and only if $\lambda < \frac{NC}{2\nu \mathbb{E}[H_\pi] + \omega \mathbb{E}[S_\pi]}$. □

From Eq. (3.3), (3.4), and (3.5), we know that the key to compare the stability conditions for different interventions is to compare the expectation of the highest reversal floor and number of stops. The analysis for FCFS can be found in the literature of elevator analytics (Barney and Al-Sharif 2015), where the Cohorting and Queue Splitting intervention are not usual management rules for elevators. Next, we compute the distribution and expectation of the highest reversal floor and number of stops and provide a comparison.

Lemma 3.4.3. *In a building with m floors, the Queue Splitting intervention divides m into $l \geq 2$ groups, each with $k \geq 2$ floors.*

(a) The cumulative function of the highest reversal floors is as follows,

$$\begin{aligned}
\mathbb{P}[H_{FCFS} \leq x] &= \left(\frac{x}{m}\right)^C, \\
\mathbb{P}[H_{QS} \leq x] &= \frac{i}{l} + \frac{1}{l}\left(\frac{j}{k}\right)^C, \\
\mathbb{P}[H_{Cohort} \leq x] &= \frac{x}{m},
\end{aligned}$$

where $x = 1, \ldots, m$, $i = \left\lfloor \frac{x}{k} \right\rfloor$, and $j = x - ik$.

(b) The expectation of the highest reversal floors is as follows,

$$\mathbb{E}[H_{FCFS}] = m - \frac{1}{m^C} \sum_{i=1}^{m-1} i^C, \tag{3.6}$$

$$\mathbb{E}[H_{QS}] = \frac{k(l+1)}{2} - \sum_{j=1}^{k-1} \left(\frac{j}{k}\right)^C, \tag{3.7}$$

$$\mathbb{E}[H_{Cohort}] = \frac{m+1}{2}. \tag{3.8}$$

(c) H_{FCFS} stochastically dominates H_{QS} and H_{Cohort}, i.e., $H_{FCFS} \succeq H_{QS} \succeq H_{Cohort}$. Therefore,

$$\mathbb{E}[H_{FCFS}] \geq \mathbb{E}[H_{QS}] \geq \mathbb{E}[H_{Cohorting}]. \tag{3.9}$$

Proof. (a) In the FCFS intervention, the random event of the highest reversal floor to be no larger than x is equivalent to the event that the destination of each passenger is randomly chosen from 1 to x, which directly gives us the result. The distribution of the highest reversal floor for Cohorting directly follows from the definition of a uniform distribution on value $1, \ldots, m$.

For the Queue Splitting intervention, conditioning on the current service group is group i, the probability of the highest reversal floor is no larger than $x = ik + j$ is equal to $\left(\frac{j}{k}\right)^C$, following the same argument for the FCFS intervention. Note that x is in floor group $i+1$, all the trips in group $1, \ldots, i$ satisfy the condition $H_{QS} \leq x$.

$$\mathbb{P}[H_{QS} \leq x] = \sum_{y=1}^{i} \mathbb{P}[H_{QS} \text{ in group } y] + \mathbb{P}[H_{QS} \leq ik + j | H_{QS} \text{ in group } i+1] = \frac{i}{l} + \frac{1}{l}\left(\frac{j}{k}\right)^C.$$

(b) It is trivial for the Cohorting intervention, as $\mathbb{E}[H_{Cohort}] = \frac{m+1}{2}$. We then use the tail

formula to derive the expectation for FCFS and Queue Splitting intervention.

$$\begin{aligned}
\mathbb{E}[H_{FCFS}] &= \sum_{i=1}^{\infty} \mathbb{P}[H_{FCFS} \geq i] \\
&= \sum_{i=1}^{m} (1 - \mathbb{P}[H_{FCFS} \leq i-1] \\
&= m - \sum_{i=1}^{m} \left(\frac{i-1}{m}\right)^C \\
&= m - \frac{1}{m^C} \sum_{i=1}^{m-1} i^C. \\
\mathbb{E}[H_{QS}] &= \sum_{x=1}^{m} \mathbb{P}[H_{QS} \geq x] \\
&= \sum_{i=0}^{l-1} \sum_{j=1}^{k} \mathbb{P}[H_{QS} \geq ik + j] \\
&= \sum_{i=0}^{l-1} \sum_{j=1}^{k} (1 - \mathbb{P}[H_{QS} \leq ik + j - 1] \\
&= \frac{k(l+1)}{2} - \sum_{j=1}^{k-1} \left(\frac{j}{k}\right)^C.
\end{aligned}$$

(c) We first prove the that the random variables preserve stochastic dominance, i.e., $H_{Cohort} \preceq H_{QS} \preceq H_{FCFS}$. By definition, we only need to verify that $\mathbb{P}[H_{Cohort} \leq x] \geq \mathbb{P}[H_{QS} \leq x] \geq \mathbb{P}[H_{FCFS} \leq x]$ is true for all x. The first inequality is easy to verify since

$$\begin{aligned}
\mathbb{P}[H_{Cohort} \leq x] &= \frac{ik + j}{kl} \\
&= \frac{i}{l} + \frac{1}{l}\frac{j}{k} \\
&\geq \frac{i}{l} + \frac{1}{l}\left(\frac{j}{k}\right)^C \\
&= \mathbb{P}[H_{QS} \leq x].
\end{aligned}$$

To verify the second inequality, we only need to use the fact that $x = ik + j$, $i \leq l$, $l \geq 2$, and all the values being positive integers.

$$\begin{aligned}
\mathbb{P}[H_{FCFS} \leq x] &= \left(\frac{x}{m}\right)^C \\
&= \left(\frac{ik+j}{kl}\right)^C \\
&= \left(\frac{i}{l} + \frac{j}{kl}\right)^C \\
&\leq \left(\frac{i}{l}\right)^C + \left(\frac{j}{kl}\right)^C \\
&\leq \frac{i}{l} + \frac{1}{l}\left(\frac{j}{k}\right)^C \\
&= \mathbb{P}[H_{QS} \leq x].
\end{aligned}$$

Immediately following the stochastic dominance result, we can get the desired ordering in expectation. □

Since our simulation results in Section 3.3 are for a 25-story building, the highest reversal floor plays an essential role in the performance of the elevator system. Lemma 3.4.3 strongly supports the good performance of Cohorting and Queue Splitting in the simulation, as the distribution of the highest reversal floor preserves stochastic dominance across the three interventions we study. With the formula of the expectation, we next provide guarantees on the potential improvement by deriving the ratio of the expected highest reversal floor between interventions.

Proposition 3.4.2. *(a) The ratio of the expected highest reversal floor between FCFS and Cohorting is at least* $\frac{2Cm}{(C+1)(m+1)}$.

(b) For the special case of $C = 2$ and $m \geq 3$, the ratio of the expected highest reversal floor between FCFS and Cohorting is equal to $\frac{4m-1}{3m}$, which at least $\frac{11}{9}$. The ratio reaches lower bound when there are only 3 floors and reaches the upper bound when the number of floors grows to infinity.

(c) For the special case of $C = 2$, the ratio of the expected highest reversal floor between FCFS and Queue Splitting is equal to $\frac{4m^2+3m-1}{3m^2+3m+mk-l}$.

Proof. (a) We first bound $\mathbb{E}[H_{FCFS}]$ and $\mathbb{E}[H_{QS}]$ by replacing summation with integral.

$$\mathbb{E}[H_{FCFS}] = m - \frac{1}{m^C}\sum_{i=1}^{m-1} i^C \geq m - \int_{x=0}^{m} x^C dx = m - \frac{m}{C+1}. \tag{3.10}$$

$$\begin{aligned}\mathbb{E}[H_{QS}] &= \frac{k(l+1)}{2} - \sum_{j=1}^{k-1}\left(\frac{j}{k}\right)^C \\ &\leq \frac{k(l+1)}{2} - \int_{x=0}^{k-1}\left(\frac{x}{k}\right)^C dx \\ &= \frac{k(l+1)}{2} - \frac{(k-1)^{C+1}}{(C+1)k^C}.\end{aligned} \tag{3.11}$$

We then compare H_{FCFS} to H_{Cohort}.

$$\begin{aligned}\frac{\mathbb{E}[H_{FCFS}]}{\mathbb{E}[H_{Cohort}]} &\geq \frac{m - m/(C+1)}{(m+1)/2} \\ &= \frac{2Cm}{(C+1)(m+1)}.\end{aligned}$$

(b) Observe that we can compute $\mathbb{E}[H_{FCFS}]$ explicitely when $C = 2$.

$$\begin{aligned}
\frac{\mathbb{E}[H_{FCFS}]}{\mathbb{E}[H_{Cohort}]} &= \frac{m - \frac{1}{m^2}\sum_{i=1}^{m-1} i^2}{(m+1)/2} \\
&= \frac{m - \frac{1}{m^2}\frac{(m-1)(m-1+1)(2(m-1)+1)}{6}}{(m+1)/2} \\
&= \frac{6m^2 - (m-1)(2m-1)}{3m(m+1)} \\
&= \frac{4m-1}{3m} \\
&= \frac{4 - \frac{1}{m}}{3}. \qquad (3.12)
\end{aligned}$$

Note that Eq. (3.12) is increasing in m, we can plug in $m = 3$ and yield the lower bound on the ratio and send m to infinity to get the upper bound.

(c) In the special case of $C = 2$, we can also explicitly compute the expectation of H_{QS}. Note that $m = kl$, we can simply the ratio and have

$$\begin{aligned}
\frac{\mathbb{E}[H_{FCFS}]}{\mathbb{E}[H_{QS}]} &= \frac{m - \frac{1}{m^2}\frac{(m-1)(m-1+1)(2(m-1)+1)}{6}}{\frac{k(l+1)}{2} - \frac{(k-1)(2k-1)}{6k}} \\
&= \frac{6m^2 - (m-1)(2m-1)}{3k^2l^2 + k^2l + 3kl - l} \\
&= \frac{4m^2 + 3m - 1}{3m^2 + 3m + mk - l}. \qquad (3.13)
\end{aligned}$$

Note that $\frac{4m^2+3m-1}{3m^2+3m+l(m-1)}$ is always greater than 1 since $l < m$. □

From Proposition 3.4.2, we can observe that the distribution of highest reversal floor can be shifted to lower values through the Queue Splitting intervention. The more groups it splits into, the greater the reduction can be. The expected value of the number of stops can be found in the book Barney and Al-Sharif (2015). We summarize the results in the

following lemma.

Lemma 3.4.4. *(a) For each intervention, the distribution of the number of stops is as follows*

$$S_{Cohort} = 1 \text{ with probability } 1,$$
$$\mathbb{P}[S_{FCFS} = x] = \frac{x!}{m^C} \binom{m}{x} \left\{ {C \atop x} \right\}, x = 1, \ldots, \min\{C, m\},$$
$$\mathbb{P}[S_{QS} = x] = \frac{x!}{k^C} \binom{k}{x} \left\{ {C \atop x} \right\}, x = 1, \ldots, \min\{C, k\},$$

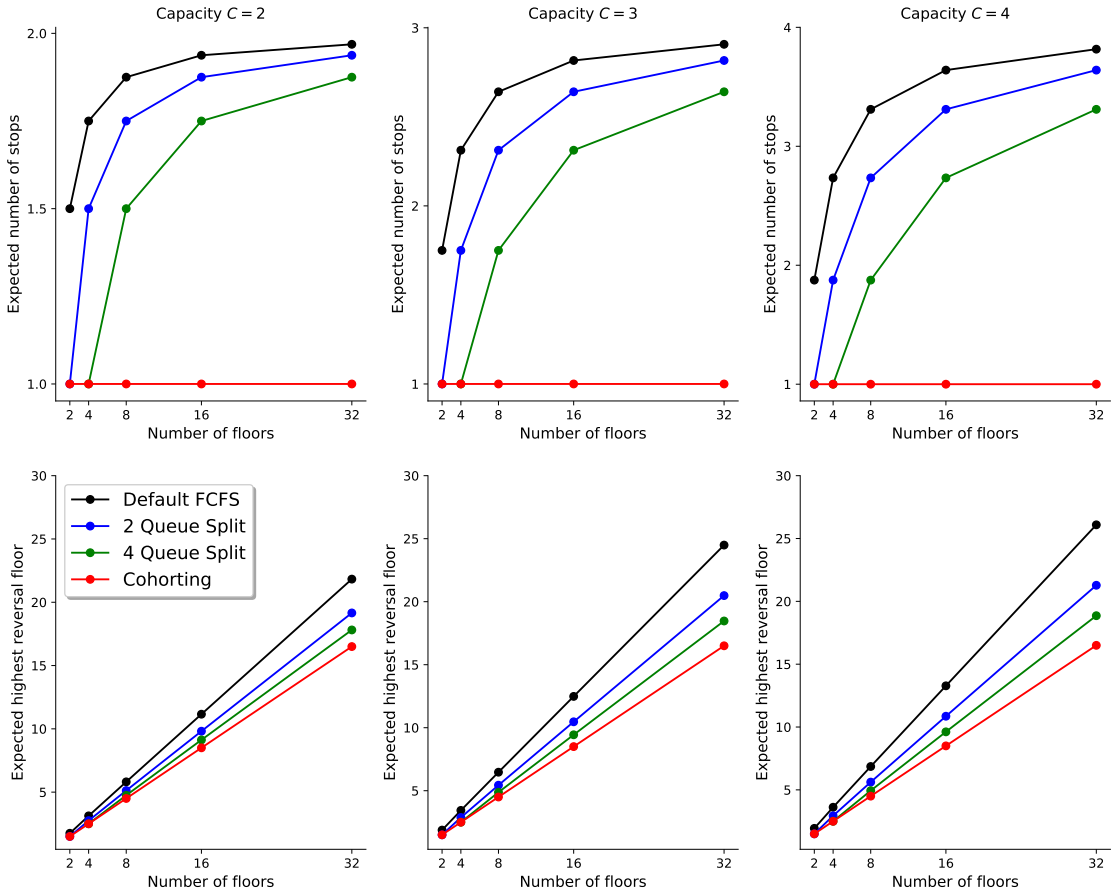
where $\left\{ {C \atop x} \right\}$ *is the Stirling number of the second kind, the number of ways to partition a set of* C *objects into* x *non-empty subsets.*

(b) The expected number of stops for FCFS and Queue Splitting is

$$\mathbb{E}[S_{FCFS}] = m \left[1 - \left(\frac{m-1}{m} \right)^C \right],$$
$$\mathbb{E}[S_{QS}] = k \left[1 - \left(\frac{k-1}{k} \right)^C \right].$$

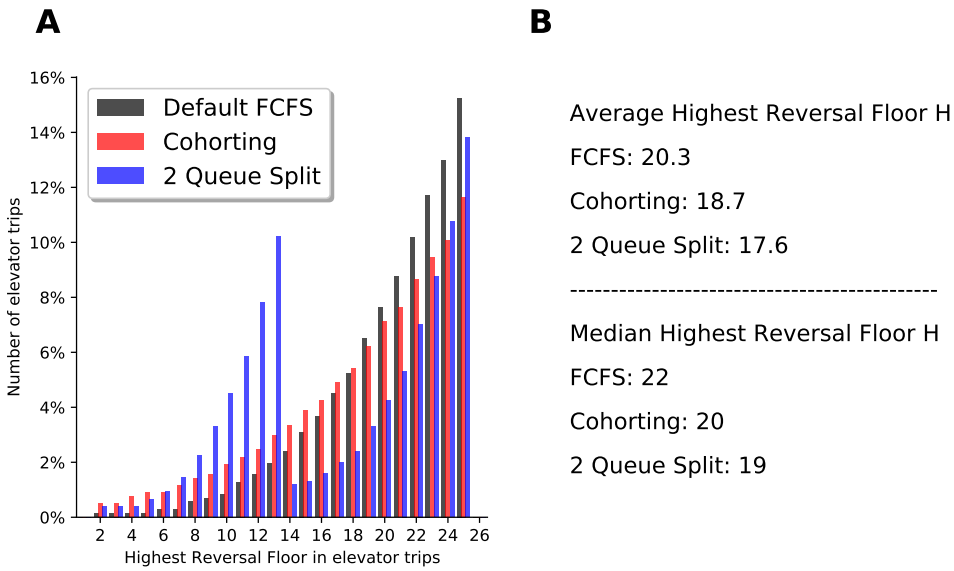
With the distribution of H and S being derived in in Lemma 3.4.3 and 3.4.4, we plot the expected values in Figure 3.5 for various parameter settings. When the number of floors m becomes large, the expected number of stops will approach the capacity C in both FCFS and Queue Splitting intervention. However, Queue Splitting is increasing much slower, and the more groups we split into, the lower the value is. Similar behavior can be observed in the expected highest reversal floor graphs. The higher the capacity C is, and the larger the number of floors m is, the more difference we can observe from the graph.

Figure 3.5: Expected highest reversal floor and number of stops in Lemma 3.4.3 and 3.4.4



Note that the analysis of Cohorting is in an unrealistic situation that always create a cohort of passengers perfectly. Although in theory Cohorting is always the winner in highest reversal floor and number of stops (Lemma 3.4.3 and 3.4.4), in the numerical results, we see that Cohorting has inferior performance in highest reversal floor comparing to 2 Queue Split in Figure 3.6. Cohorting has a slightly lower average H value (18.7) than FCFS whereas the difference is noticeable for Queue Splitting (17.6). While H may have a smaller effect than S on service times practically (for example in our simulation parameters, each stop adds at least 15s to the service time, whereas traveling through each floor adds only 1.4s), the larger difference between Queue Splitting and FCFS aids the improved system performance

Figure 3.6: Comparison of interventions using highest reversal floor for our large building case study



Note. We run 100 independent random scenarios and report the average performance. **A)** The percentage of elevator trips with different highest reversal floor H across interventions in the simulation in Section 3.3. **B)** Reporting average and median highest reversal floor for all interventions.

of Queue Splitting. The overall round trip time is still shorter for Cohorting because the number of stops tends to be lower than 2 Queue Split (see Figure 3.10 in the Appendix). The main reason is that it is unlikely to find another 3 people going to the same floor as the first passenger, so that the final elevator trip may mix traffic to high and low floors. To illustrate this, we construct a toy example here. Suppose the destinations of the current queue are 3, 4, 11, 12, 4, 14, 15, and 5. If we group floor 2 to 10 and floor 11 to 19, then the next two elevators will stop at floor 3, 4, 4, 5 and floor 11, 12, 14, 15, respectively. The highest reversal floor is 5 and 15. If Cohorting is implemented, then the first elevator will go to floor 3,4,4,11, and the second elevator will go to floor 12, 14, 15, 5. The highest reversal floor is 11 and 15. Though we have half of the passengers going to very low floors, the highest reversal floor for both elevator trips are in the higher range. In the next section,

we discuss our interventions in a more realistic and practical setting.

3.5 Practical Issues in Cohorting

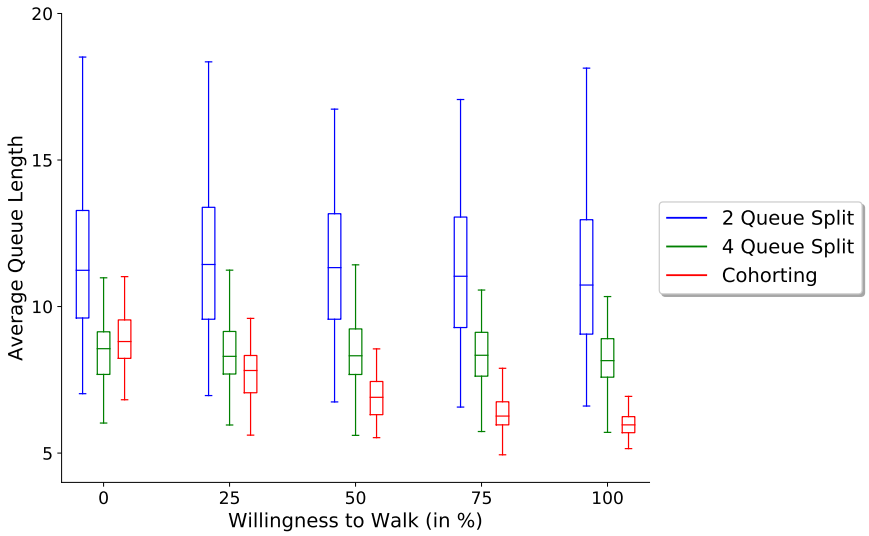
From the analysis in Section 3.4, we know that Cohorting theoretically has the best performance if there are enough passengers waiting in the lobby so that we can always find a full cohort of passengers to the same destination floor, i.e. the elevator only makes one stop in every trip. However, from the simulation results in Section 3.3, we can observe that creating the perfect cohort is quite unlikely so that the highest reversal floor and number of stops do not follow the distribution in Lemma 3.4.3 and 3.4.4. In this section, we discuss three practice-related issues that may improve or hurt the performance of the proposed interventions.

3.5.1 The Impact of Willingness-to-Walk

First, with limited queue length, we may not find enough people to the same floor. One way to increase the chance of people go to the same floor is by promoting them to take the stairs. We model this behavior using the Willingness-to-Walk (WtW) parameter indicating the probability that a given passenger would walk one floor up or down from their intended destination instead of preferring to only going to their destination floor. For example, if $WtW = 20\%$, then 20% of all passengers whose intended destination is floor d would consider the option of taking an elevator to any of the floors $d-1, d$ or $d+1$. Our consideration of WtW is inspired by literature showing that leveraging demand-side flexibility can be effective in managing operations (Tao et al. 2020, Elmachtoub et al. 2019).

When some passengers have the willingness of walking one floor up or down to their intended destination, the system may benefit from the potential reduction in the number of stops each elevator trip needs to make. Using simulation, we can see how much value it provides for different levels of WtW on the Cohorting and Queue Splitting interventions. In the Cohorting intervention, passengers line up in a single queue and the Queue Manager asks along the line whether the passengers are going to the target floor, which is the first passenger's destination. If a passenger is going to an adjacent floor of the target floor and is willing to walk, then they would say yes and join the cohort with the first passenger. In the Queue Splitting intervention, we assume that a passenger who is willing to walk will choose the shortest queue to join upon arrival, which can either go to the final destination directly, or stop at one floor lower or higher. We summarize the results in Figure 3.7.

Figure 3.7: Average queue length in the lobby v.s. Willingness-to-Walk.



Note. In the 4 queue split intervention, we divide the floor ranges equally among all the queues. The queue length under the Default FCFS intervention is on average 62 passengers across the 100 random scenarios.

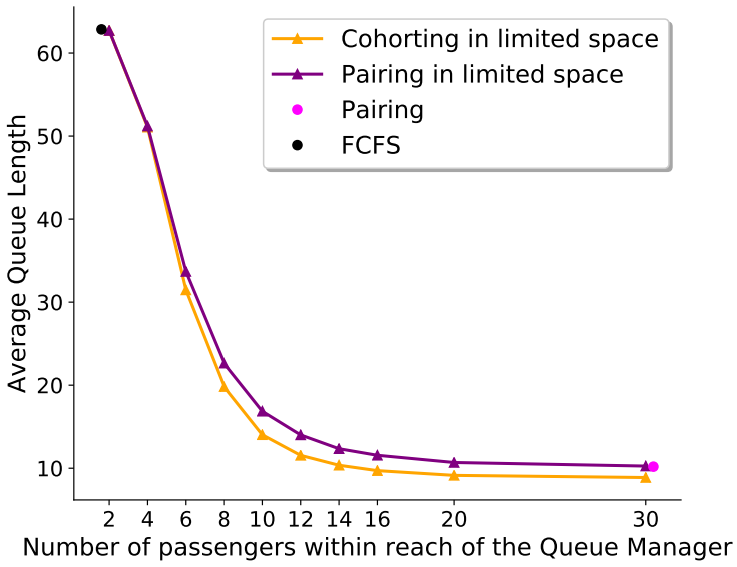
In Figure 3.7, we first observe that even with no passenger willing to walk ($WtW = 0$), all the three proposed interventions can dramatically reduce the queue length, as the average

queue length for FCFS is 62 while that for the three interventions are around 10. When $WtW = 0.25$, then Cohorting can be improved an additional $10-20\%$, while if $WtW = 100\%$ (which is idealistic), Cohorting can be improved by up to $30-40\%$. There is barely no change in the queue length when increasing WtW from 0% to 100% for Queue Splitting, though the performance varies slightly due to randomness in the 100 simulated days. This may due to the fact that the distribution of the highest reversal floor and the number of stops do not change much when we allow passengers whose destination is at the boundary of the queue ranges switching to another queue. Overall, we believe the willingness of passengers to walk one flight can improve the performance if Cohorting is implemented. However, we believe that it is rather a secondary effect since the benefit is marginal (comparing to the queue length decrease one can observe by solely using Cohorting, which is approximately 62 to 9). We note that the effect may be overestimated since passengers may not comply with their willingness to walk a flight once they board, since they can easily push the floor button they desire. Also, it may make extra time for communication when walking is included in the operations.

3.5.2 Limited Space and Communication Time

The second practical issue relates to the number of passengers the QM can reach. In a particular building, the QM may not be able to communicate with everyone in the line. One natural issue is that this flow of information may not be practically efficient or possible. The Queue Manager cannot reach out to people beyond a point, due to a turn in the hallway or the small size of the lobby. We reevaluate the Cohorting intervention with an extra

Figure 3.8: Performance of the Cohorting intervention with practical considerations.



Note. We consider two practical issues: (1) limited number of passengers within reach of the QM, and (2) only finding another passenger to be paired with the first passenger. The black, red and pink dot in the graph are performance benchmarks without the limit on the number of passengers the QM can talk to.

constraint, that the Queue Manager can only consider a certain number of passengers from the front of the queue.

The final practical issue is the extra time Cohorting may take due to the communication time it takes for the QM to learn about the passengers' destination. To simplify the Cohorting implementation in this simulation, we propose the Cohorting with Pairing intervention, which only requires the Queue Manager to find one other passenger with the same destination as the first person and create a "pair" to board the same elevator. In the original Cohorting intervention, the QM tries to match up to $C - 1$ people with the first person in line. Loading an elevator of capacity 4 with one pair leads to at most 3 stops, and two pairs leads to 2 stops being made by the elevator. A pseudocode implementation of the Cohorting with Pairing intervention is available in the Appendix Section 3.7.1.

In Figure 3.8, we show our results for the large building the Cohorting and Cohorting with Pairing intervention with a limited number of people within reach of the QM. As comparison benchmarks, we also plot the performance of the FCFS, Cohorting, and Cohorting with Pairing intervention without the constraint on the number of passengers that can be considered by the QM. The first observation is that Cohorting with Pairing is an effective and easy-to-implement intervention, as it performs almost as good as the Cohorting intervention when the QM can reach to the same number of passengers. Moreover, as the QM can approach passengers further in the queue, the average queue length shrinks rapidly. When the Queue Manager can reach out to about 10 people, the queue length is already less than 20. Therefore, it is critical to design a safe queuing plan with physical distancing such that 10 people can hear the QM, while the QM can simply implement the Cohorting with Pairing intervention in the limited lobby space. The figures may vary across different buildings and our code implementation can easily be changed to analyze different settings.

3.6 Discussion and Future Work

Through this work, we combine mathematical modeling and epidemiological principles to design interventions for safely managing elevator systems amidst a pandemic. The fundamental idea behind these interventions is to try to reduce queue buildup by maximizing the number of people in an elevator trip going to the same floor (or nearby floors), which in turn reduces boarding/deboarding times as well as travel times. The interventions we study apply to generic buildings, and we have provided open-source code so other building settings can be studied. The intervention chosen by a building may depend on its particular

simulation results, physical layout, personnel, and epidemiological principles. For example, in the elevators at the large NYC building, strict six-feet distancing allows only 2 people per elevator car, but with slightly less than six-feet distancing it allows 4 people per elevator, but much shorter lobby queues. If we change the capacity to 2, the average waiting time of passengers in the lobby goes up by 600% compared to the default FCFS and with a capacity of 3, the average waiting time still goes up by 300%. Thus, slightly violating the physical distancing rule to increase elevator capacity (to 4 in our large building case study) reduces the lobby waiting time and queue lengths significantly, even with no interventions.

In this paper, we have only considered the problem of moving people upwards in a building from the lobby. Without any elevator AI, it is near-impossible to do any interventions for downward and inter-floor movement. Using sensors, it would be possible to know how many people are in each elevator, where they are going, which floors have a request, and how many people are waiting on each floor. We could then intervene, allowing us to design algorithms that balance efficiency of the system with fair waiting times, while maintaining the safety standards necessary (Pepyne and Cassandras 1997). For instance, due to the reduced elevator capacities, a passenger on a middle floor may have difficulty leaving the building during lunchtime. Every time an elevator arrives, it may be filled with passengers from higher floors. In future work, one can design algorithms that mitigate such a scenario, which is likely (and known) to occur.

There are many other considerations to be investigated. Due to perceived inequity in interventions like Cohorting which let passengers jump the queue maybe for the greater good, there could be individual frustrations (Larson 1987, Berry et al. 2002). One can also only implement an intervention when the queue length exceeds a threshold and otherwise

rely on FCFS, which reduces the overall need of a QM. The passenger arrival patterns and destinations were generally stationary and uniform, and different effect may occur otherwise. However, we note that if some floors are more popular than others, then it may actually be easier to implement Cohorting. Finally, given more data and knowledge of the internal elevator algorithms, our models could simulate inter-floor traffic more accurately.

To summarize, the social distancing requirement during a pandemic may lead to large buildup of queues in the lobby during busy periods when using FCFS. We propose various interventions with a Queue Manager to help load passengers in the lobby. Our simulations show that the Cohorting intervention leads to lower waiting time for passengers in the lobby and reduces the number of people in the lobby (queue length) significantly. If the QM cannot talk to many people in the line, we suggest the Cohorting with Pairing intervention in limited space which is easier to implement and provides similar benefits as Cohorting, as long as the QM's announcement can reach a suitable number of people in the line. We also propose the Queue Splitting intervention which implicitly groups similar passengers together to improve efficiency while needing less communication from the QM. Queue Splitting with even a small number of queues achieves comparable performance to Cohorting. The proposed interventions are effective beyond the constraints imposed by a pandemic, and thus are still useful after the pandemic to manage lobby queues.

3.7 Supplementary Material

3.7.1 Algorithms for the proposed interventions

In this section, we describe the algorithms to evaluate the proposed interventions. For all algorithms, we simulate a passenger arrival sequence as an input file, and update the evolution of the system every $\Delta t = 1$ second. T is the total time horizon, which is equal to 2 hours in our simulation. We denote $P(t)$ as the list of passengers that arrive before time t. In the Queue Split intervention, we have k queues, and we use a queue index I to indicate from which queue we should load in a round-robin fashion. $I \to I + 1$ denotes the transition to the next queue, and specifically, the $(k + 1)$-th queue is equivalent to the first queue. The set of destinations of the I-th queue is denoted as D_I.

Algorithm 2: Cohorting

$t = 0$ `// current time`

$Q = \emptyset$ `// current queue`

$\vec{F} = \vec{0}$ `// number of passengers of an elevator deboarding at each floor`

$E = \emptyset$ `// set of empty elevators in the lobby`

while $t < T$ **do**

  $t = t + \Delta t$

  Update the current queue $Q = Q \cup P(t) \backslash P(t - \Delta t) := \{p_1, p_2, \ldots, p_l\}$, where l is the length of current queue and p_1 is the first passenger in the queue

  Record queue length $N(t) = l$

  Update the elevators in lobby $E = E \cup \{e : ReturnTime(e) \in [t - \Delta t, t)\}$

  while *there exist elevators in lobby and there are passengers waiting in the lobby* **do**

    e is the first elevator in E

    while *there exist capacity in* e *and there are passengers waiting in the lobby* **do**

      Update the current queue, l is the length of current queue and p_1 is the first passenger in the queue

      $leader = p_1$

      Remove p_1 from Q and record wait time

      Update F according to p_1's destination

      $i = 2$

      while *there exists remaining capacity in the elevator and* $i \leq l$ **do**

        if *destination of* p_i *is the same as leader* **then**

          p_i enter the current elevator and record wait time

          Remove p_i from Q

          Update $\vec{F}$ according to p_i's destination

        else

          $i \to i + 1$

        end

      end

    end

    Update $ReturnTime(e) = t + RoundTripTime(\vec{F})$

    Remove elevator e from E

    Update $\vec{F} = \vec{0}$

  end

end

Algorithm 3: Queue Splitting (k queues)

t = 0 `// current time`

$Q_i = \emptyset$ for $i = 1, \ldots, k$

$\vec{F} = \vec{0}$ `// number of passengers of an elevator deboarding at each floor`

$E = \emptyset$ `// set of empty elevators in the lobby`

I = 0 `// start from the first queue`

while t_iT **do**

  $t = t + \Delta t$

  Update the current queues $Q_i = Q_i \cup \{p : p \in P(t) \backslash P(t - \Delta t), p\text{'s destination} \in D_i\}$ for $i = 1, \ldots, k$

  Record the total queue length $N(t)$;

  Update the elevators in lobby $E = E \cup \{e : ReturnTime(e) \in [t - \Delta t, t)\}$

  while *there exist elevators in lobby and there are passengers waiting in the lobby* **do**

    e is the first elevator in E

    $RemainCap = C$

    if *there are at least C passengers in queue Q_I* **then**

      Load the elevator with the first C passengers in Q_I, remove from Q_I, record wait time, and update $\vec{F}$

      $I \to I + 1$

    else

      Load the elevator with all passengers in Q_I, remove from Q_I, record wait time, and update $\vec{F}$

      $RemainCap = C - |Q_I|$

      $I \to I + 1$ `// try to load the current elevator from the next queue`

      while *there exists remaining capacity in elevator e and there are passengers in queue Q_I* **do**

        Load the elevator with up to $RemainCap$ passengers in Q_I, remove from Q_I, record wait time, and update $\vec{F}$

        $RemainCap = RemainCap - |Q_I|$

        $I \to I + 1$

      end

    end

    Update $ReturnTime(e) = t + RoundTripTime(\vec{F})$

    Remove elevator e from E

    Update $\vec{F} = \vec{0}$

  end

end

Algorithm 4: Cohorting with Pairing

$t = 0$

$Q = \emptyset$

$\vec{F} = \vec{0}$ `// number of passengers of an elevator deboarding at each floor`

$E = \emptyset$ `// set of empty elevators in the lobby`

while $t < T$ **do**

  $t = t + \Delta t$

  Update the current queue $Q = Q \cup P(t) \backslash P(t - \Delta t) := \{p_1, p_2, \ldots, p_l\}$, where l is the length of current queue and p_1 is the first passenger in the queue

  Record queue length $N(t) = l$

  Update the elevators in lobby $E = E \cup \{e : ReturnTime(e) \in [t - \Delta t, t)\}$

  while *there exist elevators in lobby and there are passengers waiting in the lobby* **do**

    e is the first elevator in E

    while *there exist capacity in* e *and there are passengers waiting in the lobby* **do**

      Update the current queue, l is the length of current queue and p_1 is the first passenger in the queue

      $leader = p_1$

      Remove p_1 from Q and record wait time

      Update $\vec{F}$ according to p_1's destination

      $i = 2$

      `// Start finding a passenger that goes to the same destination of` p_1

      if *there exists remaining capacity in the elevator* **then**

        while *destination of* p_i *is not the same as leader and* $i \leq l$ **do**

          $i \to i + 1$

        end

        if $i \leq l$ **then** `// if` $i = l + 1$`, there is no passenger to be paired with the leader`

          p_i is paired with the leader and enters the elevator

          Record wait time of p_i

          Remove p_i from Q and update $\vec{F}$ according to p_i's destination

        end

      end

    end

    Update $ReturnTime(e) = t + RoundTripTime(\vec{F})$, remove elevator e from E, and update $\vec{F} = \vec{0}$

  end

end

3.7.2 Simulation Parameters

We summarize in Table 3.1 the parameters used in our simulations for Figures 2-8 in the paper as well as the figures in the Appendix. The parameters can be easily customized to any building in our simulation. The code for the simulation is publicly available online[1].

Parameter	Large government building in NYC	An example medium sized building	An example small building
Building Configuration			
Number of floors (m)	25	16	7
Number of elevators (N)	14	6	2
Capacity of elevators (C)	4	4	2
Elevator configuration			
Speed of elevators $T(.,.)$	1.4 sec/floor		
Speed multiplier β (coming down)	1.3, extra 30% to approximate down traffic		
Loading time $BoardingTime(.)$	15 sec to board, additional 2 sec per passenger		
Unloading time $StopTime(.)$	15 sec to deboard, additional 2 sec per passenger		
Dedication on elevators	None		
System update interval Δt	1 second		
Passenger Profile			
Number of passengers	2750	1500	400
Arrival pattern to the lobby	Poisson process between 8 AM to 10 AM (rush hour)		
Destination	Uniformly at random in 2 to 25	Uniformly at random in 2 to 16	Uniformly at random in 2 to 7
Willingness-To-Walk (WtW)	0%		

Table 3.1: Input Parameters for the simulation models

3.7.3 Results for other building types

In the main body, we primarily report the results of interventions in the large building setting. For understanding a more general performance, we also model another two examples- (1) a small building with 7 floors being served by 2 elevators with capacity 2 for 400 passengers arriving during rush hour; (2) a medium sized building with 17 floors being served by 6 elevators with capacity 4 for 1500 passengers arriving during rush hour. Figure 3.9 shows the performance in interventions in these two other examples.

[1]Code submitted for review

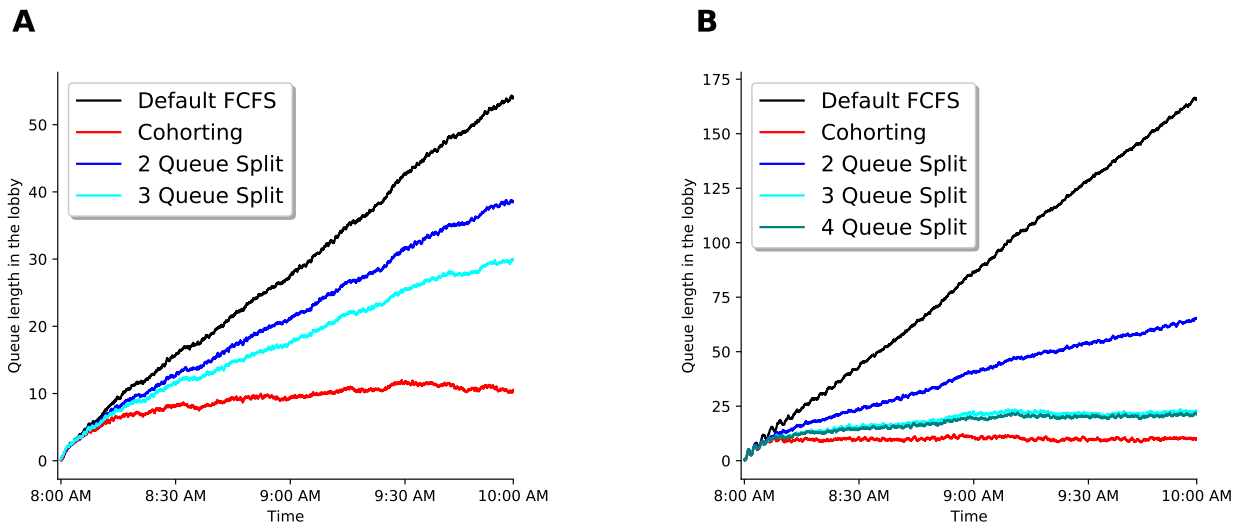
Figure 3.9 A shows that for the small building, the queue length in FCFS builds up reaching a peak of more than 50 people at the end of rush hour, whereas Cohorting has a queue length of no more than 10, which is over a 75% improvement. Queue Splitting is better than FCFS but not as beneficial as Cohorting. Queue lengths in 2 Queue Split build up to 40 people and up to 30 people in 3 Queue Split. Cohorting would be the best overall solution in this example.

Figure 3.9 B shows that for the medium sized building, the queue length in FCFS builds up reaching a peak of up to 175 people at the end of rush hour, whereas Cohorting has a queue length of around 15, which is a huge improvement. Queue Splitting is better than FCFS and the number of queues impact the performance. The queue length in 2 Queue Split steadily builds up to 60 people, whereas 3 and 4 Queue Split are better and perform similarly with only around 25 people. Thus, Cohorting or a 3 Queue Split would be good solutions in this example.

3.7.4 More quantitative metrics

In the results of our simulations in the paper, we focused on two quantitative metrics of different interventions- waiting time of a passenger at the lobby and queue length at the lobby. In this section, we analyze secondary metrics in our simulations- namely, (1) average load of the elevator, (2) average time spent by a passenger in the elevator, and (3) average number of stops made by an elevator (which corresponds to number of button presses per elevator ride). As shown in Figure 3.10, the performance of various interventions with respect to secondary metrics mimic the results in the main body.

Figure 3.9: Comparison of interventions in examples of small and medium sized buildings.

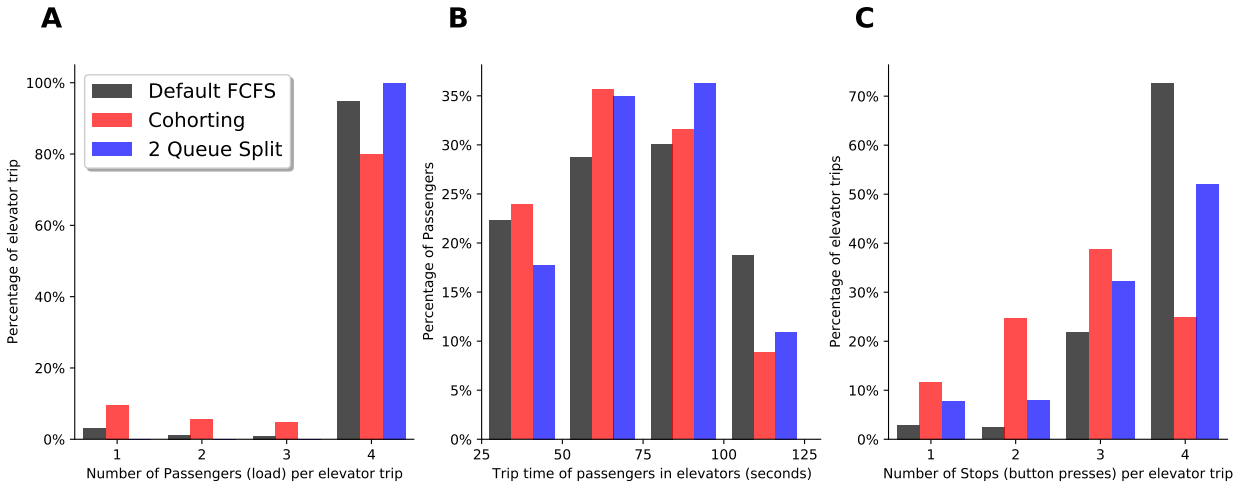


Note. In Queue splitting, we always split the floor ranges (nearly) equally among all queues. **A)** Plot of queue length in the lobby from beginning to end of the busy period across interventions for an example small building on a typical Monday morning rush hour. Between 8 to 10 AM, 400 passengers with destinations ranging from floors 2 to 7 are served by 2 elevators (each with capacity 2). **B)** Plot of queue length in the lobby from beginning to end of the busy period across interventions for an example medium building on a typical Monday morning rush hour. r. Between 8 to 10 AM, 1500 passengers with destinations ranging from floors 2 to 16 are served by 6 elevators (each with capacity 4).

In Figure 3.10.A, we plot the percentage of elevator trips with different elevator loads (number of passengers in the elevator) across interventions. Observe that in the Cohorting intervention, about 80% of elevator rides are filled up to capacity, whereas in FCFS as well as 2 Queue Splitting almost all elevators are utilized to capacity. Thus, lower queue lengths in the lobby in the Cohorting intervention leads to less elevators being filled to capacity.

In Figure 3.10.B, we plot the percentage of passengers experiencing different trip times in the elevators. In FCFS, 20% of the passengers spend between 100 to 125 seconds in the elevator, and 50% spend at least 75 seconds. Whereas in Cohorting, at least 60% of passengers ride for less than 75 seconds, with elevators making fewer stops in each trip. The 2 Queue Splitting intervention is also better than FCFS, with only 10% of the passengers

Figure 3.10: Comparison of interventions using secondary metrics for our large building case study.



*Note.*Between 8 to 10 AM, 2750 passengers with destinations ranging from floors 2 to 25 are served by 14 elevators (each with capacity 4). We run 100 independent random scenarios and report the average performance. **A)** Plot of percentage of elevator trips with different elevator loads (number of passengers) across interventions. **B)** Plot of percentage of passengers experiencing different trip times in elevators across interventions. **C)** Plot of percentage of elevator trips with different number of stops made (buttons pressed) across interventions.

spending more than 100 seconds, though a smaller fraction (50%) spend less than 75 seconds compared to Cohorting. A reduction in trip time of passengers in the elevator is beneficial, considering concerns on viral transmission inside elevators.

In Figure 3.10.C, we plot the percentage of elevator trips with different number of stops made (or equivalently the number of buttons pressed in each trip by the passengers) across interventions. In FCFS, about 75% of the trips make 4 stops. In Cohorting only about 25% of the trips make 4 stops and about 70% elevator trips make only 2 or 3 stops, which leads to shorter round trip times. 2 Queue Splitting does not perform as well as Cohorting, although it is better than FCFS, with more than 50% of the trips making 4 stops and 40% rides making only 2 or 3 stops.

3.7.5 Hard Constraints on Elevators

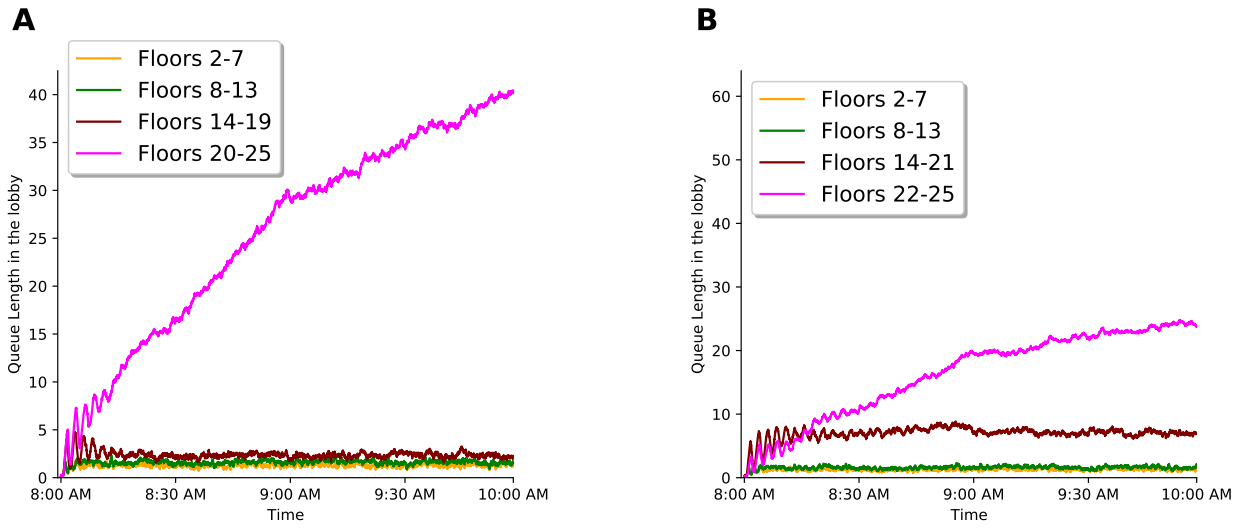
Many buildings (including the large NYC building we studied) have banks of elevators with pre-determined allocation of floor ranges for different elevators. Consider the allocation where half the elevators (7 elevators) serve half the floors $2-13$ and the other half (7 elevators) serve the floors $13-25$. Our simulations show that these *hard* constraints on elevators have an impact on many interventions.

Overall, the Cohorting and Queue Splitting interventions still perform much better than FCFS, as expected. Irrespective of interventions, passengers whose destinations are in higher floors make a big impact on performance since the round trip time of these elevators are bigger. At the very least, careful management of higher floor passengers should be considered, e.g., in queue splitting intervention, one can shorten the ranges for higher floor passengers to encourage more intrinsic Cohorting. In Figure 3.11, we consider the performance of the 4 queue split intervention by plotting the length of each of the 4 queues in the lobby over the entire busy period.

In Figure 3.11 A, we split the floor ranges $2-25$ equally among all the queues so that each queue gets exactly 6 floors. While the three lower queues have good performance (less than 5 people on average), the queue for the highest floors $20-25$ builds up over time to more than 40 people at the end of rush hour. This imbalance arises because floors $13-25$ are only served by 7 elevators, hence the passengers going to the highest floors wait longer for the busy elevators to come back to the lobby.

In Figure 3.11 B, we split the floor ranges $2-13$ equally as before, whereas in the floor range $14-25$, we assign one queue to serve eight floors $14-21$ and assign only four floors

Figure 3.11: Impact of Queue Splitting (4 queues) intervention for our large building case study.



Note. Between 8 to 10 AM, 2750 passengers with destinations ranging from floors 2 to 25 are served by 14 elevators (each with capacity 4). In this setting, not all elevators serve all floors. Instead, 7 elevators serve floors 2 to 13 and 7 serve floors 14 to 25. **(A,B)** Plots of queue length in the lobby from beginning to end of the busy period. In **A)** we evenly split the floors in the ranges (2 - 13) and (14 - 25) and each queue gets exactly 6 floors. In **B)** we evenly split the floors in the two queues serving (2 - 13) but unevenly split the two queues serving [14-25] into one serving [14-21] and the other serving (22 - 25).

$22-25$ for the last queue. The low floor queues for 2 to 13 have good performance as before, but both the high floor queues, i.e., the ones serving floors $14-21$ and $22-25$ absorb the imbalance and their lengths build up over time to at most 25 people at the end of rush hour for floors $22-25$ and at most 10 people for floors $14-21$. The queue serving $22-25$ will also have more intrinsic Cohorting since there are only four floors in this range, and the elevators taking these passengers are more likely to come back faster to the lobby. Thus tuning the floor ranges for the queues serving the higher floors leads to better performance compared to 3.11 A where all queues have their destination ranges split equally. A careful design of queue splitting is *necessary* for buildings with pre-determined floor allocations for elevators.

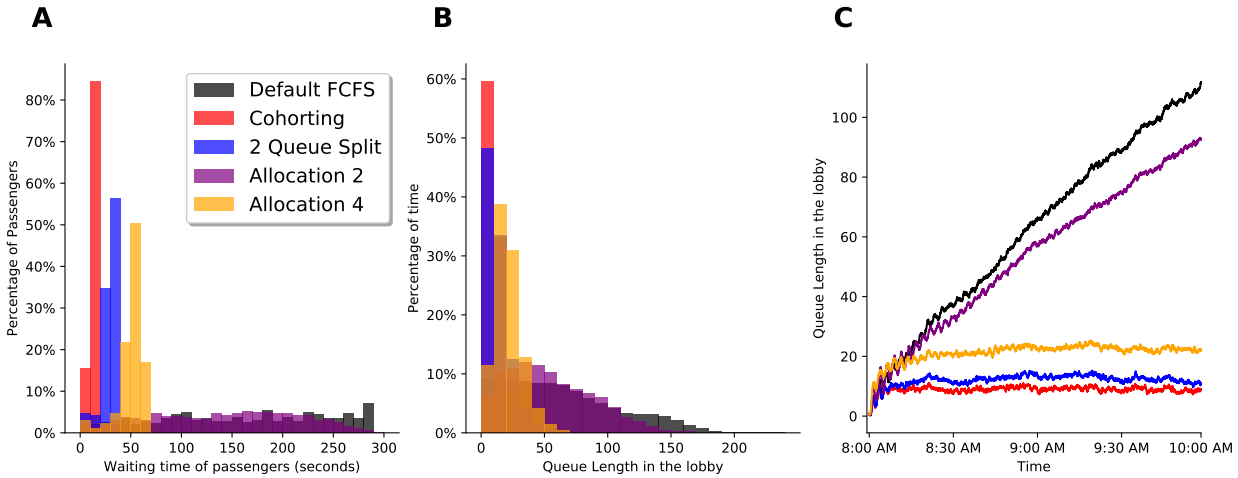
3.7.6 Performance of Allocation

We considered several distinct ways to allocate elevators to floors. Not all of them work properly. For example, we initially tested the performance of allocating half of the elevators to the odd levels and the rest to the even levels. The main motivation is to encourage people to walk up or down one level because it is always feasible for a passenger who are willing to walk to take the elevator in the other group of elevators. However, the improvement is rather negligible in comparison to FCFS even with people willing to walk. This is because the distribution of the highest reversal floor is barely changed, and it is almost as likely as FCFS that the elevator trips will end up in very high floors. Due to the poor performance, we do not describe the odd-and-even intervention in detail and do not recommend such strategies that can not reduce the average highest reversal floors.

Next, we consider the Allocation intervention that dedicate each elevator to a predetermined floor range. It is a common practice in high rise buildings that a dedicated group of elevators serves the higher floors, and other elevators serve low floors. By implementing this allocation intervention, the chances of two random passengers in the same group going to the same floor becomes relatively high, and trips to high floors are grouped together. This results in a natural cohorting phenomenon and travel time reduction. Note that Queue Splitting is theoretically better than the Allocation intervention with the same division of floor ranges, as there is an extra constraint in the usage of elevators in the Allocation intervention. We observe the drastic difference in Figure 3.12.

In the case study of the 25-floor high rise building, we numerically evaluate the performance of the Allocation 4 intervention, where we divide the 14 elevators into groups of

Figure 3.12: Comparison of interventions, including Allocation intervention for our large building case study.

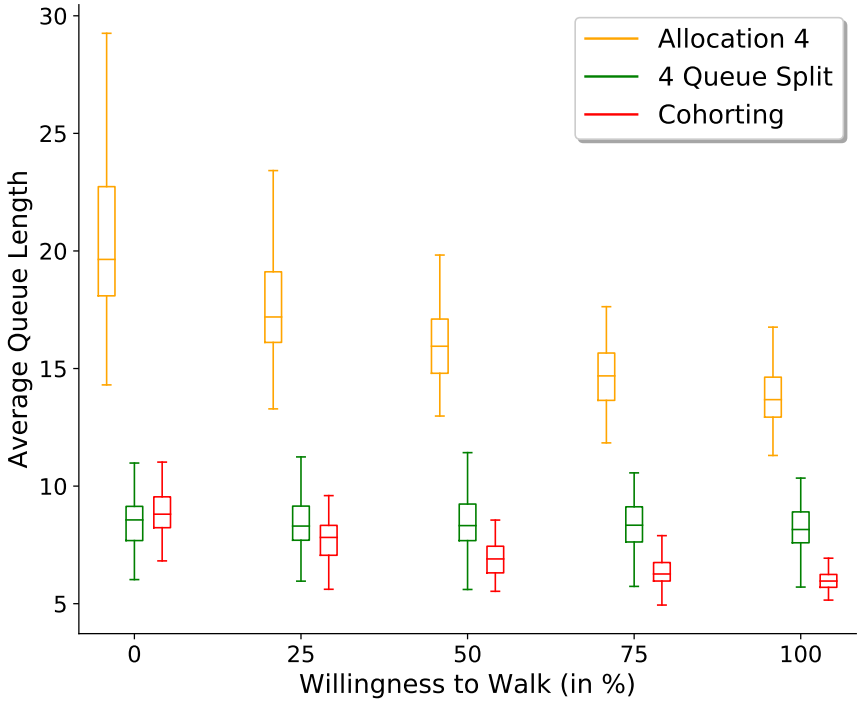


Note. Between 8 to 10 AM, 2750 passengers with destinations ranging from floors 2 to 25 are served by 14 elevators (each with capacity 4). We run 100 independent random scenarios and report the average performance. **A)** Plot of percentage of passengers experiencing different waiting times in the lobby across interventions. **B)** Plot of percentage of time different queue lengths in the lobby occur (measured every 1 second) across interventions. **C)** Plot of queue length in the lobby from beginning to end of the busy period across interventions.

$3, 3, 4, 4$, with each group serving 6 floors. The two groups with 4 elevators serve the relatively higher floor ranges. We also try the Allocation 2 intervention in which we divide the elevators into 2 groups of 7 elevators and each group serves 12 floors.

The results are shown in Figure 3.12. The Allocation 2 intervention can delay the build-up of queues slightly comparing with FCFS, but generally still have a large queue length. Using the Allocation 4 intervention, the queue length and waiting time can be reduced quite significantly, and the queue does not keep building up in the lobby. Therefore, with proper allocation of elevators, the safety concerns in elevator management can be controlled. However, in comparison with the performance of Cohorting and 2 Queue Split, the Allocation 4 intervention results in much longer queue length. In addition to the worse performance, there are higher complexity to implement the Allocation intervention as the elevators require

Figure 3.13: The impact of WtW in the Allocation 4 Intervention.



Note. Plot showing the average queue lengths in the lobby of both Cohorting and Allocation 4 interventions with the WtW parameter varying between 0% to 100%. As the WtW increases, there is negligible impact on Cohorting whereas the performance of Allocation 4 improves markedly.

extra programming and control with the relevant floor ranges. Generally speaking, using the Allocation intervention can significantly reduce the queue length, though not as much as the Cohorting and Queue Splitting intervention.

In Figure 3.13, we show the impact of WtW under the Allocation 4 intervention. Unlike the impact on Cohorting and Queue Splitting intervention, the improvement with respect to the increased WtW is much more dramatic. With no one willing to walk, the Allocation 4 intervention performs much worse than Cohorting, though significantly better than FCFS. Therefore, in a building where Allocation is physically implemented, the management team may try to ask passengers to walk up/down one level to better utilize the elevator capacities. This is an alternative way to reduce the queue length.

In the rest of the subsection, we derive the stability condition for the Allocation intervention. In the Allocation intervention where m floors and N elevators are divided into l groups, the queueing system is essentially l independent queue with FCFS service rule. The standard stability condition (3.1) for Allocation is

$$\frac{\lambda}{Cl}\mathbb{E}[\tau(H,S)|\text{group } i] < \frac{N}{l} \text{ for } i = i, \ldots, l.$$

Comparing with the stability condition of Queue Splitting, we can easily see that the stability condition is more strict for the Allocation intervention. This explains why Allocation 2 leads to an unstable queue in the simulation while 2 Queue Split leads to a stable queue with excellent performance.

3.7.7 Proof of Lemma 3.4.1

Lemma 3.4.1 is a combination of the following existing results, as we list in Lemma 3.7.1 and 3.7.2. A unitary network is a very general type of stochastic processing network. In short words, it requires a one-to-one relationship between the service activity and the buffers, and there is only one way to process jobs of any given class. The queueing system we study in this paper is a special case of a unitary network, where each job class is processed by servers from a single specified server pool, and each such service is accomplished by a single server from that pool.

Lemma 3.7.1 (Proposition 5.1 and Theorem 5.2 in Dai and Harrison (2020))**.** *If a unitary network is stable, then it satisfies the standard load condition $\rho < b$.*

Lemma 3.7.2 (Theorem 8.14 in Dai and Harrison (2020))**.** *In a feedforward queueing network, if the standard load condition holds, then the queueing network is stable under any non-idling policy.*

First, we note that the feedforward queueing network is a special case of a unitary network. Then by Lemma 3.7.1, we can conclude that if a feedforward queueing network is stable, then the standard load condition $\rho < b$ must hold. Finally, combining with Lemma 3.7.2, we can conclude that the condition $\rho < b$ is indeed a sufficient and necessary condition for the queueing network we are interested in.

Bibliography

Agatz, Niels, Ann Campbell, Moritz Fleischmann, Martin Savelsbergh. 2011. Time slot management in attended home delivery. *Transportation Science* **45**(3) 435–449.

Al-Sharif, Lutfi, Husam M Aldahiyat, Laith M Alkurdi. 2012. The use of monte carlo simulation in evaluating the elevator round trip time under up-peak traffic conditions and conventional group control. *Building Services Engineering Research and Technology* **33**(3) 319–338.

Al Sukkar, Ghazi, Lutfi Al-Sharif, Mahmoud Mansour, Mohammad Gharbieh, Esraa Farraj, Rawan Jarrah, Rasha Milekh, Noor Zaben. 2017. Reconciling the value of the elevator round trip time between calculation and simulation. *Simulation* **93**(8) 707–722.

Alexandris, NA. 1977. Statistical models in lift systems. Ph.D. thesis, The University of Manchester.

Ananthanarayanan, Sai Mali, Charles C Branas, Adam N Elmachtoub, Clifford Stein, Yeqing Zhou. 2020. Queuing safely for elevator systems amidst a pandemic. *Available at SSRN* .

Asadpour, Arash, Xuan Wang, Jiawei Zhang. 2019. Online resource allocation with limited flexibility. *Management Science* .

Atkins, Derek R, Paul O Iyogun. 1988. Periodic versus "can-order" policies for coordinated multi-item inventory systems. *Management Science* **34**(6) 791–796.

Balintfy, Joseph L. 1964. On a basic class of multi-item inventory problems. *Management science* **10**(2) 287–297.

Barney, GC, SM Dos Santos. 1975. Improved traffic design methods for lift systems. *Building Science* **10**(4) 277–285.

Barney, Gina, Lutfi Al-Sharif. 2015. *Elevator traffic handbook: theory and practice*. Routledge.

Berry, Leonard L, Kathleen Seiders, Dhruv Grewal. 2002. Understanding service convenience. *Journal of marketing* **66**(3) 1–17.

Bertsekas, Dimitri P. 2017. *Dynamic programming and optimal control*, vol. 1. Athena Scientific.

Bramson, Maury, et al. 1994. Instability of fifo queueing networks. *The Annals of Applied Probability* **4**(2) 414–431.

Campbell, Ann Melissa, Martin Savelsbergh. 2005. Decision support for consumer direct grocery initiatives. *Transportation Science* **39**(3) 313–327.

Campbell, Ann Melissa, Martin Savelsbergh. 2006. Incentive schemes for attended home delivery services. *Transportation science* **40**(3) 327–341.

Dai, J.G., J.M. Harrison. 2020. *Processing Networks: Fluid Models and Stability*. Cambridge University Press. URL `https://books.google.com/books?id=QdX7DwAAQBAJ`.

Dai, JG, Caiwei Li. 2003. Stabilizing batch-processing networks. *Operations Research* **51**(1) 123–136.

Désir, Antoine, Vineet Goyal, Yehua Wei, Jiawei Zhang. 2016. Sparse process flexibility designs: is the long chain really optimal? *Operations Research* **64**(2) 416–431.

Elmachtoub, Adam N, Michael L Hamilton. 2021. The power of opaque products in pricing. *Management Science* .

Elmachtoub, Adam N, Yehua Wei, Yeqing Zhou. 2015. Retailing with opaque products. *Available at SSRN* .

Elmachtoub, Adam N, David Yao, Yeqing Zhou. 2019. The value of flexibility from opaque selling. *Available at SSRN* .

Eppen, Gary D. 1979. Note – effects of centralization on expected costs in a multi-location newsboy problem. *Management science* **25**(5) 498–501.

Fay, S., J. Xie. 2008. Probabilistic goods: A creative way of selling products and services. *Marketing Science*, vol. 27. 674–690.

Finschi, Lukas. 2010. State-of-the-art traffic analyses. *Elevator Technology* **18** 106–115.

Fujino, Atsuya, Toshimitsu Tobita, Kazuhiro Segawa, Kenji Yoneda, Akihiro Togawa. 1997. An elevator group control system with floor-attribute control method and system optimization using genetic algorithms. *IEEE Transactions on Industrial Electronics* **44**(4) 546–552.

Gale, David, Themistocles Politof. 1981. Substitutes and complements in network flow problems. *Discrete Applied Mathematics* **3**(3) 175–186.

Gallego, Guillermo, Robert Phillips. 2004. Revenue management of flexible products. *Manufacturing & Service Operations Management* **6**(4) 321–337.

Hakonen, Henri, Marja-Liisa Siikonen. 2008. Elevator traffic simulation procedure. *Elevator World* **57**(9) 180–190.

Ignall, Edward. 1969. Optimal continuous review policies for two product inventory systems with joint setup costs. *Management Science* **15**(5) 278–283.

Jerath, Kinshuk, Serguei Netessine, Senthil K Veeraraghavan. 2009. Selling to strategic customers: Opaque selling strategies. *Consumer-driven demand and operations management models*. Springer, 253–300.

Jiang, Yabing. 2007. Price discrimination with opaque products. *Journal of Revenue and Pricing Management* **6**(2) 118–134.

Jordan, William C, Stephen C Graves. 1995. Principles on the benefits of manufacturing process flexibility. *Management Science* **41**(4) 577–594.

Köhler, Charlotte, Jan Fabian Ehmke, Ann Melissa Campbell. 2020. Flexible time window management for attended home deliveries. *Omega* **91** 102023.

Larson, Richard C. 1987. Or forum—perspectives on queues: Social justice and the psychology of queueing. *Operations research* **35**(6) 895–905.

Lee, Yutae, Tai Suk Kim, Ho-Shin Cho, Dan Keun Sung, Bong Dae Choi. 2009. Performance analysis of an elevator system during up-peak. *Mathematical and Computer modelling* **49**(3-4) 423–431.

Liu, Nan, Peter M van de Ven, Bo Zhang. 2019. Managing appointment booking under customer choices. *Management Science* .

Melchiors, Philip. 2002. Calculating can-order policies for the joint replenishment problem by the compensation approach. *European Journal of Operational Research* **141**(3) 587–595.

Pepyne, D. L., C. G. Cassandras. 1997. Optimal dispatching control for elevator systems during uppeak traffic. *IEEE Transactions on Control Systems Technology* **5**(6) 629–643.

Peres, Yuval, Kunal Talwar, Udi Wieder. 2010. The (1+ β)-choice process and weighted balls-into-bins. *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*. SIAM, 1613–1619.

Raab, Martin, Angelika Steger. 1998. "balls into bins"—a simple and tight analysis. *International Workshop on Randomization and Approximation Techniques in Computer Science*. Springer, 159–170.

Richa, Andrea W, M Mitzenmacher, R Sitaraman. 2001. The power of two random choices: A survey of techniques and results. *Combinatorial Optimization* **9** 255–304.

Ross, Sheldon. 2013. *Simulation (Fifth Edition)*. Academic Press.

Ross, Sheldon M, John J Kelly, Roger J Sullivan, William James Perry, Donald Mercer, Ruth M Davis, Thomas Dell Washburn, Earl V Sager, Joseph B Boyce, Vincent L Bristow. 1996. *Stochastic processes*, vol. 2. Wiley New York.

Shi, Cong, Yehua Wei, Yuan Zhong. 2019. Process flexibility for multiperiod production systems. *Operations Research* **67**(5) 1300–1320.

Silver, Edward A. 1965. Letter to the editor—some characteristics of a special joint-order inventory model. *Operations Research* **13**(2) 319–322.

Silver, Edward A. 1974. A control system for coordinated inventory replenishment. *International Journal of Production Research* **12**(6) 647–671.

Simchi-Levi, David, Yehua Wei. 2012. Understanding the performance of the long chain and sparse designs in process flexibility. *Operations research* **60**(5) 1125–1141.

Smith, Paige. 2020. Distancing at Reopened Offices Will Mean Long Elevator Lines : Bloomberg Law. URL `https://news.bloomberglaw.com/daily-labor-report/elevator-questions-highlight-ups-and-downs-of-reopening-offices`.

Soeffker, Ninja, Marlin W Ulmer, Dirk Mattfeld. 2017. On fairness aspects of customer acceptance mechanisms in dynamic vehicle routing. *Proceedings of Logistikmanagement* **2017** 17–24.

Ströhle, Philipp, Christoph M Flath, Johannes Gärttner. 2018. Leveraging customer flexibility for car-sharing fleet optimization. *Transportation Science* **53**(1) 42–61.

Swinarski, David. 2020. Modelling elevator traffic with social distancing in a university classroom building. *Building Services Engineering Research and Technology* **0**(0).

Tao, Shuang, Woo-Hyung Cho, Jamol Pender. 2020. The value of flexible customers via join the shortest of d queues .

Tsitsiklis, John N, Kuang Xu. 2012. On the power of (even a little) resource pooling. *Stochastic Systems* **2**(1) 1–66.

Tsitsiklis, John N, Kuang Xu. 2017. Flexible queueing architectures. *Operations Research* **65**(5) 1398–1413.

van Rijn, Cees, G Aernout Somsen, Leonard Hofstra, Ghassan Dahhan, Reinout A Bem, Stefan Kooij, Daniel Bonn. 2020. Reducing aerosol transmission of sars-cov-2 in hospital elevators. *Indoor air* **30**(6) 1065–1066.

Wang, Shixin, Xuan Wang, Jiawei Zhang. 2019. A review of flexible processes and operations. *Production and Operations Management* .

Wang, Xuan, Jiawei Zhang. 2015. Process flexibility: A distribution-free bound on the performance of k-chain. *Operations Research* **63**(3) 555–571.

Weber, Lauren. 2020. Office Elevator In COVID Times: Experts Weigh In On How To Stay Safe : Shots - Health News : NPR. URL `https://www.npr.org/sections/health-shots/`

2020/06/08/869595720/the-office-elevator-in-covid-times-experts-weigh-in-on-safer-ups-and-downs.

Wilson, Michael. 2020. It Might Become the Scariest Part of Your Commute: The Elevator: New York Times. URL https://www.nytimes.com/2020/10/26/nyregion/new-york-city-elevators-coronavirus.html.

Xiao, Yongbo, Jian Chen. 2014. Evaluating the potential effects from probabilistic selling of similar products. *Naval Research Logistics (NRL)* **61**(8) 604–620.

Xu, Zhen, Hailun Zhang, Jiheng Zhang, Rachel Q Zhang. 2020. Online demand fulfillment under limited flexibility. *Management Science* **66**(10) 4667–4685.

Yang, Xinan, Arne K Strauss, Christine SM Currie, Richard Eglese. 2014. Choice-based demand management and vehicle routing in e-fulfillment. *Transportation science* **50**(2) 473–488.