Capacity Planning for Resource Turnaround Operations

(Authors' names blinded for peer review)

Problem Definition: Many shared resources, such as hotel rooms or rental cars, require cleaning, charging, or some other operation to "turn around" the resources between successive customer uses. We study staffing and shift planning decisions for the turnaround service capacity in order to minimize the sum of customer waiting and staffing costs. Random customer departures, random customer arrivals, and worker shifts with breaks add to the managerial challenge.

Methodology/Results: Using the frameworks of diminishing return submodularity and M-convexity, we demonstrate analytical properties for capacity decisions in three staffing scenarios, including our primary model that focuses on shift planning. We propose a solution heuristic that efficiently provides near-optimal solutions. We illustrate the value of our model for hotel housekeeping operations using data from a large city-center hotel. Reallocating some room attendants to different shift start times, especially later in the day compared to current practice, can effectively eliminate guest waiting after the posted check-in time.

Managerial Implications: Hotels can reduce room attendant idleness and room readiness issues by departing from the common industry practice of all workers starting at 8:00 am. Simply having two shift start times in the morning may virtually eliminate waiting and help in recruiting and retaining workers.

Key words: Staffing; service operations; hotel housekeeping; discrete convexity; diminishing return submodularity

1. Introduction

Some service systems include two types of resources: physical assets used by customers and servers that "turn around" those assets after use to prepare them for the next customer. Hotel rooms, hospital beds, and rental cars may require cleaning, recharging, or some other form of servicing between successive customer uses. In these settings, serving customers requires managing both the asset rented to customers and the service capacity to restore the asset to a serviceable state amid uncertainty about customer arrival and departure times. As one prominent example, hotels rely on their housekeeping workforce to clean rooms between and during guest stays. This labor-intensive operation is critical to avoid the service failure of not having a room ready when promised. In this paper, we study shift planning and other staffing decisions for the workforce that turns around a resource between customer usage intervals.

While we focus on our model's value for hotel housekeeping, additional applications to which our model is relevant include cleaning operations for rental cars, planes, and hospital beds, as well as charging operations for rented resources like electric vehicles. With the increasing adoption of electric vehicles, car rental companies must choose the charging capacity at their locations in a way that balances the cost of charging stations and the cost of customer waiting (Naughton 2021). For hospital emergency departments, studies such as Pellicone and Martocci (2006) and Patel et al. (2014) have connected the management of housekeepers and bed cleaning processes with admission delays.

Through modeling flows of departing and arriving customers over a finite horizon, our model captures the key trade-off between staffing costs and wait time issues for arriving customers caused by resources not being ready. For hotels, Kandampully and Suhartanto (2003) report that the performance of the housekeeping department, including the responsiveness of the housekeeping staff and room cleanliness, is deemed as the most significant factor for brand image and customer satisfaction. The explicit cost associated with guest waiting mostly appears to be compensation paid to guests who experienced waiting. Anecdotally, a room manager of a large center-city hotel reported to us that the hotel's standard practice was to offer a $50 food and beverage credit if a room was not available for a guest upon arrival after the stated check-in time. They increased compensation of one-half that day's room rate for more significant delays or other special circumstances with guests. Many of these guests came to the hotel to attend weddings and related events; a delay in room readiness often meant that the guest had to change attire in hotel restrooms rather than the guest's room.

For the hotel industry, emerging trends in guest stay patterns underscore the importance of models to provide decision support for housekeeping operations. The major hotel brands, including Marriott, Hilton, and IHG, have offered various types of flexible check-in and check-out policies. Some offer this as a guaranteed amenity for loyalty program members, and some sell flexible stay policies as an opportunity to increase revenue. Hotels may also make flexible check-in and check-out times an explicit benefit: The Peninsula Hong Kong offers the "Peninsula Time" program for guests to check in as early as 6 AM and check out as late as 10 PM at no extra charge (Eaton 2020). The American Express Fine Hotels + Resorts program guarantees a 4:00 PM check-out and offers a check-in at noon based on availability; one general manager reported to the authors that approximately 10% of her hotel's guests book through this program, which can induce significant challenges in room operations. The owners of the TWA Hotel, which opened at Kennedy Airport in New York to much acclaim in 2019, announced a target of 200% occupancy — i.e., two different guests checking into and out of each room each day — to be achieved by offering short stays to airport travelers (Morris 2019). Other hotels, including some luxury resorts, simply offer early check-in and late check-out based on availability. An operations executive at a luxury resort that attracts visitors from around the world reported to us that many guests expect to have a room ready upon their arrival, regardless of posted check-in times. A deeper understanding of this operation can also lead to solutions to staffing shortages and enable alternate stay models, such as daytime stays, late check-outs, or early check-ins.

Regardless of whether a hotel uses innovative or conventional stay policies, our work identifies human resources strategies that can be helpful to a hotel struggling to find workers. We address a fundamental job design question — shift planning — for the more than 355,000 people who work as housekeepers in the hospitality industry of the United States hospitality industry (United States Bureau of Labor Statistics 2020). Speaking about the tight labor market for hotel room attendants, one general manager said, "Everyone I speak to in the industry is having trouble getting housekeeping staffed. It's always been one of the hardest jobs to fill, and harder than ever now (Weed 2019)." The trend of staffing shortages has continued in the post-pandemic environment: Total compensation paid by hotels has increased by more than 20% from 2019 to 2024, even though employment in the sector has fallen by roughly 9% (Feuer 2024). Consultants, general managers, and even one hotel company CEO have each described to us a hypothesis that devotion to a strict 8 AM shift start time for nearly all room attendants limits the ability of hotels to recruit and retain workers. In the absence of models to evaluate staffing decisions, managers may be hesitant to deviate from industry staffing conventions. Our model helps managers evaluate different staffing policies, enabling new strategies and supporting optimization efforts. In particular, using data from a large city-center hotel, we validate the strategy of allowing some workers to start later in the morning, which could help address staffing shortages. This strategy fits into a broader approach that McKinsey has described as "gigs go internal," which is "the use of internal talent in a 'gig' manner (Gupta et al. 2021)." The *Wall Street Journal* also reports an increasing trend of allowing manufacturing, warehouse, and hospitality workers to set their own shifts (Hufford 2022).

We present a discrete-time stochastic model of the staffing decisions, especially the shift construction decisions, for a service system with a workforce to turn around resources between customer uses. The objective of our model is to minimize the combined cost of staffing costs and guest waiting costs. Sample-path methods enable us to provide analytical results using diminishing return (DR-)submodularity and M$^\natural$-convexity for general departure and arrival processes. We make the following contributions:

1. *Model and framework.* We are the first to model staffing and shift construction decisions for turnaround operations. The two types of resources — assets used by customers and servers to turn them around — differentiate our model from many other queueing systems. Constraints on the availability of assets to be turned around influence server capacity planning, and the optimal strategy may be to allocate capacity in advance of demand rather than to match demand.

2. *Methodology and structural results.* We are among the first to identify DR-submodularity and M$^\natural$-convexity in service capacity planning problems. We are also the first to show M$^\natural$-convexity in an operations context by applying the local exchange axioms of Murota and Shioura (2018) to one staffing scenario we study. These approaches show promise for establishing structural properties within other service capacity management problems.

3. *Managerial insights from a numerical case study.* Using guest arrival and departure data from a large city-center hotel, we find that: (a) Reallocating some room attendants to later shift start times can nearly eliminate guest waiting for rooms after the posted check-in time. (b) The hotel can start its afternoon shifts earlier to gain flexibility in how many workers to allocate to morning versus afternoon shifts while still eliminating guest waiting. (c) Starting some workers' shifts later in the morning allows a hotel to offer workers flexibility — which could help with recruitment and retention — and improves operational performance. (d) The option to employ some room attendants for shorter shifts can reduce staffing costs when the volume of rooms to clean is high.

The remainder of this paper is organized as follows: Section 2 reviews the relevant literature. We present the model in Section 3 and establish structural properties in Section 4. We provide a solution heuristic in Section 5 and managerial insights from a numerical case study in Section 6. We conclude with a summary of our findings in Section 7.

2. Literature Review

Research on hospital bed availability that considers both discharges and admissions provides the closest context for our model of turnaround operations. Mills et al. (2021) study operational strategies that hospitals can use to manage both discharge and arrival processes to improve a hospital's ability to accommodate a surge of patients. Shi et al. (2015) demonstrate the impact of discharge times on inpatient (check-in) delay due to bed availability. Based on this study Zychlinski et al. (2020) apply a fluid model for the inpatient flow congestion problem in a hospital setting. Gaughan et al. (2015) examine the association between the number of nurses and the bed-blocking time using regression models. While these papers model bed availability, they do not explicitly model the cleaning workforce, limitations to its capacity, or shift planning decisions in any detail. Other research in health care settings has shown a similar high-level qualitative insight on the value of staggering shift start times: Sinreich and Jabali (2007) demonstrate that staggering shifts can reduce staffing hours while maintaining operational performance. Huang et al. (2021) also report that staggering shift start times can improve nurse job satisfaction.

The capacity management decisions that we study most closely resemble work in the operations management literature that simultaneously considers staffing levels and worker tour scheduling. Call center staffing motivates much of the research with this focus. A common approach to staffing decisions uses a *stationary independent period by period* (SIPP) assumption that allows staffing levels to be chosen using a series of stationary queueing models. Green et al. (2001) critique the effectiveness of SIPP approaches for call center applications. Atlason et al. (2008) propose a solution method that more explicitly models worker shifts. Despite these similarities, staffing for turnaround operations has a fundamentally different orientation: ideally, the workers service the assets in advance of customer arrivals, and an inventory of turned assets that are available to distribute can form. A SIPP-like approach allocating server capacity to match customer arrival patterns may perform poorly if turnaround times are not trivial. However, scheduling capacity too far in advance risks servers being idle due to a lack of returned assets to service.

The problem of machine breakdown and repair also relates to our work as a specific instance of a turnaround operation. To clarify the analogy in a hotel context, rooms would correspond to machines that break down (i.e., when a guest departs), and room attendants are like repair workers who render the room suitable for occupation. Neuts and Lucantoni (1979) study the relationship between the size of the repair crew and the queue length under Markovian assumptions. Moinzadeh and Aggarwal (1997) develop a production-inventory system with machine breakdowns and deterministic repair time to study the throughput of the model. Based on this work, Sabri-Laghaie et al. (2012) present search algorithms to find the optimal number of repair crews. Delasay et al. (2012) study routing decisions for a finite-population queueing system in a specific turnaround operation — shovels "serving" trucks in surface mines — that the authors connect to machine repair problems. The turnaround operations we study differ in their scale and modeling assumptions: Each customer arrival and departure pair corresponds to one asset that needs to be serviced, and we require general models of each process over a finite time horizon. Furthermore, in contrast to machine breakdown and repair models, we focus on shift planning, which is especially important for the potentially large service workforce in our model's applications.

Methodologically, we utilize a sample-path approach to analyze the system's capacity to service resources and present our results within a discrete convexity framework.

Sample path analysis has been widely applied to complicated service systems, often to support coupling results and the stochastic ordering of service policies. For example, Slaugh et al. (2016) use a sample path approach to prove structural results and identify the optimal rental unit selection rule for a system with rental inventory. Freund et al. (2022) also use a sample path approach to analyze the problem of allocating bicycles among stations in a bike-sharing system.

Our structural results utilize two frameworks, DR-submodularity and $M^\natural$-convexity, that have limited exposure to operations applications, especially service capacity management. For our most general model, we use the concept of DR-submodularity of Soma and Yoshida (2018), who build on (Nemhauser et al. 1978) to consider submodular functions defined over an integer lattice. Küçükyavuz and Yu (2023) identify various applications of DR-submodularity, such as sensor placement and social media influence propagation. For $M^\natural$-convexity, Chen and Li (2021b) discuss its application to multiproduct inventory models with substitution, and Chen and Li (2021a) also consider the framework's application to network flow problems and production problems. Zacharias and Pinedo (2017) and Zacharias and Yunes (2020) utilize discrete convexity frameworks to study the design of service systems that use appointments. Kaspi et al. (2017) prove M$^\natural$-convexity for a bivariate bike-sharing model. However, Freund et al. (2022) shows the property of M$^\natural$-convexity is not preserved in a model with unusable bikes. They instead rely on the concept of multimodularity. Shioura (2022) shows M$^\natural$-convexity for a related bicycle dock reallocation problem. Like these results, we find that M$^\natural$-convexity does not hold for the most general version of our problem but does hold with an additional assumption. To prove our result, we use the simpler local exchange axioms of Murota and Shioura (2018) — which we are the first to employ in the operations management literature — as the complexity of the state equations prohibits an approach similar to Shioura (2022).

While we are the first to provide analytical results specific to the domain of hotel housekeeping, hotel operations have received some attention in the operations literature. Bitran and Gilbert (1996) model reservation acceptance decisions and presented heuristics to decide how many rooms to allocate to "walk-in" customers. Soltani and Wilkinson (2010) study how hotels can use flexible room attendants to improve housekeeping efficiency. Chen et al. (2021) conduct a field study to show that dividing housekeeping work into "contaminated" and "sanitary" tasks improves hygiene and labor efficiency. Motivated by front desk staffing, Thompson and Goodale (2006) present a tour scheduling approach for a service workforce with heterogeneous productivity rates among workers. Sari (2017) measures room attendant performance by a fuzzy score-based model. Malony et al. (2012) and Kadry et al. (2017) simulate the housekeeping process of a hotel using discrete event simulation software to reduce customer waiting and staffing costs, but did not provide a formal model or offer managerial insights useful to a large-scale hotel. Wood et al. (2005) describe metrics for the performance of housekeeping operations through audit questionnaires.

3. Model

We represent the staffing problem to turn around resources between customer usage periods using a discrete-time model with equal-length time periods over a finite horizon. While servers are either idle or busy, each resource goes through a cycle of four states. For convenience, we label these states using terminology from a cleaning-focused operation: *occupied*, *vacant dirty*, *in-process*, and *vacant clean*. While a resource is in use, it is *occupied*. Once the customer departs and releases occupancy of a resource, the resource is *vacant dirty* and available to be serviced. Once servicing begins, the resource's state is *in-process*, and the server is unavailable until servicing is completed. The resource then becomes *vacant clean*, and the server is available to service another resource. A vacant clean resource may then be assigned to another customer. If a customer arrives and no vacant clean resources are available, the customer waits in a queue for available resources. Although we use the terminology "dirty" and "clean" for our resource states throughout this paper, the static capacity level may represent other settings in which machines — rather than scheduled labor working in

Table 1 Model Notation

Time Parameters	
T	Length of time horizon
$\beta \subseteq \{1,\ldots,T-1\}$	Periods in which servers may start a shift; i.e., $y_t = 0$ for $t \notin \beta$
N	Number of periods in which servers may start a shift, $N = \lvert\beta\rvert$
H	Number of time periods required to service a resource
L_1	Work duration of a shift from start until mid-shift break
B	Mid-shift break duration
L_2	Work duration from mid-shift break to end of shift
Sample Path	
A_t	Number of arrivals in period t (realized as a_t)
D_t	Number of departures in period t (realized as d_t)
Decision Variables and States	
$\boldsymbol{y} = (y_1,\ldots,y_{T-1})$	Number of servers starting shifts in each period
$V_t(\boldsymbol{y})$	Number of vacant dirty resources at the beginning of period t
$I_t(\boldsymbol{y}), I_0$	Inventory position for vacant clean resources at the end of period t
$W_t(\boldsymbol{y})$	Number of customers waiting for a resource at the end of period t
$S_t(\boldsymbol{y})$	Number of resources for which servicing starts in period t
$\boldsymbol{z}(\boldsymbol{y})$	Number of active servers in each period t, $\boldsymbol{z}(\boldsymbol{y}) = \{z_1(\boldsymbol{y}),\ldots,z_T(\boldsymbol{y})\}$
$R_t(\boldsymbol{y})$	Number of servers available to begin servicing a resource in period t
Costs	
k_t	Cost of one unit of capacity (i.e., shift worked) beginning in period t
b_t	Cost of waiting per customer at the end of period t
$C(\boldsymbol{y})$	Total cost for a single sample path of arrivals and departures

defined shifts — perform a necessary servicing task. For example, servers in this model could be charging stations for a car rental company that uses electric vehicles.

Throughout this paper, we assume that all resources are interchangeable and suitable for any customer. Also, any server can service any resource. All functions refer to a single sample path of customer departures and arrivals, d_t and a_t, respectively, for $t = 1,\ldots,T$; we omit the notation explicitly denoting the sample path to simplify the expressions; i.e., $f(y, a_t, d_t) := f(y)$. Additionally, we let d_0 represent the number of vacant dirty resources at the beginning of the time horizon. The objective is to minimize the total expected cost of staffing and customer waiting over all possible sample paths of arrivals and departures. All proofs appear in the online supplement. A guide to notation appears in Table 1.

To model the shift planning decision, we define the *staffing vector* as $\boldsymbol{y} = (y_1,\ldots,y_{T-1}) \in \mathbb{Z}_+^{T-1}$, where y_t is the number of worker shifts starting in period t and $\mathbb{Z}_+$ is the set of nonnegative integers. The sequence of events at the beginning of each period $t = 1,\ldots,T$ is as follows: First, we observe the number of departures, d_t, and arrivals, a_t, which are realizations of D_t and arrivals A_t, respectively. Each departure corresponds to one resource being added to the inventory of vacant dirty resources. The variable $V_t(\boldsymbol{y})$ represents the number of vacant dirty resources for a system with staffing vector $\boldsymbol{y}$ after the departures are realized and before workers are assigned to servicing the resources — a relationship defined by (1). Second, all resources that began a turnaround operation in period $t - H$ are added to the inventory of vacant clean resources, where $H \in \mathbb{N}$ represents the number of periods required to complete the turnaround operation, and $\mathbb{N}$ denotes the set of natural numbers. The servers may become available to service another resource. Third, the turnaround servicing begins for $S_t(\boldsymbol{y})$ resources, where $S_t(\boldsymbol{y})$ denotes the number of resources for which turnaround servicing starts in period t (after servers who started cleaning in period $t-H$ become available again). As defined by (2), this value is constrained by the number of available servers in period t, $R_t(\boldsymbol{y})$, and the number of vacant dirty resources, $V_t(\boldsymbol{y})$. Finally, as many cleaned resources as possible are assigned to customers, and the inventory position of cleaned resources, $I_t(\boldsymbol{y})$, updates. The inventory position at the end of period t, $I_t(\boldsymbol{y})$, is defined by (3) as the difference between the cumulative number of resources fully serviced and the cumulative number of customer arrivals by period t. We also allow $I_0 > 0$ to represent the vacant clean resources available at the beginning of the time horizon.

We use five states: $V_t(\boldsymbol{y}), S_t(\boldsymbol{y}), I_t(\boldsymbol{y}), z_t(\boldsymbol{y}), R_t(\boldsymbol{y})$ — which are the number of vacant dirt resources, number of resource turnaround operations to start, inventory position, number of on-duty staff, and number of available staff at time t, respectively — to specify the system state and the state transitions. The following state equations characterize our system:

$$V_t(\boldsymbol{y}) = d_0 + \sum_{n=1}^{t} d_n - \sum_{n=1}^{t-1} S_n(\boldsymbol{y}), \tag{1}$$

$$S_t(\boldsymbol{y}) := \min\left\{V_t(\boldsymbol{y}), R_t(\boldsymbol{y})\right\}, \tag{2}$$

$$I_t(\boldsymbol{y}) = I_0 + \sum_{n=1}^{t-H} S_n(\boldsymbol{y}) - \sum_{n=1}^{t} a_n. \tag{3}$$

In the context of many labor-intensive services, workers take a mid-shift break, which is important to consider for shift planning decisions to accurately predict when resources become vacant and clean. Servers may begin working a shift in any period $t \in \beta$, $\beta \subseteq \{1, \ldots, T-1\}$. One shift consists of three time intervals: a work period of duration $L_1 > 0$ periods before a mid-shift break, a break of duration $B \geq 0$ periods, and a work period of $L_2 \geq 0$ periods after the break. This structure means that if a server starts in period t, then the server shift concludes at the end of period $t + L_1 + B + L_2 - 1$. In addition, we require $y_t = 0$ for $t \notin \beta$. The potential difference between β and $\{1, \ldots, T-1\}$ allows for a system in which, for example, shifts may only start "on the hour" even if each time step corresponds to five minutes. For convenience in the analysis to follow, we denote the cardinality of β by N; that is, $N := |\beta|$. In particular, we use the decision vector representation $\boldsymbol{y} \in Z_+^N$, where $N = |\beta|$ is the number of allowable starting times. This representation is equivalent to the full vector representation $\boldsymbol{y} \in Z_+^{T-1}$ with the constraint that $y_t = 0$ for $t \notin \beta$, as there is a one-to-one correspondence between the two: each $\boldsymbol{y} \in Z_+^N$ defines a unique $\boldsymbol{y} \in Z_+^{T-1}$ with $y_t \geq 0$ for $t \in \beta$ and $y_t = 0$ otherwise, and vice versa.

From the staffing vector, we derive the *capacity vector* $\boldsymbol{z}(\boldsymbol{y}) = (z_1(\boldsymbol{y}), \ldots, z_T(\boldsymbol{y}))$, which is the number of "on-duty" servers in each period; i.e.,

$$z_t(\boldsymbol{y}) := \sum_{n=t-L_1+1}^{t} y_n + \sum_{n=t-L_1-B-L_2+1}^{t-L_1-B} y_n, \tag{4}$$

where the two terms are the number of active housekeepers who have yet to take their break and those who have already completed their break.

A model with turnaround service spanning multiple periods (i.e., $H > 1$) requires extra attention to work rules before the mid-shift break and at the end of the shift. Specifically, a server may begin servicing a resource in the last $H - 1$ periods before a break or the end of a shift, but that resource would need to be finished at another time and perhaps by another server. However, it is possible that another server might not be available immediately. To accommodate this situation without significant complexity from explicitly modeling individual servers and resources, we assume that the number of available servers in period t is

$$R_t(\boldsymbol{y}) := z_t(\boldsymbol{y}) - \sum_{n=t-H+1}^{t-1} S_n(\boldsymbol{y}), \tag{5}$$

which allows $R_t(\boldsymbol{y}) < 0$. The negative part of the number of available servers, $[R_t(\boldsymbol{y})]^-$, represents the deficit in the number of available servers and is the number of resources in progress that need to be completed as soon as another server becomes available. Effectively, a resource that is partially serviced and waiting for service to resume is fractionally allocated to a customer. These fractions resolve to integers once sufficient capacity becomes available. This assumption allows us to avoid a secondary server-to-resource assignment optimization problem and other algebraic complications that prohibit analytical results. We provide a simple numerical illustration and discuss this in more detail in Section 2 of the online appendix.

From (2), we express the cumulative number of starts as

$$\sum_{n=1}^{t} S_n(\boldsymbol{y}) = \sum_{n=1}^{t-1} S_n(\boldsymbol{y}) + S_t(\boldsymbol{y}) = \sum_{n=1}^{t-1} S_n(\boldsymbol{y}) + \min\left\{V_t(\boldsymbol{y}), R_t(\boldsymbol{y})\right\}.$$

Substituting using (1) and (5),

$$\sum_{n=1}^{t} S_n(\boldsymbol{y}) = \sum_{n=1}^{t-1} S_n(\boldsymbol{y}) + \min\left\{\sum_{n=0}^{t} d_n - \sum_{n=1}^{t-1} S_n(\boldsymbol{y}), z_t(\boldsymbol{y}) - \sum_{n=t-H+1}^{t-1} S_n(\boldsymbol{y})\right\}$$

which is equivalent to

$$\sum_{n=1}^{t} S_n(\boldsymbol{y}) = \min\left\{\sum_{n=0}^{t} d_n, \sum_{n=1}^{t-H} S_n(\boldsymbol{y}) + z_t(\boldsymbol{y})\right\}. \tag{6}$$

Each server whose shift begins in period $t \in \beta$ has a cost $k_t > 0$. The penalty for waiting is $b_t \geq 0$ per customer in each period $t = 1, \ldots, T$. The value of b_T may be set to be especially large to penalize instances of customers waiting at the end of the time horizon. Let $W_t(\boldsymbol{y}) = [I_t(\boldsymbol{y})]^-$ denote the number of guests waiting at the end of period t. We can then define the objective as minimizing the expectation of the cost function, $C(y)$, which represents the total cost of staffing and customer dissatisfaction from waiting:

$$\min_{\boldsymbol{y}\in\mathbb{Z}_+^{T-1}} \mathbb{E}\left[C(\boldsymbol{y})\right] = \mathbb{E}\left[\sum_{t=1}^{T} b_t W_t(\boldsymbol{y}) + \sum_{t=1}^{T-1} k_t y_t\right]. \tag{7}$$

4. Analytical Results

We first provide analytical results for the main model with server shifts in Section 4.1, which is motivated by hotel housekeeping. In Section 4.2, we provide stronger results enabled by the model with single-period shifts, which can serve as a benchmark on the value of flexible capacity. Finally, Section 4.3 includes analytical results for a turnaround operation performed by machines that provide static capacity levels over time, such as electric vehicle recharging stations.

4.1. Model with Server Shifts

Our analysis of turnaround operations with server shifts relies on the concept of *diminishing return submodularity/supermodularity* (DR-submodularity/supermodularity) to understand the connection between the cost function and the staffing decision. In the analysis to follow, let $\boldsymbol{e}_i$ be the unit vector with the i-th entry being 1. We define DR-submodularity as follows:

Definition 1. (Soma and Yoshida 2018) A function $f : \mathbb{Z}^N \to \mathbb{R} \cup \{+\infty\}$ with $\mathbf{dom} f \neq \emptyset$ is DR-submodular if $f(\boldsymbol{x} + \boldsymbol{e}_u) - f(\boldsymbol{x}) \geq f(\boldsymbol{y} + \boldsymbol{e}_u) - f(\boldsymbol{y})$ for arbitrary $\boldsymbol{x} \leq \boldsymbol{y}$ (coordinate-wise) and $u \in \{1, \ldots, V\}$.

DR-submodularity is stronger than submodularity; i.e. any DR-submodular function is also submodular. By Lemma 2.3 of Soma and Yoshida (2015), a submodular function is DR-submodular if and only if it satisfies the coordinate-wise concave condition: $f(\boldsymbol{x} + \boldsymbol{e}_u) - f(\boldsymbol{x}) \geq f(\boldsymbol{x} + 2\boldsymbol{e}_u) - f(\boldsymbol{x} + \boldsymbol{e}_u)$ for any $\boldsymbol{x}$ and $u \in \{1, \ldots, V\}$. Furthermore, if a function f is DR-submodular, its negation, $-f$, is DR-supermodular.

We use Definition 1 to show that the cumulative number of service episodes started by time period t, $\sum_{n=1}^{t} S_n(\boldsymbol{y})$, for all $t \in \{1, \ldots, T\}$ is DR-submodular in $\boldsymbol{y}$. To do so, we first demonstrate a local condition on the marginal value of the additional server. This condition is stronger than the requirement to establish DR-submodularity and is useful in the subsequent analysis.

Lemma 1. *The cumulative number of turnaround operations started by each period,* $\sum_{n=1}^{t} S_n(\boldsymbol{y})$*, satisfies the following local diminishing marginal return property:*

$$\sum_{n=1}^{t} S_n(\boldsymbol{y} + \boldsymbol{e}_i + \boldsymbol{e}_j) - \sum_{n=1}^{t} S_n(\boldsymbol{y} + \boldsymbol{e}_i) \leq \sum_{n=1}^{t} S_n(\boldsymbol{y} + \boldsymbol{e}_j) - \sum_{n=1}^{t} S_n(\boldsymbol{y}),$$

for $t \in \{1, \ldots, T\}$*,* $\boldsymbol{y} \in \mathbb{Z}_+^N$*, and* $i, j \in \beta$*.*

The proof is by induction on each time period going forward through the horizon. This result is intuitive: the value of adding an additional server, in any period, to a schedule with less capacity surpasses the value of a schedule with more capacity. Consider two schedules, $\boldsymbol{y}$ and $\boldsymbol{y}'$, which are identical except that $\boldsymbol{y}'$ has one additional server working a shift starting at some arbitrary time. The marginal value of adding one more server to the $\boldsymbol{y}$ schedule is at least as great as adding one more server at the same time to the $\boldsymbol{y}'$ schedule. We show that Lemma 1 implies DR-submodularity in Proposition 1.

Proposition 1. *The cumulative number of turnaround operations begun by any period t, $\sum_{n=1}^{t} S_n(\boldsymbol{y})$, is DR-submodular for $t \in \{1, \ldots, T\}$ and $\boldsymbol{y} \in \mathbb{Z}_+^N$ on any sample path.*

Following Proposition 1, we can show that the queue length in each period is DR-supermodular.

Lemma 2. *The number of customers waiting at the end of each period, $W_t(\boldsymbol{y})$, is DR-supermodular on any sample path.*

We extend this result to the expected total cost, $\mathbb{E}[C(\boldsymbol{y})]$, and show that it is DR-supermodular in $\boldsymbol{y}_t$. To do so, we rely on Lemma 2 and the property that DR-submodularity (supermodularity) is closed under addition (Soma and Yoshida 2018).

Proposition 2. *The total expected cost, $\mathbb{E}[C(\boldsymbol{y})]$, is DR-supermodular in the scheduling decision $\boldsymbol{y}$.*

Furthermore, we observe that the total expected cost function is a *non-monotone* DR-supermodular function with respect to the staffing decision vector $\boldsymbol{y}$, as adding more servers decreases the waiting cost but increases the labor cost. Leveraging this property, the minimization problem defined in Equation (7) can be addressed using either a pseudopolynomial-time algorithm or a slightly faster polynomial-time algorithm, both proposed by Soma and Yoshida (2017). Notably, Soma and Yoshida (2017) establish that both heuristics achieve a tight optimality rate of $\frac{1}{2}$, meaning that the performance of the heuristic solution is at most $\frac{1}{2}$ away from that of the optimal solution.

4.2. Model with Fully Flexible Capacity

A second model of resource turnaround operations assumes fully flexible capacity in which shift lengths are only one period; i.e., $L = 1$. The primary purpose of this simpler model is to illuminate the difficulties of analyzing more complicated staffing problems, such as the model in Section 4.1. In particular, it allows us to explore a discrete convexity concept that has so far been shown in very few operations applications. Certain long-duration maintenance or cleaning scenarios — such as airplane maintenance checks or cruise ship turnaround — in which a task requires approximately the entire shift also serve to motivate this model.

We first use indicator functions to characterize the marginal value of a shift when the shift length is $L = 1$.

Lemma 3. *When $L=1$, adding a server in period i has the following marginal effect on the number of customers waiting at the end of period t:*

$$W_t(\boldsymbol{y}) - W_t(\boldsymbol{y}+\boldsymbol{e}_i) = \mathbf{1}\{W_t(\boldsymbol{y})>0\}\prod_{n=i}^{t-H}\mathbf{1}\{V_n(\boldsymbol{y})>z_n(\boldsymbol{y})\}. \tag{8}$$

We next introduce M-convexity and describe the property's implications. For a vector $z\in\mathbb{Z}^N$, we denote the positive and negative supports of z by

$$\mathrm{supp}^+(z)=\{i|z_i>0\}, \qquad \mathrm{supp}^-(z)=\{j|z_j<0\}.$$

Definition 2. (M$^\natural$-convex in Murota (2021)) A function $f:\mathbb{Z}^N\to\mathbb{R}\cup\{+\infty\}$ with $\mathbf{dom}f\neq\emptyset$ is M$^\natural$-convex if, for any $\boldsymbol{y}$, $\boldsymbol{y}'\in\mathbb{Z}^N$ and $i\in\mathrm{supp}^+(\boldsymbol{y}-\boldsymbol{y}')$, we have (i)

$$f(\boldsymbol{y})+f(\boldsymbol{y}')\geq f(\boldsymbol{y}-\boldsymbol{e}_i)+f(\boldsymbol{y}'+\boldsymbol{e}_i) \tag{9}$$

or (ii) there exists some $j\in\mathrm{supp}^-(\boldsymbol{y}-\boldsymbol{y}')$ such that

$$f(\boldsymbol{y})+f(\boldsymbol{y}')\geq f(\boldsymbol{y}-\boldsymbol{e}_i+\boldsymbol{e}_j)+f(\boldsymbol{y}'+\boldsymbol{e}_i-\boldsymbol{e}_j). \tag{10}$$

This property is referred to as the *exchange property.*

An M$^\natural$-convex function, $f:\mathbb{Z}_+^N\to\mathbb{R}\cup\{+\infty\}$, has a corresponding *M-convex* function, $\tilde{f}:\mathbb{Z}_+^{N+1}\to\mathbb{R}\cup\{+\infty\}$, if its domain, $\mathbf{dom}\tilde{f}$, is contained within a hyperplane defined by $\left\{(\boldsymbol{y},y_T)\in\mathbb{Z}_+^{N+1}\mid\sum_{t=1}^T y_t=Y\right\}$ for some $Y\in\mathbb{Z}_+$ and a slack variable $y_T\in\mathbb{Z}_+$(Murota and Shioura 2018). As stated in Murota (2021), "M-convex functions and M$^\natural$-convex functions are equivalent concepts, in that M$^\natural$-convex functions in $n-1$ variables can be obtained as projections of M-convex functions in n variables." Thus, a function $f:\mathbb{Z}_+^N\to\mathbb{R}\cup\{+\infty\}$ is M$^\natural$-convex if and only if the function $\tilde{f}:\mathbb{Z}_+^{N+1}\to\mathbb{R}\cup\{+\infty\}$ defined by

$$\tilde{f}(y_0,\boldsymbol{y})=\begin{cases} f(\boldsymbol{y}) & if \quad y_0=-\sum_{t=1}^{T-1}y_t, \\ +\infty & otherwise. \end{cases} \qquad (y_0\in\mathbb{Z}) \tag{11}$$

is an M-convex function (Murota 2021). For our model, we use $y_0=y_T-Y$ so that the hyperplane constraint is $\sum_{t=1}^T y_t=Y$. Thus, by Theorem 4.1 of that same paper, the M-convexity of $\tilde{f}(y_0,\boldsymbol{y})$ corresponds to the M$^\natural$-convexity of $f(\boldsymbol{y})$.

In our context, the total server constraint restricts scheduling decisions to this hyperplane by requiring that the total number of servers equals Y, with y_T representing servers who are left "unscheduled" by assigning them a shift start time of the very last period so that any resources serviced by those servers would have no effect on the objective function. Therefore, we will show that the total cost function is M-convex in the scheduling decision $\boldsymbol{y}$. This result allows us to use the steepest descent algorithm in Shioura (2022) to find the optimal schedule that minimizes the total cost for the turnaround operation with fully flexible capacity.

Building on Definition 2, we find that a *local exchange axiom* best facilitates sample path analysis of our model given its complicated operational dynamics. Unfortunately, the local exchange axioms of Murota (2003) require simultaneous consideration of any four dimensions of $\boldsymbol{y}$, which makes the analysis prohibitively complex. However, Murota and Shioura (2018) propose simpler exchange axioms that require only three dimensions. To the best of our knowledge, we are the first to apply these axioms to an operations model with state dynamics.

Specifically, we use Theorem 4.5 of Murota and Shioura (2018), which allows us to show M$^\natural$-concavity through the following conditions:

$$\begin{aligned}
&(L1[\mathbb{Z}])\ \forall x \in \mathbb{Z}^N, \forall i,j \in \beta : f(x+\boldsymbol{e}_i+\boldsymbol{e}_j)+f(x) \leq f(x+\boldsymbol{e}_i)+f(x+\boldsymbol{e}_j)\\
&(L2[\mathbb{Z}])\ \forall x \in \mathbb{Z}^N, \forall i,j,k \in \beta \text{ with } k \notin \{i,j\}:\\
&\qquad\qquad f(x+\boldsymbol{e}_i+\boldsymbol{e}_j)+f(x+\boldsymbol{e}_k)\\
&\qquad\qquad \leq \max\left[f(x+\boldsymbol{e}_i+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_j), f(x+\boldsymbol{e}_j+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_i)\right]
\end{aligned}$$

We note that M$^\natural$-concavity is equivalent to M$^\natural$-convexity for the negative of a function, as seen by comparing (M$^\natural$-EXC[$\mathbb{Z}$]) in Murota and Shioura (2018) with (M$^\natural$-EXC) in Murota (2021). To facilitate the proof, we reformulate ($L2[\mathbb{Z}]$) in the following lemma.

Lemma 4 (***L2* Equivalency**). *L2 in Murota and Shioura (2018) holds* $\forall x \in \mathbb{Z}^N, \forall i,j,k \in \beta$ *with* $k \notin \{i,j\}$ *if and only if at least one of the following is true:*

$$f(x+\boldsymbol{e}_i+\boldsymbol{e}_j)+f(x+\boldsymbol{e}_k) = f(x+\boldsymbol{e}_i+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_j) \geq f(x+\boldsymbol{e}_j+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_i) \tag{12}$$

$$f(x+\boldsymbol{e}_i+\boldsymbol{e}_j)+f(x+\boldsymbol{e}_k) = f(x+\boldsymbol{e}_j+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_i) \geq f(x+\boldsymbol{e}_i+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_j) \tag{13}$$

$$f(x+\boldsymbol{e}_j+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_i) = f(x+\boldsymbol{e}_i+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_j) \geq f(x+\boldsymbol{e}_i+\boldsymbol{e}_j)+f(x+\boldsymbol{e}_k) \tag{14}$$

Our problem satisfies the conditions of Theorem 4.5 necessary for us to prove M$^\natural$-convexity through ($L1[\mathbb{Z}]$) and ($L2[\mathbb{Z}]$): $\boldsymbol{y} = (0,\ldots,0)$ is a feasible solution and an arbitrarily large maximum value can be chosen for each $y_i \in \boldsymbol{y}$ as the maximum number of servers to work in any time period. Showing that ($L1[\mathbb{Z}]$) holds follows from Lemma 1. We show that ($L2[\mathbb{Z}]$) holds in a specific manner: in particular, when $i \leq j \leq k$, $f(x+\boldsymbol{e}_i+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_j)$ is equal to either $f(x+\boldsymbol{e}_i+\boldsymbol{e}_j)+f(x+\boldsymbol{e}_k)$ or $f(x+\boldsymbol{e}_j+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_i)$ and greater than or equal to both of these terms.

Proposition 3. *For a fully flexible staffing model (i.e.,* $L=1$*), the number of customers waiting at the end of any period* t*,* $W_t(\boldsymbol{y})$*, is M*$^\natural$*-convex on any sample path.*

We next use Proposition 3 to show that the expected cost function is M$^\natural$-convex in $\boldsymbol{y}$. While the sum of two M$^\natural$-convex functions is not necessarily M$^\natural$-convex, we show that the $(L1[\mathbb{Z}])$ and $(L2[\mathbb{Z}])$ conditions continue to hold for the average cost over multiple sample paths.

Proposition 4. *The expected total cost, $\mathbb{E}[C(\boldsymbol{y})]$, is $M^\natural$-convex in $\boldsymbol{y}$.*

Because M$^\natural$-convex functions satisfy $(M^\natural - EXC[\mathbb{Z}])$ by definition, M-convexity follows immediately from Theorem 4.1 of Murota and Shioura (2018) by defining

$$\tilde{C}(y_0, \boldsymbol{y}) = \begin{cases} C(\boldsymbol{y}) & \text{if} \qquad y_0 = -\sum_{t=1}^{T-1} y_t, \\ +\infty & \text{otherwise.} \end{cases} \qquad (y_0 = y_T - Y) \tag{15}$$

While this step is trivial, it ensures that we can use search algorithms that require M-convexity.

Corollary 1. *The expected total cost, $\mathbb{E}\left[\tilde{C}(y_0, \boldsymbol{y})\right]$, is M-convex in $(y_0, \boldsymbol{y})$.*

However, M-convexity may fail to hold more generally, particularly when shifts have a duration greater than one period; i.e., $L > 1$.

Proposition 5. *When $L > 1$, the number of customers waiting at the end of period t, $W_t(\boldsymbol{y})$, is not necessarily $M^\natural$-convex.*

In a counterexample provided as Table 8 as part of the proof of Proposition 5 in the online supplement, the $d_1 = 9$ departures occur in the first period, and no further departures occur. With shift duration $L = 3$, the pair of sample paths with $\boldsymbol{y} = (1, 2, 2)$ and $\boldsymbol{y} = (2, 1, 1)$ starts servicing a combined total of 18 resources by the end of period 3, which is strictly more than 17 service starts from the pair of sample paths with $\boldsymbol{y} = (2, 1, 2)$ and $\boldsymbol{y} = (1, 2, 1)$. In turn, that exceeds the 16 service starts from the pair of sample paths with $\boldsymbol{y} = (2, 2, 1)$ and $\boldsymbol{y} = (1, 1, 2)$. Taken together, this violates $(L2[\mathbb{Z}])$ for the negative of the number of waiting customers at the end of period 3 if at least 18 customers have arrived by then. While M-convexity does not extend to a more general model of resource turnaround operations, this result is useful for other researchers to explore other operations applications of Murota and Shioura (2018).

4.3. Model with Static Capacity

Some technology-oriented turnaround operations, such as charging vehicles or sanitizing medical instruments, use machines with a static capacity level over time rather than human labor. We analyze a static capacity decision problem as another way to instantiate the model presented in Section 3. In this model, servers are available for all periods over the time horizon, and the decision becomes choosing $y \in \mathbb{Z}_+$, the number of servers for the system.

While the objective is to minimize the total cost, we first provide results for the relationship between waiting for and staffing level. We represent the cost of one unit of capacity for the entire horizon as $k > 0$. State equations (1) and (3) remain unchanged for the model with static capacity except for the use of variable y instead of staffing vector $\boldsymbol{y}$. However, we modify the state equation (2) for the number of service episodes to begin in a period:

$$S_t(y) := \min\left\{V_t(y), y - \sum_{n=t-H+1}^{t-1} S_n(y)\right\}, \tag{16}$$

which reflects that the number of available servers in period t is $y - \sum_{n=t-H+1}^{t-1} S_n(y)$

As in Section 4.1, we first examine the cumulative number of instances in which servicing begins for resources as a preliminary result.

Lemma 5. *For $t = 1, \ldots, T$, the cumulative number of resources for which servicing begins by periods $1, \ldots, t$, $\sum_{n=1}^{t} S_n(y)$, is discretely concave and non-decreasing in the staffing level y for any realization of departures $d_1, \ldots, d_T$.*

Lemma 5 is useful to show the relationship between the capacity level and the amount of waiting by customers in any period.

Proposition 6. *The number of customers waiting, $W_t(y)$, for an available resource at the end of each period t is discretely convex and non-increasing in the staffing level y for any realization of departures $d_1, \ldots, d_T$ and arrivals $a_1, \ldots, a_T$.*

The system's objective remains the minimization of the total cost of staffing and customer dissatisfaction, which we now express as

$$C(y) = \left(b_t \sum_{t=1}^{T} W_t(y)\right) + ky. \tag{17}$$

Proposition 7. *The expected cost, $\mathbb{E}\left[C_t(y)\right]$, is discretely convex in the staffing level y.*

Proposition 7 ensures that any local optimal solution found is also a global optimal solution and simple algorithms like steepest descent can be used to quickly find the global optimal solution.

5. Solution Methods

The shift planning and staffing problem defined in Section 3 is difficult to solve numerically. The action space can be of high dimension if the set of allowable shift start times is not trivially small. Random arrivals and departures further complicate the problem. We propose a method that utilizes the structural properties we showed in Section 4 and sample average approximation (SAA) methods to efficiently solve this stochastic discrete optimization problem. We describe the SAA method in the context of our problem in Section 5.1 and how SAA can be used in conjunction with searching algorithm in Section 5.2. A linear program (LP) to help find a good initial solution is described in Section 5.3. In Section 5.4, we report the computational performance.

5.1. Sample Average Approximation

We adopt the SAA method for stochastic discrete optimization described by Kleywegt et al. (2002) to find the optimal staffing vector. Let $\omega^1, \dots, \omega^M$ be independently and identically distributed random samples of $\{d_1, \dots, d_T\}$ and $\{a_1, \dots, a_T\}$. The sample average function is the average total cost over a fixed number of sample paths; using the notation from Section 3, we write it as

$$\hat{c}_M(\boldsymbol{y}) := \frac{1}{M} \sum_{j=1}^{M} C(\boldsymbol{y}|\omega^j).$$

Defining $\mathcal{Y} = \mathbb{Z}_+^M$ as the domain of $\boldsymbol{y}$. Let c^* and $\hat{c}^*$ denote the optimal values,

$$c^* := \min_{\boldsymbol{y} \in \mathcal{Y}} \mathbb{E}\left[C(\boldsymbol{y})\right], \qquad \hat{c}_M := \min_{\boldsymbol{y} \in \mathcal{Y}} \hat{c}_M(\boldsymbol{y}),$$

with respective sets of optimal solutions $\mathcal{S}^*$ and $\hat{\mathcal{S}}_M$. We also consider ϵ-optimal solutions $\mathcal{S}^\epsilon$ and $\hat{\mathcal{S}}_M^\epsilon$ for which $\bar{x}$ is an optimal solution if $\bar{x} \in \mathcal{S}$ and $c(\bar{x}) \leq c^* + \epsilon$. This approach allows us to relax our stochastic discrete optimization problem into a deterministic optimization problem whose solution converges to the true solution as the sample size increases. By Kleywegt et al. (2002), as the number of sample paths increases, the solution from solving the sample path converges to the optimal solution almost surely; i.e., as $M \to \infty$, $\hat{c}_M \to c^*$ with probability 1, and for any $\epsilon \geq 0$ the event $\{\hat{\mathcal{S}}_M^\epsilon \subset \mathcal{S}^\epsilon\}$ occurs with probability 1.

5.2. Search Algorithms on the Integer Lattice

In this section, we introduce a search heuristic to find solutions for the scheduling problem using the analytical properties we showed in Section 4. Since the proofs in Section 4 are based on sample path arguments, the structural properties established in Propositions 2, 4, and 7 also apply to the sample average total cost function, $\hat{c}_M(\boldsymbol{y})$. Specifically, $\hat{c}_M(\boldsymbol{y})$ is DR-supermodular in the staffing vector $\boldsymbol{y}$ under general server shift lengths. Moreover, with y_0 as a slack variable and defining $\tilde{c}_M(y_0, \boldsymbol{y})$ that relates to $\hat{c}_M(\boldsymbol{y})$ according to (15), $\hat{c}_M(y_0, \boldsymbol{y})$ is M-convex in $(y_0, \boldsymbol{y})$ when the length of the server shift is one period. The total cost function, $\hat{c}_M(y)$, is also discretely convex in y when the length of the server shift spans the entire planning horizon. Accordingly, we connect Algorithm 1 — a modified minimization algorithm for DR-supermodular functions under cardinality constraints based on Soma and Yoshida (2017) — to Algorithm 2, the steepest descent algorithm for M-convex functions from Shioura (2022). We characterize Algorithm 1 and Algorithm 2 in Section 4 of the Appendix. This combination enables us to efficiently identify near-optimal solutions for models with general shift lengths and identify the optimal solution for models with full flexible and static capacity. Our heuristic has the following steps:

1. Initiate Algorithm 1 to get the output solution $\hat{\boldsymbol{y}}_M^0$.
2. Input $\hat{\boldsymbol{y}}_M^0$ into the Algorithm 2 and get the solution $\hat{\boldsymbol{y}}_M^*$

Since $\hat{c}_M(\boldsymbol{y})$ is DR-supermodular in $\boldsymbol{y}$ under general server shift lengths, Algorithm 1 finds $\hat{\boldsymbol{y}}_M^0$ that performs at most $\frac{1}{2}$ worse than the optimal solution. For the cost function of the model with general shift lengths, the greedy nature of Algorithm 2 ensures that the performance of the incumbent solution either improves or remains unchanged with each iteration. For M-convex functions, we have a guarantee that Algorithm 2 finds an optimal solution (Shioura 2022). Murota (2004) also demonstrates another important property of the steepest descent algorithm for M-convex functions: We can find an upper bound on the number of iterations in terms of the distance to an optimal schedule, $\hat{\boldsymbol{y}}_M^*$, from the initial vector, $\hat{\boldsymbol{y}}_M^0$. Specifically, the number of iterations from $\hat{\boldsymbol{y}}_M^0$ to a global minimizer of $\hat{c}_M$ is bounded by $\frac{||\hat{\boldsymbol{y}}_M^0 - \hat{\boldsymbol{y}}_M^*||_1}{2}$. This result suggests that an initial solution close to the optimal solution reduces the number of iterations needed for the steepest descent algorithm.

5.3. A Linear Program to Choose a Starting Point and Bound the Solution

To reduce the number of steepest descent iterations, we can obtain a favorable starting point from a linear programming formulation of the problem. We seek to minimize the sample average function $\frac{1}{M}\sum_{n=1}^{M} C(\boldsymbol{y}|\omega^n)$. The state transition equations from Section 3 provide constraints.

$$\min \sum_{t=1}^{T-1} k_t y_t + \frac{1}{M}\sum_{n=1}^{M}\left(\sum_{t=1}^{T} b_t W_t^n(\boldsymbol{y})\right) \tag{18}$$

subject to

$$\begin{aligned}
&y_t \geq 0\\
&z_t^n = \sum_{i=t-L_1+1}^{t} y_i + \sum_{i=t-L_1-B-L_2+1}^{t-L_1-B} y_i,\\
&R_t^n = z_t^n - \sum_{i=t-H+1}^{t-1} S_i^n,\\
&V_t^n = \sum_{i=1}^{t} d_i^n - \sum_{i=1}^{t-1} S_i^n,\\
&S_t^n \leq V_t^n,\\
&S_t^n \leq R_t^n,\\
&I_t^n = \sum_{i=1}^{i-H} S_i^n - \sum_{i=1}^{t} a_i^n,\\
&W_t^n \geq 0,\\
&W_t^n \geq \sum_{i=1}^{t} a_i^n - \sum_{i=1}^{t-H} S_i^n,\\
&\qquad \forall t \in \{1, 2, \cdots, T\}, \quad \forall n \in \{1, 2, \cdots, M\},\\
&y_t' = 0, \quad \forall t' \notin \beta.
\end{aligned}$$

In this linear program, all integer variables are relaxed to be continuous to facilitate a quick solution. To get an admissible initial point for our heuristic, we round each value y_t to the nearest integer. We denote the solution produced by our search algorithm starting from the rounded LP-relaxation problem by $\boldsymbol{y}'_{LP}$. Then, we use it as a benchmark solution to improve the efficiency and performance of our search algorithm. Specifically, we refine our heuristic as follows:

1. Initiate Algorithm 1 to obtain the output solution $\hat{\boldsymbol{y}}^0_M$.
2. Solve the LP-relaxation problem specified by (18) to obtain rounded LP-relaxation solution $\boldsymbol{y}'_{LP}$.
3. Denote the best performing solution between $\{\hat{\boldsymbol{y}}^0_M, \boldsymbol{y}'_{LP}\}$ as $\hat{\boldsymbol{y}}'_M$; i.e., $\hat{\boldsymbol{y}}'_M = \underset{\boldsymbol{y} \in \{\hat{\boldsymbol{y}}^0_M, \boldsymbol{y}'_{LP}\}}{\operatorname{argmax}} \; \hat{c}_M(\boldsymbol{y})$.
4. Input $\hat{\boldsymbol{y}}'_M$ into the Algorithm 2 and get the solution $\hat{\boldsymbol{y}}^*_M$.

The additional heuristic steps help generate an input, $\hat{\boldsymbol{y}}'_M$, that is closer to the optimal solution, thereby reducing the number of iterations required in Algorithm 2. Since each step strictly improves the solution's performance, our heuristic maintains an optimality guarantee of $\frac{1}{2}$ for the general model and ensures the optimal solution for the fully flexible model. In Section 5.4, we use computational tests to demonstrate the effectiveness of this modification to our heuristic.

The unrounded LP-relaxation solution of problem (18), denoted by $\boldsymbol{y}^*_{LP}$, also provides a lower bound to the cost minimization problem for the server shift model. The values $\hat{c}_M(\boldsymbol{y}^*_{LP})$ and $\hat{c}_M(\boldsymbol{y}'_{LP})$ produce lower and upper bounds, respectively, of the total cost function:

$$\hat{c}_M(\boldsymbol{y}^*_{LP}) \leq \hat{c}_M(\hat{\boldsymbol{y}}^*_M) \leq \hat{c}_M(\boldsymbol{y}'_{LP}). \tag{19}$$

The distance between $\hat{c}_M(\boldsymbol{y}^*_{LP})$ and $\hat{c}_M(\boldsymbol{y}'_{LP})$ — i.e., the tightness of the bounds — provides insights on the performance of $\hat{\boldsymbol{y}}^*_M$, which is the solution found by our heuristics. In special case of $\hat{c}_M(\boldsymbol{y}^*_{LP}) = \hat{c}_M(\hat{\boldsymbol{y}}^*_M)$, $\hat{\boldsymbol{y}}^*_M$ is an optimal solution. To evaluate the performance of our heuristic, we measure the lower bound of the optimality rate of solution $\boldsymbol{y}'_{LP}$ by

$$\underline{r_{opt}} = 1 - \frac{\hat{c}_M(\hat{\boldsymbol{y}}^*_M) - \hat{c}_M(\boldsymbol{y}^*_{LP})}{\hat{c}_M(\boldsymbol{y}^*_{LP})}. \tag{20}$$

The interpretation of the optimality rate is that the performance of $\hat{\boldsymbol{y}}^*_M$' is at least $\underline{r_{opt}}$ as good as the optimal solution $\hat{c}_M(\boldsymbol{y}^*)$.

5.4. Computational Tests

In our refined solution heuristic, Algorithm 1 is a pseudopolynomial-time algorithm that demonstrates rapid convergence in practice. Conversely, while Algorithm 2 is generally effective for discrete problems (Shioura 2022), its implementation often results in longer run times because Algorithm 2 evaluates switching capacity between all combinations of possible start times. As discussed at the end of Section 5.2, reducing the number of iterations required for Algorithm 2 depends on selecting

Table 2 Steepest Descent Algorithm Solution Time by Starting Solution

Sample Paths	Rooms	Starting Solution	Number of Descents			CPU Minutes			$\underline{r_{opt}}$
			Minimum	Maximum	Average	Minimum	Maximum	Average	
100	400	Algorithm 1 Output	12	18	13.17	10.7	14.4	11.4	100%
		8 AM	13	16	13.48	11.3	13.6	11.5	100%
		LP Solution	0	2	0.79	2.0	3.3	2.5	100%
	800	Algorithm 1 Output	30	41	33.6	20.7	24.9	22.9	100%
		8 AM	33	37	33.9	22.0	23.9	23.0	100%
		LP Solution	0	3	1.74	2.13	4.6	3.2	100%
	1200	Algorithm 1 Output	42	57	45.6	22.8	37.6	26.9	98.3%
		8 AM	45	55	52.33	25.6	34.9	32.3	97.7%
		LP Solution	0	4	2.43	1.9	5.0	3.7	98.9%
300	400	Algorithm 1 Output	14	22	15.41	33.6	36.1	34.1	100%
		8 AM	20	21	20.3	36.9	37.1	37.6	100%
		LP Solution	0	2	0.81	2.2	3.7	2.9	100%
	800	Algorithm 1 Output	24	26	24.4	40.2	43.1	40.9	100%
		8 AM	27	27	27	46.3	47.6	47.0	100%
		LP Solution	0	2	0.87	2.3	2.6	3.0	100%
	1200	Algorithm 1 Output	42	48	43.2	57.1	67.4	60.4	99.8%
		8 AM	46	48	46.9	66.3	69.6	67.3	99.7%
		LP Solution	0	3	0.92	2.64	3.66	2.96	100%

a good initial solution. We computationally compare the performance and run times of Algorithm 2 under different starting solutions, including the output from Algorithm 1 and the rounded LP solution from Section 5.3. We also consider a solution based on the predominant industry heuristic of nearly all workers starting at 8 AM: the starting solution is simply for $\left\lceil \mathbb{E}\left[\sum_{n=0}^{T} A_n\right] / ((L_1 + L_2)/H) \right\rceil$ workers to start at 8 AM, where $\lceil \cdot \rceil$ is the ceiling function. We report the time-to-solution and a lower bound on the optimality rate for each starting point.

We set the total number of rooms on each sample path equal to the number of arriving guests and the number of departing guests, and all rooms must be cleaned before occupancy (i.e., $V_0 = 0$ and $I_0 = 0$). Each scenario is evaluated in batches of 30 sets of sample paths. Performance measures include the total number of descent iterations and total solution times in CPU minutes. The simulation uses parameters that will be presented in Section 6.

Table 2 shows a significant performance difference in finding the globally optimal solution when comparing the LP-based starting solution to the other two approaches. Using the LP to find a favorable starting point greatly reduces the number of descent iterations needed to reach the sample path optimal schedule: On average, approximately only one additional iteration was required when starting from the rounded LP solution to find a local optimal solution. Additionally, the solution time appears to be mostly independent as the number of rooms to clean increases. However, solution time increases with the number of rooms for the other two methods of obtaining a starting solution. Thus, the starting point recommended by the LP may be particularly appealing for larger-scale hotels.

6. Numerical Case Study

We illustrate the value of our model in the context of the daily shift planning decision for turning around — which can be expressed as "turning over" or, most commonly in the hotel context, simply

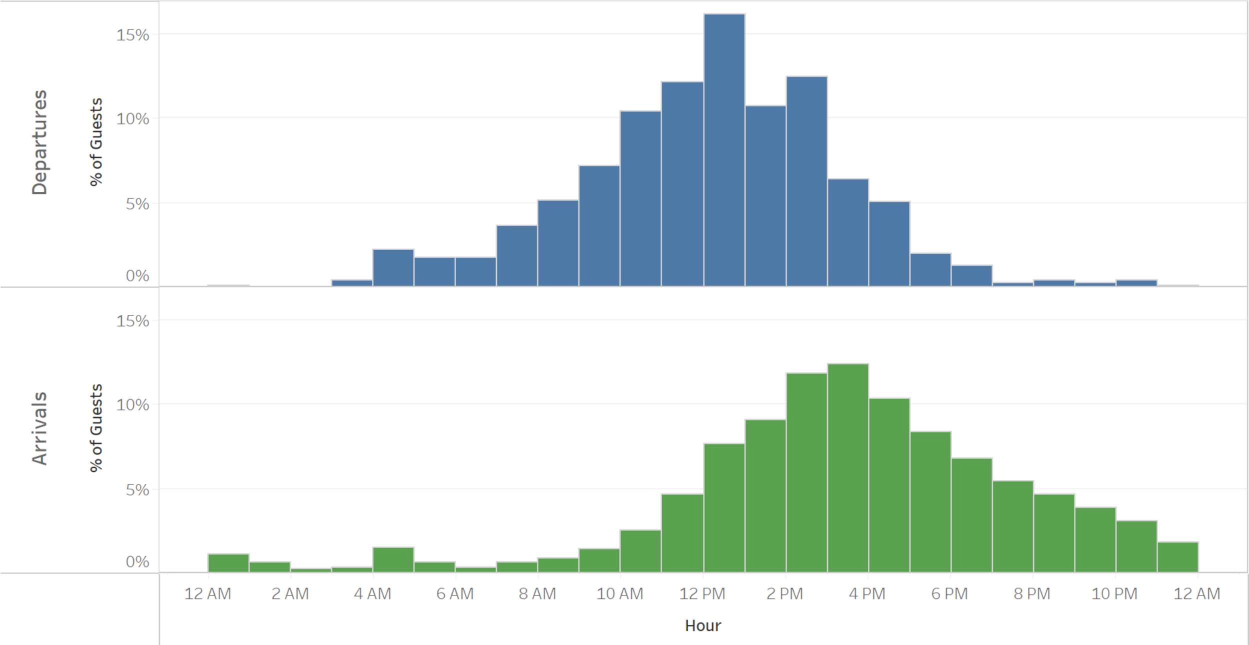


Figure 1 **Distribution of departure and arrival times for a large city-center hotel.**

"turning" — hotel rooms. Specifically, we simulate a typical "sold out" day with 1300 departures and 1300 arrivals at a large city-center hotel. Because of the high occupancy, we assume there are $d_0 = 0$ vacant dirty rooms and $I_0 = 0$ vacant clean rooms (and no waiting guests) at the beginning of the time horizon. The hotel's posted check-out and check-in times are 12:00 PM and 3:00 PM, respectively. Given the trend away from *stayover* cleaning hastened by the coronavirus pandemic so that rooms are not cleaned during the guest's stay (King 2021), we do not initially include stayover cleaning in our model. We discuss this feature further in Section 6.3.2.

Time Parameters. We use a discrete-time model with each period representing 5 minutes. Figure 1 shows the distribution of all guest departure and arrival times from October 2018 through January 2020. To minimize arrivals and departures at the beginning or end of the time horizon, we select a time horizon starting at 3:00 AM (period 0) and ending at 3:00 AM (period 287) the following day. Each departure time is independently drawn from a normal distribution with a mean of 12:10 PM (period 110) and a standard deviation of 3.17 hours (38 periods). Arrival times are similarly generated with a mean of 3:25 PM (period 149) and a standard deviation of 4.25 hours (51 periods). The resulting times are rounded to the nearest integer; any generated values outside the horizon are resampled. Because guests do not always notify the hotel upon their departure, the distribution that we use may overestimate guest waiting times; i.e., room attendants might discover vacated rooms earlier if more aggressive about knocking on doors.

Room attendants each work an eight-hour shift during which they can clean 13 rooms. With 20 minutes of preparation time at the start of the shift and 5 minutes of additional preparation time after an unpaid 30 minute mid-shift break, the structure of the shift is $(L_1, B, L_2) = (44, 7, 47)$ periods.

Each attendant has a deterministic cycle time of 35 minutes, or $H = 7$ periods, for cleaning rooms. To simplify the staffing decision and avoid having too many different shift start times, room attendants may only start their shifts in certain periods: The period in which a room attendant first enters the room must be "on the hour" within some range of the day.

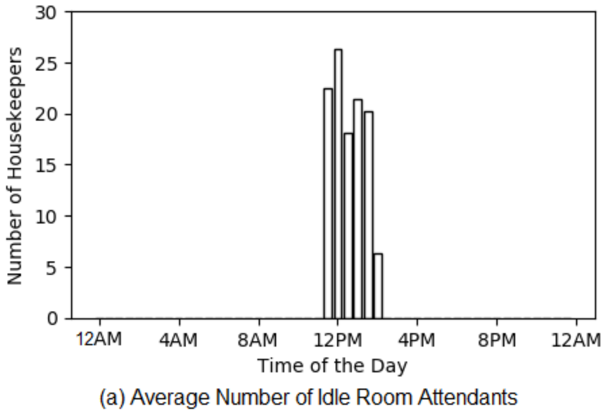
Cost Parameters. The cost of scheduling a room attendant is $k_t = \$240$ for an eight-hour shift starting in any period t. Guest waiting is not penalized before noon. From noon until 3:00 PM— i.e., $t \in \{108, \ldots, 143\}$ — the waiting cost is $b_t = \$0.60$ per period, which corresponds to \$5 per hour. Starting at the posted check-in time of 3:00 PM, it rises to \$50 per hour ($b_t = 4.17$), $t \in \{144, \ldots, 287\}$. Failing to provide a room to a guest by the end of the time horizon has a penalty of $b_{288} = \$240$ per guest, which could either represent a last-minute "rush" option for hiring extra capacity or a sufficiently high penalty so that all arriving guests get their keys by end of the horizon.

Sample Path Selection. We generate two sets of sample paths: *solving* sample paths and *testing* sample paths. We apply the heuristic introduced in Section 5.3 to the solving sample paths to solve for the optimal solution and then report the performance of this solution on the testing sample paths. We set the size of both the solving and the testing sample paths to be 100; see Section 6 of the online supplement for testing results that support this choice so that the solution found corresponds to the actual optimal solution in every combination of solving and testing sample paths.

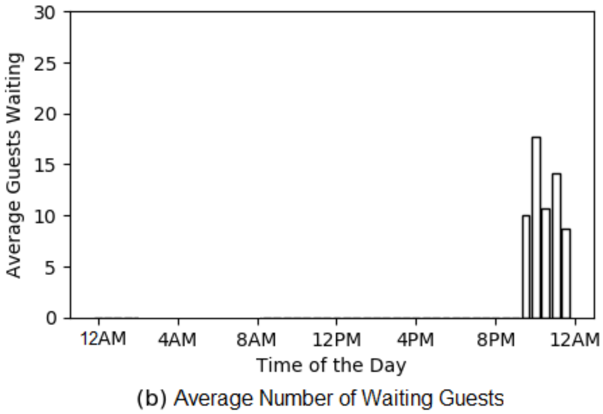
6.1. Results for Base Model

We first compare the performance of the hotel's current scheduling strategy for a typical sold-out night to that of the optimal schedule from our model under different sets of allowable shift start times. The hotel currently chooses its staffing level by dividing the number of rooms to clean by the daily cleaning capacity of a room attendant. We refer to the resulting quantity as the *minimum staffing level*. The hotel then assigns 90% of these room attendants to a morning shift (8:40 AM- 5:10 PM) and the remaining 10% to the evening shift (3:40 PM- 12:10 AM). Given 20 minutes of preparation time at the start of each shift, room attendants start cleaning their first rooms at 9:00 AM and 4:00 PM for the morning and afternoon shifts, respectively.

Figure 2 displays how this heuristic performs in expectation for a sold-out day. It reveals that room attendants sometimes run out of rooms to clean around noon and that guests sometimes have to wait for rooms in the later evening hours. The expected extra cost associated with this waiting amounts to \$2763, as reported in Table 4. Table 3 shows how the optimal staffing configuration changes based on the set of allowable shift start times and compares to starting 90 attendants at 9:00 AM and 10 attendants at 4:00 PM. While the total number of room attendants for each policy matches the minimum staffing level of 100 room attendants, the distribution of ideal start times throughout the day differs. Specifically, the results support a policy of having more attendants starting later than in



(a) Average Number of Idle Room Attendants



(b) Average Number of Waiting Guests

Figure 2 **Number of idle room attendants and guests waiting by time of day for current schedule.**

Table 3 **Optimal number of room attendants to start each hour by allowable start times.**

	Shift Start Time																	Total
	4 AM	5 AM	6 AM	7 AM	8 AM	9 AM	10 AM	11 AM	12 PM	1 PM	2 PM	3 PM	4 PM	5 PM	6 PM	7 PM	8 PM	
Current Practice	0	0	0	0	0	90	0	0	0	0	0	0	10	0	0	0	0	100
Optimal (All Periods)	1	2	0	18	17	18	9	8	9	9	8	0	0	0	1	0	0	100
Optimal (7 AM -7PM)	X	X	X	18	18	18	10	8	8	9	9	0	0	1	1	0	X	100
Optimal (8 AM - 6PM)	X	X	X	X	34	23	22	3	5	5	5	0	0	1	1	X	X	100
Optimal (9 AM - 5 PM)	X	X	X	X	X	41	23	27	2	1	2	2	2	2	X	X	X	100

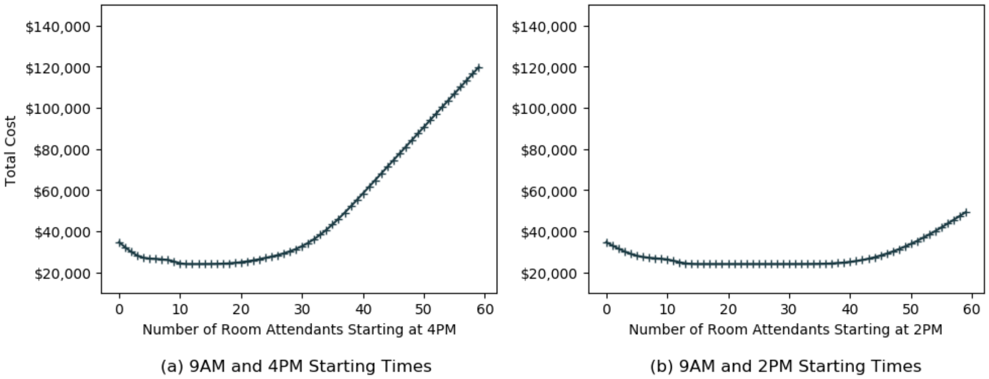
Table 4 **Performance of optimal schedules from Table 3.**

	Total Cost	Optimality Rate	Labor Cost	Waiting Cost	Waiting after 3 PM
Current Practice	$26,763	88.49%	$24,000	$2763.29	1.6 minutes
Optimal (All Periods)	$24,000	100%	$24,000	$0.29	0.0 minutes
Optimal (7AM-7PM)	$24,001	100%	$24,000	$0.54	0.0 minutes
Optimal (8AM-6PM)	$24,001	100%	$24,000	$0.78	0.0 minutes
Optimal (9AM-5PM)	$24,001	100%	$24,000	$0.99	0.1 minutes

current practice. For all four sets of allowable starting times, at least 40% of shifts are recommended to start between 10:00 AM and 2:00 PM.

Table 4 shows the performance of the schedules presented in Table 3. It reveals an opportunity to nearly eliminate the two related issues revealed in Figure 2: room attendant idleness from the late morning until the early afternoon and guest waiting later in the evening. These issues correspond to observations from managers about occasional idleness or a slower cleaning pace in the morning, as well as room readiness issues in the evening. All four optimal schedules effectively eliminate customer waiting penalties; restricting the set of allowable start times has only a very minor effect on customer waiting. By eliminating waiting, any of these policies would reduce the total cost by approximately 11%.

Figure 3 **Total cost vs. number of workers starting later shift when 100 workers are divided among an earlier and later start time.**



(a) 9AM and 4PM Starting Times

(b) 9AM and 2PM Starting Times

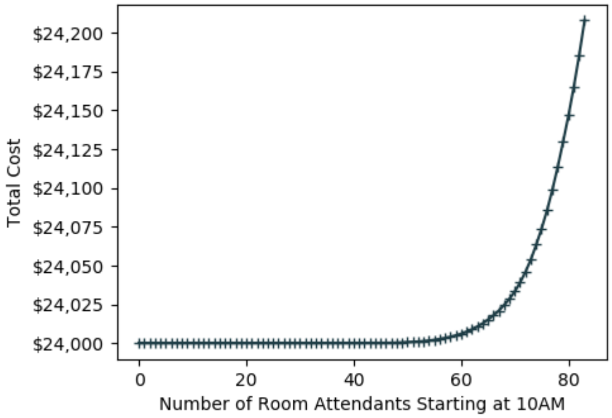
6.2. Alternate Schedules with Limited Shift Start Times

Next, we account for the practical constraint of having only a small number of shift start times, as managers may still wish to foster camaraderie and give instruction through “line up” meetings at the start of each shift. This tendency has resulted in widespread policies of all room attendants starting around 8:00 AM (or 9:00 AM, predominantly in resort-focused markets, such as South Florida) that has been reported to us by consultants and human resources directors. We have heard differing opinions about the feasibility and wisdom of dividing the room attendant workforce over more than two or three shift start times. Some have suggested that simple innovations like daily prerecorded videos by housekeeping supervisors can replace large group line-up meetings; others fear that large group meetings would be difficult to replace and that doing so might hurt the sense of belonging among room attendants.

We first focus on the current schedule with 9:00 AM and 4:00 PM as the only allowable start times and examine how to allocate 100 workers between the earlier and later start options. If too many room attendants start at 9:00 AM, room attendants might clean all vacant dirty rooms and become idle; however, if too many room attendants start at 4:00 PM, the number of guests arriving to check in might exceed the number of rooms that have been cleaned. Figure 3(a) shows that this decision appears to have a convex relationship with the cost decreasing when the number of room attendants starting at 9:00 AM is between 0 and 83 (i.e., 17 or more room attendants starting at 4:00 PM) and otherwise increases. The optimal allocation is to start with 84 attendants at 9:00 AM and 16 at 4:00 PM for a total cost of $24,010. Furthermore, starting as few as 82 and as many as 89 room attendants at 9:00 AM achieves a solution that has less than $100 in expected guest waiting costs. This analysis suggests that the hotel can achieve near-optimal performance without dramatic changes in staffing policies.

Second, we consider the impact of moving the later shift start time from 4:00 PM to 2:00 PM to understand if performance could improve while still using only two shift start times. Figure 3(b)

Figure 4 **Total cost when 16 workers start at 2:00 pm and the remaining 84 workers are divided among 9:00 am and 10:00 am start times.**



reveals that there is indeed a broader set of allocations that incur less than $100 in expected waiting costs. This corresponds to a longer flat region at the bottom of the total cost curve, which remains under $24,010 when there are between 63 and 84 workers starting at 9 AM. When workers may start at either 9:00 AM or 2:00 PM, the manager has more flexibility to allocate cleaning capacity without risking significant guest waiting for ready rooms.

Third, a conversation with the general manager of a Ritz Carlton beach resort motivated us to consider the value of two morning shift start times. While Ritz Carlton hotels are renowned for achieving strong workplace culture, this GM shared that her housekeeping department uses two shift start times — 9:00 AM and 10:00 AM— and consequently holds two "line up" meetings every morning. This practice both reduces the risk of room attendant idleness and allows her to recruit and retain workers for whom a later start time would be more convenient. Thus, we investigate the value of this practice for our model by allowing workers to start in one of three possible shift start times: 9:00 AM, 10:00 AM, or 2:00 PM. For simplicity, we fix the number of room attendants starting at 2:00 PM to 16 and vary the remaining 84 across 9:00 AM and 10:00 AM start times. Figure 4 shows how the total cost is affected by the different number of housekeepers allocated between 9:00 AM and 10:00 AM. As in the previous scenario, we observe a flat region enabling flexibility in managers' workforce decisions. The total cost when between 0 and 50 room attendants are assigned to start at 10:00 AM is unchanged. Thus, this hotel should consider offering alternate or flexible shift start times to help address the staffing shortage without worrying about its impact on room readiness for guests.

6.3. Sensitivity to Key Parameters

We test the sensitivity of our model when key parameters change to understand the robustness of our solution and insights. Specifically, we consider different cleaning times, changes to the number of rooms to clean, and changes to the shift length. A scenario in which room attendants are available to work four-hour shifts also appears in Section 5 of the Appendix.

Table 5 Optimal schedules by cleaning duration.

Cleaning Duration	Shift Start Time											Total
	8 AM	9 AM	10 AM	11 AM	12 PM	1 PM	2 PM	3 PM	4 PM	5 PM	6 PM	
30 minutes	31	19	20	7	3	4	2	0	0	1	1	88
35 minutes	34	23	22	3	5	5	5	0	0	1	1	100
40 minutes	33	26	24	7	7	4	8	0	0	1	1	111

6.3.1. Room Cleaning Duration Two trends have the potential to significantly increase or decrease the average time that it takes room attendants to clean a room: first, the move away from stayover cleaning during the pandemic might result in longer cleaning times upon a guest's departure — a prominent topic in post-pandemic labor negotiations. Second, smart robots hold the promise to work alongside room attendants to take over tasks such as vacuuming (Yang et al. 2020). To understand the impact of these changes, we show the performance of our methods when the mean cleaning duration is increased or decreased by 5 minutes (i.e., H is increased or decreased by one period). With $H = 8$, each room attendant can clean 11 rooms per shift; with $H = 6$, each room attendant can clean 15 rooms. For this analysis, room attendants may start their shifts on the hour between 8:00 AM and 6:00 PM.

Table 5 reports the optimal schedule for these three different values of the cleaning duration. While the minimum staffing requirement changes with the cleaning duration, the distribution of cleaning capacity is similar for these three scenarios: a manager should spread out the number of workers starting shifts among the morning start times to reduce the risk of idleness and assign two room attendants in the late afternoon to handle late check-outs. In all cases, the schedules provided by our model almost eliminate the waiting time after 3:00 PM, incurring less than $1 in guest waiting penalties.

6.3.2. Cleaning Load and Stayover Requirements We next evaluate the effect of changes in the total cleaning workload on the optimal workforce strategy through a stayover cleaning requirement or a change in the number of changeover cleanings, which could result from occupancy fluctuations or could be viewed as a change in the hotel size. Adding a departure at 9:00 AM when stayover cleaning begins and an arrival at the typical stayover cleaning deadline of 5:00 PM approximates the requirement to clean one stayover room in our model, especially in an opt-in stayover cleaning model for which those guests opting in may have an especially dirty room. Because this extension requires identical waiting penalties and cleaning times for both changeovers to be the same, the resulting model may be conservative in that it overestimates guest waiting. However, models with a wide range of alternate optimal solutions, such as those presented in Section 6.2, may still provide a solution with negligible waiting costs even with conservative assumptions. As in the previous section, we allow housekeepers to start every hour from 8:00 AM to 6:00 PM.

Table 6 Optimal schedules for different numbers of rooms to clean

Rooms to Clean	Schedule											Total
	8 AM	9 AM	10 AM	11 AM	12 PM	1 PM	2 PM	3 PM	4 PM	5 PM	6 PM	
700	23	12	8	2	4	4	0	0	0	0	1	54
1000	28	12	20	4	4	4	0	0	0	0	1	77
1300	34	23	22	3	5	5	5	0	0	1	1	100
1600	33	36	24	9	4	11	9	0	0	2	1	129
1900	47	29	28	21	6	14	5	1	1	2	2	156

Figure 5 Optimal Costs with Different Mean Arrival and Departure Time (Standard Deviation Unchanged)

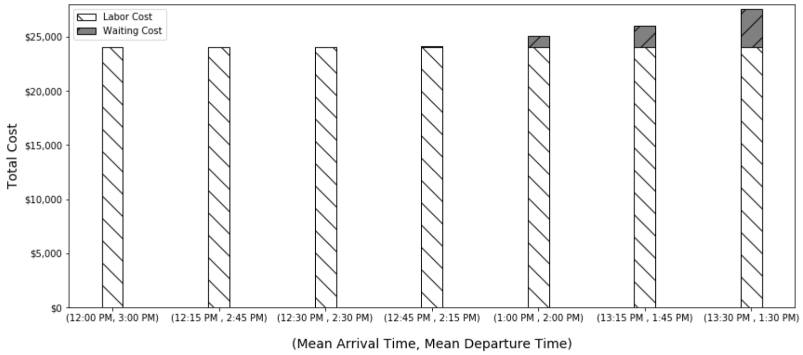


Table 6 presents the optimal schedules for different cleaning volumes. All schedules achieve a total guest waiting cost under $5 and have a similar distribution. However, when the cleaning volume exceeds 1,300 rooms, the optimal policy is to choose a staffing level higher than the minimum staffing level. This indicates a tipping point past which the optimal response to the additional cleaning requirements is to add extra workers beyond the minimum rather than incur waiting costs.

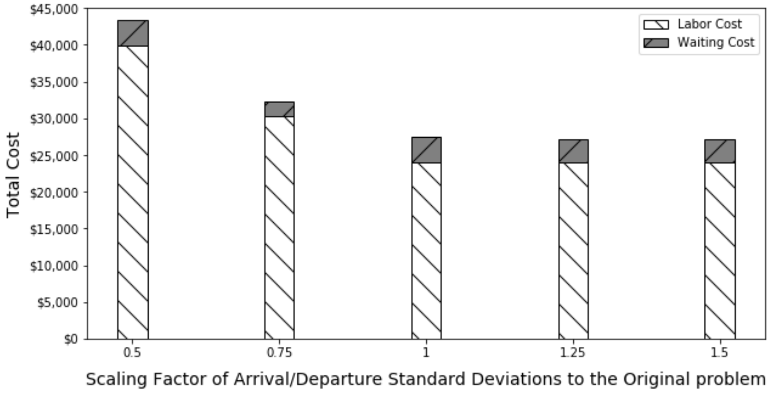
6.3.3. Tighter Turnaround Windows The difficulty of cleaning all rooms in time for guest arrivals may vary significantly by property or even by specific days at a property. Because the decision making becomes more important as the problem is more difficult, we test our model by simultaneously increasing the mean departure time and decreasing the mean arrival time by identical amounts. Specifically, we start with a mean departure time of 12:00 PM and arrival time of 3:00 PM and move those two times closer together. Table 7 shows the cost of the optimal schedules with tighter time windows for cleaning. The optimal cost and waiting time after 3:00 PM increase as the mean arrival and departure times become closer together. The results indicate that the hotel has the potential to allow or sell some late check-outs and early check-ins. However, the waiting cost increases significantly when the time between the mean departure and arrival times is around one hour or less.

Also, we study the impact of the standard deviation of the guest's arrival and departure distributions on the optimal cost. We use the particularly challenging scenario in which the mean arrival time and departure time are both 1:30 PM and scale the standard deviations for both distributions by

Table 7 Performance of optimal schedules with different cleaning windows

Mean Arrival and Departure Time	Total Cost	Optimality Rate	Labor Cost	Waiting Cost	Waiting after 3 PM
(12:00 PM and 3:00 PM)	$24,000	100%	$24,000	$0	0 minutes
(12:15 PM and 2:45 PM)	$24017.45	99.2%	$24,000	$17.45	0 minutes
(12:30 PM and 2:30 PM)	$24039.35	98.9%	$24,000	$39.35	0.1 minutes
(12:45 PM and 2:15 PM)	$24102.96	98.3%	$24,000	$102.96	0.1 minutes
(1:00 PM and 2:00 PM)	$25073.49	98.7%	$24,000	$1073.49	0.9 minutes
(1:15 PM and 1:45 PM)	$26046.12	97.2%	$24,000	$2046.12	1.4 minutes
(1:30 PM and 1:30 PM)	$27532.91	96%	$24,000	$3532.91	2.2 minutes

Figure 6 Optimal Costs with Different Standard Deviations when Mean Arrival Time and Mean Departure Time are 1:30 pm

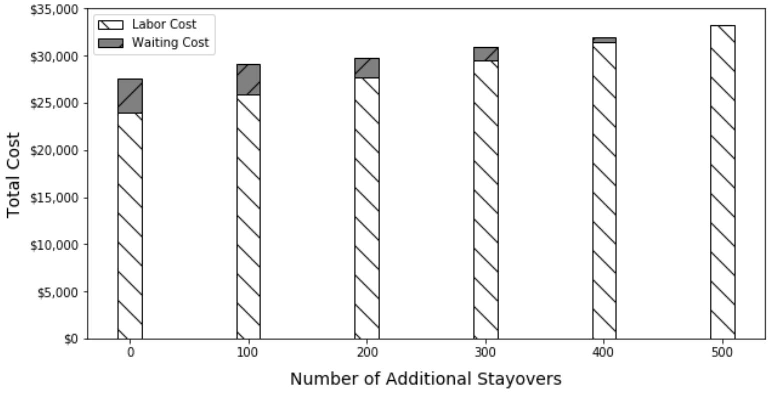


ratios of 0.5, 0.75, 1.25, and 1.5. The resulting staffing problem becomes increasingly hard as arrivals cluster together and departures cluster together. Figure 6 shows that when the standard deviations of the arrival and departure distributions are lower than the original setting, the staffing cost can increase by almost 70% as the hotel needs to overstaff to turn more rooms in the early afternoon and avoid excessive waiting later in the day.

6.4. Stayover Rooms as a Capacity Buffer

The trend away from stayover cleaning also motivates another question: does the workforce capacity to clean stayover rooms provide a buffer for changeover cleaning, thereby making the scheduling problem easier to solve? In other words, can housekeepers manage stayover cleanings throughout the day so that they can devote capacity to changeover cleanings when it is most needed? To answer this question, we use the operationally difficult model from the previous section with mean guest arrivals and departures both at 1:30 PM and add stayover cleanings. Figure 7 confirms that stayover cleanings do provide valuable capacity flexibility: guest waiting decreases as the number of stayover rooms to clean increases. Waiting almost disappears entirely when there are 500 stayover rooms to clean. Thus, hotels that remove stayover cleaning should recognize an increased risk of guest waiting.

Figure 7 **Stayover cleanings lessen the operational difficulty of the housekeeping problem for a model with mean departure and arrival times of 1:30 pm.**



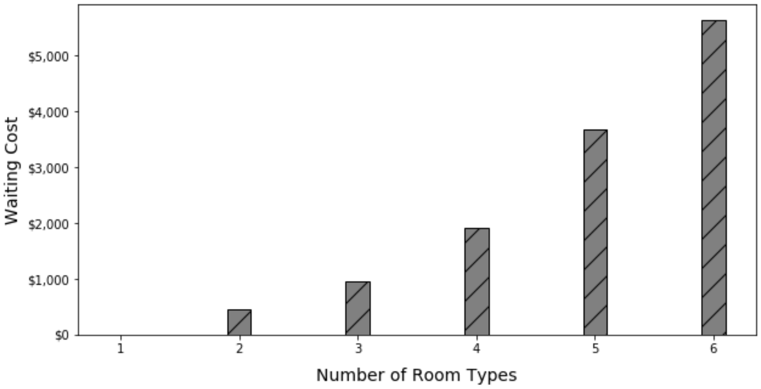
6.5. Room Types

Up to this point, our model has assumed full interchangeability of rooms among guests; in reality, hotels may have several room types, and guests reserve specific room types. Some hotels even allow guests to reserve specific rooms. We show how these restrictions may complicate the daily housekeeping problem. To do so, we divide the hotel's rooms into equally sized subsets defined by their room type. Departing and arriving customers are also randomly associated with these room types. We assume that room attendants randomly choose among the available vacant dirty rooms; thus, our estimate of guest waiting is conservative as room attendants could opportunistically select rooms to clean whose type is low on available vacant clean rooms. All other parameters match the base problem. Figure 8 shows that this fracturing of the hotel and customer population can significantly increase wait times. The guest waiting time appears to increase exponentially with the number of room types. Thus, hotels with many room types may need to invest in technology to dynamically plan room attendants' schedules and route them through the hotel over the course of a shift.

7. Conclusions and Future Research

Our models offer a new approach to support managerial decisions about cleaning or charging capacity for service operations with shared resources. In particular, they can help managers make capacity timing decisions for a cleaning workforce or charging stations to optimally navigate the trade-off between the capacity costs and the cost of customer waiting. Focusing on worker shift scheduling makes our work particularly relevant to labor-intense cleaning operations, such as hotel housekeeping, that have received little attention in the operations management literature. Besides providing a tactical decision model and solution strategy, we validate the high-level workforce strategy of flexibility by allowing some room attendants to start their shifts later in the day without risking room readiness issues.

Figure 8 Guest waiting costs increase as the number of different types of rooms increases.



Future work on hotel housekeeping and other resource turnaround operations could extend structural results and simulation models to accommodate additional problem features, such as dynamic overtime decisions or special cost penalties imposed by union contracts or local regulations. Expanding the planning horizon of the model — which would require more explicit forecasting of sales and cancellations — could also be valuable for tactical staffing decisions one or two weeks in advance. Finally, industry innovations, such as robots that work alongside room attendants or apps that allow customers to choose rooms, present related models that require deeper analysis.

As labor pressures for service systems become more acute due to rising costs and accommodations for workers, shift planning and other staffing decisions become more important for all service operations. Frameworks such as DR-submodularity and discrete convexity can enable good decision-making in many more applications.

References

Atlason J, Epelman M, Henderson S (2008) Optimizing call center staffing using simulation and analytic center cutting-plane methods. *Management Science* 54(2):295–309.

Bitran G, Gilbert S (1996) Managing hotel reservations with uncertain arrivals. *Operations Research* 44(1):35–49.

Chen X, Li M (2021a) Discrete convex analysis and its applications in operations: A survey. *Production and Operations Management* 30(6):1904–1926.

Chen X, Li M (2021b) $M^\natural$-convexity and its applications in operations. *Operations Research* 69(5):1396–1408.

Chen X, Wang R, Zhang J (2021) Divide and conquer: A hygienic, efficient, and reliable assembly line for housekeeping. *Manufacturing & Service Operations Management* .

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

Eaton M (2020) The Peninsula says goodbye to check-in and check-out times. *Hong Kong Living* URL https://hongkongliving.com/peninsula-says-goodbye-to-check-in-and-check-out-times/.

Feuer (2024) Hotel staff shortages threaten to push travel costs even higher. *Wall Street Journal* URL https://www.wsj.com/business/hospitality/hotel-staff-shortages-threaten-to-push-travel-costs-even-higher-004e5341.

Freund D, Henderson S, Shmoys D (2022) Minimizing multimodular functions and allocating capacity in bike-sharing systems. *Operations Research* 70(5):2715–2731.

Gaughan J, Gravelle H, Siciliani L (2015) Testing the bed-blocking hypothesis: does nursing and care home supply reduce delayed hospital discharges? *Health Economics* 24:32–44.

Green L, Kolesar P, Soares J (2001) Improving the SIPP approach for staffing service systems that have cyclic demands. *Operations Research* 49(4):549–564.

Gupta V, Jacobson R, Kline P, Mehndiratta M, Raabe J (2021) An on-demand revolution in customer-experience operations? URL https://www.mckinsey.com/capabilities/operations/our-insights/an-on-demand-revolution-in-customer-experience-operations, last accessed 23 September 2022.

Huang L, Tang L, Yu L, Wu Y, Huang Z, Tian X (2021) The effect of staggered shift scheduling mode on nurses in the covid-19 isolation ward-a cross sectional study. *INQUIRY: The Journal of Health Care Organization, Provision, and Financing* 58:0046958021997223.

Hufford A (2022) Choosing your own hours isn't just for remote workers anymore. *Wall Street Journal* URL https://www.wsj.com/articles/choosing-your-own-hours-isnt-just-for-remote-workers-anymore-11661938202.

Kadry S, Bagdasaryan A, Kadhum M (2017) Simulation and analysis of staff scheduling in hospitality management. *2017 7th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO)*, 1–6 (IEEE).

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

Kaspi M, Raviv T, Tzur M (2017) Bike-sharing systems: User dissatisfaction in the presence of unusable bicycles. *IISE Transactions* 49(2):144–158.

King K (2021) Housekeepers, hotel owners square off over daily service. *Wall Street Journal* URL https://www.wsj.com/articles/housekeepers-hotel-owners-square-off-over-daily-service-11635854400.

Kleywegt A, Shapiro A, Homem-de Mello T (2002) The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization* 12(2):479–502.

Küçükyavuz S, Yu Q (2023) Mixed-integer programming approaches to generalized submodular optimization and its applications. *Tutorials in Operations Research: Advancing the Frontiers of OR/MS: From Methodologies to Applications*, 1–30 (INFORMS).

Malony R, Liu D, Vincenzi D (2012) Using discrete event simulation as a supplemental teaching aid for analyzing staff scheduling. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 56, 2487–2491 (SAGE Publications Sage CA: Los Angeles, CA).

Mills A, Helm J, Wang Y (2021) Surge capacity deployment in hospitals: effectiveness of response and mitigation strategies. *Manufacturing & Service Operations Management* 23(2):367–387.

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

Morris K (2019) Twa hotel at kennedy airport targets 200% occupancy. *Wall Street Journal* URL https://www.wsj.com/articles/twa-hotel-at-kennedy-airport-targets-200-occupancy-11557843164.

Murota K (2003) *Discrete Convex Analysis*. SIAM Monographs on Discrete Mathematics and Applications (Society for Industrial and Applied Mathematics).

Murota K (2004) On steepest descent algorithms for discrete convex functions. *SIAM Journal on Optimization* 14(3):699–707.

Murota K (2021) A survey of fundamental operations on discrete convex functions of various kinds. *Optimization Methods and Software* 36(2-3):472–518.

Murota K, Shioura A (2018) Simpler exchange axioms for m-concave functions on generalized polymatroids. *Japan Journal of Industrial and Applied Mathematics* 35:235–259.

Naughton N (2021) Hertz-Tesla deal signals broad shift to EVs for rental-car companies. *Wall Street Journal* URL https://www.wsj.com/articles/hertz-tesla-deal-signals-broad-shift-to-evs-for-rental-car-companies-11638009000.

Nemhauser G, Wolsey L, Fisher M (1978) An analysis of approximations for maximizing submodular set functions—i. *Mathematical Programming* 14:265–294.

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

Patel P, Combs M, Vinson D (2014) Reduction of admit wait times: the effect of a leadership-based program. *Academic Emergency Medicine* 21(3):266–273.

Pellicone A, Martocci M (2006) Faster turnaround time. *Quality Progress* 39(3):31–36.

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

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

Shi P, Chou M, Dai J, Ding D, Sim J (2015) Models and insights for hospital inpatient operations: Time-dependent ed boarding time. *Management Science* 62(1):1–28.

Shioura A (2022) M-convex function minimization under l1-distance constraint and its application to dock reallocation in bike-sharing system. *Mathematics of Operations Research* 47(2):1566–1611.

Sinreich D, Jabali O (2007) Staggered work shifts: a way to downsize and restructure an emergency department workforce yet maintain current operational performance. *Health Care Management Science* 10(3):293–308.

Slaugh V, Biller B, Tayur S (2016) Managing rentals with usage-based loss. *Manufacturing & Service Operations Management* 18(3):429–444.

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

Soma T, Yoshida Y (2015) A generalization of submodular cover via the diminishing return property on the integer lattice. *Advances in Neural Information Processing Systems* 28.

Soma T, Yoshida Y (2017) Non-monotone dr-submodular function maximization. *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Soma T, Yoshida Y (2018) Maximizing monotone submodular functions over the integer lattice. *Mathematical Programming* 172:539–563.

Thompson G, Goodale J (2006) Variable employee productivity in workforce scheduling. *European Journal of Operational Research* 170(2):376–390.

United States Bureau of Labor Statistics (2020) Occupational employment and wage statistics: 37-2012 maids and housekeeping cleaners. URL https://www.bls.gov/oes/current/oes372012.htm, last accessed 15 February 2022.

Weed J (2019) Streamlining hotel housekeeping in a tight labor market. URL https://www.nytimes.com/2019/11/26/business/hotel-housekeepers.html.

Wood D, Moreo P, Sammons G (2005) Hotel housekeeping operational audit: A questionnaire approach. *International Journal of Hospitality & Tourism Administration* 6(3):1–10.

Yang L, Henthorne T, George B (2020) Artificial intelligence and robotics technology in the hospitality industry: Current applications and future trends. *Digital Transformation in Business and Society* 211–228.

Zacharias C, Pinedo M (2017) Managing customer arrivals in service systems with multiple identical servers. *Manufacturing & Service Operations Management* 19(4):639–656.

Zacharias C, Yunes T (2020) Multimodularity in the stochastic appointment scheduling problem with discrete arrival epochs. *Management Science* 66(2):744–763.

Zychlinski N, Mandelbaum A, Momčilović P, Cohen I (2020) Bed blocking in hospitals due to scarce capacity in geriatric institutions—cost minimization via fluid models. *Manufacturing & Service Operations Management* 22(2):396–411.

Online Supplement
Capacity Planning for Resource Turnaround Operations

1. Proofs

Proof of Lemma 1 We use an inductive argument to prove this property. First, we show that the the diminishing return property holds for $S_1(\cdot)$ as the based case. When $t=1$, we want to show

$$S_1(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j)-S_1(\boldsymbol{y}+\boldsymbol{e}_i)\leq S_1(\boldsymbol{y}+\boldsymbol{e}_j)-S_1(\boldsymbol{y}),$$

which is equivalent to

$$\min\{d_0+d_1,z_1(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j)\}-\min\{d_0+d_1,z_1(\boldsymbol{y}+\boldsymbol{e}_i)\}\leq\min\{d_0+d_1,z_1(\boldsymbol{y}+\boldsymbol{e}_j)\}-\min\{d_0+d_1,z_1(\boldsymbol{y})\}.$$

We note that $z_1(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j)\geq z_1(\boldsymbol{y}+\boldsymbol{e}_i)$ and $z_1(\boldsymbol{y}+\boldsymbol{e}_j)\geq z_1(\boldsymbol{y})$. Because $z_1(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j)-z_1(\boldsymbol{y}+\boldsymbol{e}_i)\leq 1$ and $z_1(\boldsymbol{y}+\boldsymbol{e}_j)-z_1(\boldsymbol{y})\leq 1$, we can re-write the above inequality as:

$$\mathbf{1}\{d_0+d_1\geq z_1(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j),z_1(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j)>z_1(\boldsymbol{y}+\boldsymbol{e}_i)\}\leq\mathbf{1}\{d_0+d_1\geq z_1(\boldsymbol{y}+\boldsymbol{e}_j),z_1(\boldsymbol{y}+\boldsymbol{e}_j)>z_1(\boldsymbol{y})\},$$

which holds because $d_0+d_1\geq z_1(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j)$ implies $d_0+d_1\geq z_1(\boldsymbol{y}+\boldsymbol{e}_j)$ and $z_1(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j)>z_1(\boldsymbol{y}+\boldsymbol{e}_i)$ implies $z_1(\boldsymbol{y}+\boldsymbol{e}_j)>z_1(\boldsymbol{y})$.

Assuming $\sum_{n=1}^{t'}S_n(\cdot)$ satisfies the diminishing return property for all $t'<t$, by Equation (6), we want to show the property holds for period t:

$$\begin{aligned}&\min\left\{\sum_{n=0}^{t}d_n,\sum_{n=1}^{t-H}S_n(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j)+z_t(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j)\right\}-\min\left\{\sum_{n=0}^{t}d_n,\sum_{n=1}^{t-H}S_n(\boldsymbol{y}+\boldsymbol{e}_i)+z_t(\boldsymbol{y}+\boldsymbol{e}_i)\right\}\\&\quad\leq\min\left\{\sum_{n=0}^{t}d_n,\sum_{n=1}^{t-H}S_n(\boldsymbol{y}+\boldsymbol{e}_j)+z_t(\boldsymbol{y}+\boldsymbol{e}_j)\right\}-\min\left\{\sum_{n=0}^{t}d_n,\sum_{n=1}^{t-H}S_n(\boldsymbol{y})+z_t(\boldsymbol{y})\right\}\end{aligned}\tag{21}$$

Without loss of generality, we assume $i<j$ and analyze the system based on the value of t relative to i and j. Based on changes in the number of on-duty servers, we consider two cases: (1) $t<j$, and (2) $t\geq j$.

When $t<j$, it is trivial to see that this result holds as both sides of the equation are the differences between two equivalent quantities and thus equal zero.

When $t\geq j$, we divide our proof into six subcases based on the relationship between $\sum_{n=0}^{t}d_n$ and $\sum_{n=1}^{t-H}S_n(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j)+z_t(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j)$, $\sum_{n=1}^{t-H}S_n(\boldsymbol{y}+\boldsymbol{e}_i)+z_t(\boldsymbol{y}+\boldsymbol{e}_i)$, $\sum_{n=1}^{t-H}S_n(\boldsymbol{y}+\boldsymbol{e}_j)+z_t(\boldsymbol{y}+\boldsymbol{e}_j)$, and $\sum_{n=1}^{t-H}S_n(\boldsymbol{y})+z_t(\boldsymbol{y})$. In this case, while the first term of the four is the largest and the last term is the smallest, the relationship between $\sum_{n=1}^{t-H}S_n(\boldsymbol{y}+\boldsymbol{e}_i)+z_t(\boldsymbol{y}+\boldsymbol{e}_i)$ and $\sum_{n=1}^{t-H}S_n(\boldsymbol{y}+\boldsymbol{e}_j)+z_t(\boldsymbol{y}+\boldsymbol{e}_j)$

is ambiguous. This ambiguity arises because a schedule with a server starting a later shift may catch up and surpass a schedule with a server starting an earlier shift if sufficient departures occur too late in the time horizon (or during the break) to be fully serviced by the server starting in the time period i.

The number of active servers (servers on duty and not on break) in this time interval varies across different schedules according to the following relationship:

$$\begin{aligned} z_t(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j) - z_t(\boldsymbol{y}+\boldsymbol{e}_i) &= z_t(\boldsymbol{y}+\boldsymbol{e}_j) - z_t(\boldsymbol{y}) \qquad (22) \\ &= \mathbf{1}\{t \in \{j, \ldots, j+L_1-1\} \cup \{j+L_1+B, \ldots, j+L_1+L_2+B-1\}\}. \qquad (23) \end{aligned}$$

In other words, the difference between these terms depends on whether a server that starts at time j is active at time t. Furthermore, the maximum difference among the terms is 1. For ease of exposition, we specify an active period of the housekeeper starting in period j as $A(j)$:

$$A(j) = \{j, \ldots, j+L_1-1\} \cup \{j+L_1+B, \ldots, j+L_1+L_2+B-1\},$$

which simplifies $\mathbf{1}\{t \in \{\{j, \ldots, j+L_1-1\} \cup \{j+L_1+B, \ldots, j+L_1+L_2+B-1\}\}$ to $\mathbf{1}\{t \in A(j)\}$

Case $\boldsymbol{\sum_{n=0}^{t} d_n \le \sum_{n=1}^{t-H} S_n(y) + z_t(y)}$***:***

In this case, Equation (21) becomes

$$\sum_{n=0}^{t} d_n - \sum_{n=0}^{t} d_n \le \sum_{n=0}^{t} d_n - \sum_{n=0}^{t} d_n,$$

which holds as $0 \le 0$.

Case $\boldsymbol{\sum_{n=1}^{t-H} S_n(y+e_i) + z_t(y+e_i) \ge \sum_{n=0}^{t} d_n \ge \sum_{n=1}^{t-H} S_n(y+e_j) + z_t(y+e_j)}$***:***

In this case, Equation (21) becomes

$$0 \le \sum_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_j) + z_t(\boldsymbol{y}+\boldsymbol{e}_j) - \sum_{n=1}^{t-H} S_n(\boldsymbol{y}) - z_t(\boldsymbol{y}),$$

which holds because $\sum_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_j) \ge \sum_{n=1}^{t-H} S_n(\boldsymbol{y})$ and $z_t(\boldsymbol{y}+\boldsymbol{e}_j) \ge z_t(\boldsymbol{y})$ on any sample path.

Case $\boldsymbol{\sum_{n=1}^{t-H} S_n(y+e_j) + z_t(y+e_j) \ge \sum_{n=0}^{t} d_n \ge \sum_{n=1}^{t-H} S_n(y+e_i) + z_t(y+e_i)}$***:***

In this case, Equation (21) becomes

$$\sum_{n=0}^{t} d_n - \sum_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_i) - z_t(\boldsymbol{y}+\boldsymbol{e}_i) \le \sum_{n=0}^{t} d_n - \sum_{n=1}^{t-H} S_n(\boldsymbol{y}) - z_t(\boldsymbol{y}),$$

which holds because $\sum_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_j) \ge \sum_{n=1}^{t-H} S_n(\boldsymbol{y})$ and $z_t(\boldsymbol{y}+\boldsymbol{e}_j) \ge z_t(\boldsymbol{y})$ on any sample path.

Case $\mathbf{\sum_{n=0}^{t} d_n \geq \sum_{n=1}^{t-H} S_n(y) + z_t(y)}$**,** $\mathbf{\sum_{n=0}^{t} d_n \leq \sum_{n=1}^{t-H} S_n(y+e_i) + z_t(y+e_i)}$**,** ***and*** $\mathbf{\sum_{n=0}^{t} d_n \leq \sum_{n=1}^{t-H} S_n(y+e_j) + z_t(y+e_j)}$***:***

In this case, Equation (21) becomes

$$0 \leq \sum_{n=0}^{t} d_n - \sum_{n=1}^{t-H} S_n(\boldsymbol{y}) - z_t(\boldsymbol{y}),$$

which holds by the case assumption.

Case $\mathbf{\sum_{n=0}^{t} d_n \geq \sum_{n=1}^{t-H} S_n(y+e_i+e_j) + z_t(y+e_i+e_j)}$*:* We note that $z_t(\boldsymbol{y}+\boldsymbol{e}_j) - z_t(\boldsymbol{y}) = z_t(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j) - z_t(\boldsymbol{y}+\boldsymbol{e}_i) = \mathbf{1}\{t \in A(j)\}$ on any sample path. In this case, Equation (21) becomes

$$\sum_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j) - \sum_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_i) + \mathbf{1}\{t \in A(j)\} \leq \sum_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_j) - \sum_{n=1}^{t-H} S_n(\boldsymbol{y}) + \mathbf{1}\{t \in A(j)\},$$

which holds by the inductive assumption: $\sum_{n=1}^{t'} S_n(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j) - \sum_{n=1}^{t'} S_n(\boldsymbol{y}+\boldsymbol{e}_i) \leq \sum_{n=1}^{t'} S_n(\boldsymbol{y}+\boldsymbol{e}_j) - \sum_{n=1}^{t'} S_n(\boldsymbol{y})$ holds for all $t' < t$.

Case: $\mathbf{\sum_{n=0}^{t} d_n \leq \sum_{n=1}^{t-H} S_n(y+e_i+e_j) + z_t(y+e_i+e_j)}$**,** $\mathbf{\sum_{n=0}^{t} d_n \geq \sum_{n=1}^{t-H} S_n(y+e_i) + z_t(y+e_i)}$**, .** ***and*** $\mathbf{\sum_{n=0}^{t} d_n \geq \sum_{n=1}^{t-H} S_n(y+e_j) + z_t(y+e_j)}$***:***

In this case Equation (21) becomes

$$\sum_{n=0}^{t} d_n - \sum_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_i) - z_t(\boldsymbol{y}+\boldsymbol{e}_i) \leq \sum_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_j) + z_t(\boldsymbol{y}+\boldsymbol{e}_j) - \sum_{n=1}^{t-H} S_n(\boldsymbol{y}) - z_t(\boldsymbol{y}).$$

Since $\sum_{n=0}^{t} d_n \leq \sum_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j) + z_t(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j)$, we have

$$\begin{aligned}&\sum_{n=0}^{t} d_n - \sum_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_i) - z_t(\boldsymbol{y}+\boldsymbol{e}_i)\\ &\qquad \leq \sum_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j) + z_t(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j) - \sum_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_i) - z_t(\boldsymbol{y}+\boldsymbol{e}_i).\end{aligned}$$

By the induction assumption, we have

$$\sum_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j) - \sum_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_i) \leq \sum_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_j) - \sum_{n=1}^{t-H} S_n(\boldsymbol{y}).$$

Adding $\mathbf{1}\{t \in A(j)\}$ to both sides, we have

$$\sum_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j) - \sum_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_i) + \mathbf{1}\{t \in A(j)\} \leq \sum_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_j) - \sum_{n=1}^{t-H} S_n(\boldsymbol{y}) + \mathbf{1}\{t \in A(j)\},$$

which is equivalent to

$$\sum_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j) + z_t(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j) - \sum_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_i) - z_t(\boldsymbol{y}+\boldsymbol{e}_i) \leq \sum_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_j) + z_t(\boldsymbol{y}+\boldsymbol{e}_j) - \sum_{n=1}^{t-H} S_n(\boldsymbol{y}) - z_t(\boldsymbol{y}).$$

Thus, we have

$$\sum_{n=0}^{t} d_n - \sum_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_i) - z_t(\boldsymbol{y}+\boldsymbol{e}_i) \leq \sum_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_j) + z_t(\boldsymbol{y}+\boldsymbol{e}_j) - \sum_{n=1}^{t-H} S_n(\boldsymbol{y}) - z_t(\boldsymbol{y}),$$

which is the condition that we sought to prove. □

Proof of Proposition 1 For arbitrary schedules $x, y \in \mathbb{Z}_+^V$ such that $x < y$ (coordinate-wise), y can be written in terms of x in following way:

$$y = x + a_1\boldsymbol{e}_1 + a_2\boldsymbol{e}_2 \ldots + a_T\boldsymbol{e}_T = x + \sum_{i=1}^{T} a_i\boldsymbol{e}_i,$$

since schedules are on an integer lattice. Let $x_j^k = x + \sum_{i=1}^{j-1} a_i\boldsymbol{e}_i + k * \boldsymbol{e}_j$, i.e. $x_1^1 = x + e_1, x_1^2 = x + e_1 + e_1, \ldots x_1^{a_1} = x + a_1\boldsymbol{e}_1 \ldots x_T^{a_T} = x + \sum_{i=1}^{T} a_i\boldsymbol{e}_i = y$. By Lemma 1, we have that

$$\sum_{n=1}^{t} S_n(x + \boldsymbol{e}_j) - \sum_{n=1}^{t} S_n(x) \geq \sum_{n=1}^{t} S_n(x + \boldsymbol{e}_i + \boldsymbol{e}_j) - \sum_{n=1}^{t} S_n(x + \boldsymbol{e}_i),$$

for $t \in \{1 \ldots T\}$, $\boldsymbol{y} \in \mathbb{Z}_+^N$, and $i, j \in \beta$. Let $i = 1$, we have

$$\sum_{n=1}^{t} S_n(x + \boldsymbol{e}_j) - \sum_{n=1}^{t} S_n(x) \geq \sum_{n=1}^{t} S_n(x_1^1 + \boldsymbol{e}_j) - \sum_{n=1}^{t} S_n(x_1^1).$$

Using the same logic, we have

$$\sum_{n=1}^{t} S_n(x_1^1 + \boldsymbol{e}_j) - \sum_{n=1}^{t} S_n(x_1^1) \geq \sum_{n=1}^{t} S_n(x_1^2 + \boldsymbol{e}_j) - \sum_{n=1}^{t} S_n(x_1^2).$$

Using an inductive argument, we apply this logic through all schedules between x and y on the lattice to arrive at

$$\sum_{n=1}^{t} S_n(x + \boldsymbol{e}_j) - \sum_{n=1}^{t} S_n(x) \geq \sum_{n=1}^{t} S_n(y + \boldsymbol{e}_j) - \sum_{n=1}^{t} S_n(y),$$

for arbitrary schedules $x, y \in \mathbb{Z}_+^N$ such that $x < y$ (coordinate-wise), for all $j \in \beta, t \in \{1 \ldots T\}$, which is the characterization of the DR-submodularity. □

Proof of Lemma 2 $W_t(\boldsymbol{y}) = I_0^- + \sum_{n=1}^{t} a_n - \min\{\sum_{n=1}^{t} a_n, \sum_{i=1}^{t-H} S_n(y)\}$, i.e., the queue length at the end of each period equals the cumulative number of arrivals plus the initial queue length and minus the cumulative number of service episodes started H or more periods ago. We show that $W_t(\boldsymbol{y})$ is a DR-supermodular function by showing $-W_t(\boldsymbol{y})$ is DR-submodular via showing it fulfills the diminishing margin property described in Lemma 1. Specifically, we want to show:

$$\begin{aligned}\min\left\{\sum_{n=1}^{t} a_n, \sum_{n=1}^{t-H} S_n(\boldsymbol{y} + \boldsymbol{e}_i + \boldsymbol{e}_j)\right\} - \min\left\{\sum_{n=1}^{t} a_n, \sum_{n=1}^{t-H} S_n(\boldsymbol{y} + \boldsymbol{e}_i)\right\} \\ \leq \min\left\{\sum_{n=1}^{t} a_n, \sum_{n=1}^{t-H} S_n(\boldsymbol{y} + \boldsymbol{e}_j)\right\} - \min\left\{\sum_{n=1}^{t} a_n, \sum_{n=1}^{t-H} S_n(\boldsymbol{y})\right\},\end{aligned} \tag{24}$$

We establish the following claim to facilitate our proof of this inequality:

Claim 1. *Let $\alpha, \beta, \gamma, \delta, \psi \in \mathbb{R}$ satisfy $\alpha \geq \beta \geq \delta$ and $\alpha \geq \gamma \geq \delta$, and suppose*

$$\alpha - \beta \leq \gamma - \delta.$$

Then,

$$\min(\psi, \alpha) - \min(\psi, \beta) \leq \min(\psi, \gamma) - \min(\psi, \delta).$$

We distinguish three cases.

Case 1: $\psi \geq \alpha$. Then $\min(\psi,\alpha)=\alpha$, $\min(\psi,\beta)=\beta$, $\min(\psi,\gamma)=\gamma$, and $\min(\psi,\delta)=\delta$, so the inequality reduces to $\alpha-\beta \leq \gamma-\delta$, which holds by assumption.

Case 2: $\psi \leq \delta$. Then, $\min(\psi,\alpha)=\min(\psi,\beta)=\min(\psi,\gamma)=\min(\psi,\delta)=\psi$, so both sides equal zero.

Case 3: $\delta<\psi<\alpha$. Here, $\min(\psi,\alpha)=\psi$. If $\beta \geq \psi$, then $\min(\psi,\beta)=\psi$ and the left-hand side is 0, so the inequality holds. If $\beta<\psi\leq\gamma$, then $\min(\psi,\beta)=\beta$ and $\min(\psi,\gamma)=\psi$. Thus,

$$\min(\psi,\gamma)-\min(\psi,\delta)=\psi-\min(\psi,\delta)\geq\psi-\delta\geq\psi-\beta,$$

since $\beta\geq\delta$. If instead $\gamma<\psi$, then $\min(\psi,\gamma)=\gamma$. Because $\psi-\beta\leq\alpha-\beta\leq\gamma-\delta$, the inequality follows. Thus, in every case, we have

$$\min(\psi,\alpha)-\min(\psi,\beta)\leq\min(\psi,\gamma)-\min(\psi,\delta),$$

which completes the proof. □

Continuing the proof of Lemma 2, we observe that Claim 1 directly applies to equation (24), we let $\alpha=\sum_{n=1}^{t-H}S_n(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j)$, $\beta=\sum_{n=1}^{t-H}S_n(\boldsymbol{y}+\boldsymbol{e}_i)$, $\gamma=\sum_{n=1}^{t-H}S_n(\boldsymbol{y}+\boldsymbol{e}_j)$, $\delta=\sum_{n=1}^{t-H}S_n(\boldsymbol{y})$, and $\psi=\sum_{n=1}^{t}a_n$. We know that

1. $\sum_{n=1}^{t-H}S_n(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j)>\sum_{n=1}^{t-H}S_n(\boldsymbol{y})$
2. $\sum_{n=1}^{t-H}S_n(\boldsymbol{y}+\boldsymbol{e}_i)>\sum_{n=1}^{t-H}S_n(\boldsymbol{y})$
3. $\sum_{n=1}^{t-H}S_n(\boldsymbol{y}+\boldsymbol{e}_j)>\sum_{n=1}^{t-H}S_n(\boldsymbol{y})$
4. $\sum_{n=1}^{t-H}S_n(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j)>\sum_{n=1}^{t-H}S_n(\boldsymbol{y}+\boldsymbol{e}_i)$
5. $\sum_{n=1}^{t-H}S_n(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j)>\sum_{n=1}^{t-H}S_n(\boldsymbol{y}+\boldsymbol{e}_j)$,

and by Lemma 1, we know that $\sum_{n=1}^{t-H}S_n(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j)-\sum_{n=1}^{t-H}S_n(\boldsymbol{y}+\boldsymbol{e}_i)\leq\sum_{n=1}^{t-H}S_n(\boldsymbol{y}+\boldsymbol{e}_j)-\sum_{n=1}^{t-H}S_n(\boldsymbol{y})$, that is, $\alpha-\beta\leq\gamma-\delta$. Applying Claim 1, we know that equation (24) holds.

Therefore, we know that $-W_t(\boldsymbol{y})$ is a DR-submodular function, and $W_t(\boldsymbol{y})$ is a DR-supermodular function. □

Proof of Proposition 2 The proof directly follows the result that DR-submodularity (supermodularity) is closed under addition (Soma and Yoshida 2018). To see this, we can write out the expectation of the total cost as follows:

$$\mathbb{E}\left[C(\boldsymbol{y})\right]=\sum_{\boldsymbol{a}\in\mathbb{Z}_+^T}\sum_{\boldsymbol{d}\in\mathbb{Z}_+^T}C(\boldsymbol{y}|\boldsymbol{A}=\boldsymbol{a},\boldsymbol{D}=\boldsymbol{d})P(\boldsymbol{A}=\boldsymbol{a},\boldsymbol{D}=\boldsymbol{d}),$$

where $C(\boldsymbol{y}|\boldsymbol{A}=\boldsymbol{a},\boldsymbol{D}=\boldsymbol{d})$ is the total cost function when the arrival sample path, $\boldsymbol{A}:=(A_1,A_2,\ldots,A_T)$ takes value $\boldsymbol{a}:=(a_1,a_2,\ldots,a_T)$ and the departure sample path, $\boldsymbol{D}:=(D_1,D_2,\ldots,D_T)$, takes value $\boldsymbol{d}:=(d_1,d_2,\ldots,d_T)$. □

Proof of Lemma 3. We examine the impact of an additional unit of capacity in period i:

$$\begin{aligned} W_t(\boldsymbol{y}) - W_t(\boldsymbol{y}+\boldsymbol{e}_i) &= \sum_{n=1}^{t} a_n - \min\{\sum_{n=1}^{t} a_n, \sum_{n=1}^{t-H} S_n(\boldsymbol{y})\} - \sum_{n=1}^{t} a_n + \min\{\sum_{n=1}^{t} a_n, \sum_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_i)\} \\ &= \min\{\sum_{n=1}^{t} a_n, \sum_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_i)\} - \min\{\sum_{n=1}^{t} a_n, \sum_{n=1}^{t-H} S_n(\boldsymbol{y})\} \end{aligned} \tag{25}$$

Considering the possible orderings of $\sum\limits_{n=1}^{t} a_n$, $\sum\limits_{n=1}^{t-H} S_n(\boldsymbol{y})$, and $\sum\limits_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_i)$,

$$W_t(\boldsymbol{y}) - W_t(\boldsymbol{y}+\boldsymbol{e}_i) = \mathbf{1}\{\sum_{n=1}^{t-H} S_n(\boldsymbol{y}) < \sum_{n=1}^{t} a_n, \sum_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_i) > \sum_{n=1}^{t-H} S_n(\boldsymbol{y})\} \tag{26}$$

The indicator function, $\mathbf{1}\{\sum\limits_{n=1}^{t-H} S_n(\boldsymbol{y}) < \sum\limits_{n=1}^{t} a_n, \sum\limits_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_i) > \sum\limits_{n=1}^{t-H} S_n(\boldsymbol{y})\}$, can be viewed as a product of two indicator functions:

$$\mathbf{1}\{\sum_{n=1}^{t-H} S_n(\boldsymbol{y}) < \sum_{n=1}^{t} a_n\} \times \mathbf{1}\{\sum_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_i) > \sum_{n=1}^{t-H} S_n(\boldsymbol{y})\}$$

For the first indicator function, we observe that

$$\mathbf{1}\{\sum_{n=1}^{t-H} S_n(\boldsymbol{y}) < \sum_{n=1}^{t} a_n\} = \mathbf{1}\{W_t(\boldsymbol{y}) > 0\},$$

because $W_t(\boldsymbol{y}) = \left[\sum\limits_{n=1}^{t} a_n - \sum\limits_{n=1}^{t-H} S_n(\boldsymbol{y})\right]^+$. For the second indicator function, we want to show

$$\mathbf{1}\{\sum_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_i) > \sum_{n=1}^{t-H} S_n(\boldsymbol{y})\} = \prod_{n=i}^{t-H} \mathbf{1}\{V_n(\boldsymbol{y}) > z_n(\boldsymbol{y})\},$$

which is equivalent to showing that $\sum\limits_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_i) > \sum\limits_{n=1}^{t-H} S_n(\boldsymbol{y})$ holds if and only if $V_n(\boldsymbol{y}) > z_n(\boldsymbol{y})$ holds for all $n = i, i+1, ..., t-H$.

We show this equivalency holds by induction on period $n = i, \ldots, t-H$. For $n = i$, $S_i(\boldsymbol{y}+\boldsymbol{e}_i) = S_i(\boldsymbol{y}) + 1$ if and only if $V_i(\boldsymbol{y}) > z_i(\boldsymbol{y})$ by the state equations. Suppose the inductive hypothesis holds through period $\tau - 1$; we must show it holds for period τ, $\tau = i+1, \ldots, t-H$.

In the case that $\sum\limits_{n=1}^{\tau-1} S_n(\boldsymbol{y}+\boldsymbol{e}_i) > \sum\limits_{n=1}^{\tau-1} S_n(\boldsymbol{y})$ and $\prod_{n=i}^{\tau-1} \mathbf{1}\{V_n(\boldsymbol{y}) > z_n(\boldsymbol{y})\} = 1$, then we must consider the following cases:

1. If $V_\tau(\boldsymbol{y}) \le z_\tau(\boldsymbol{y})$, then $\sum\limits_{n=1}^{\tau} S_n(\boldsymbol{y}+\boldsymbol{e}_i) = \sum\limits_{n=1}^{\tau} S_n(\boldsymbol{y})$. To see this, we first observe that $S_\tau(\boldsymbol{y}) = S_\tau(\boldsymbol{y}+\boldsymbol{e}_i) + 1$ because $V_\tau(\boldsymbol{y}+\boldsymbol{e}_i) = V_\tau(\boldsymbol{y}) - 1$ and $\min\{V_\tau(\boldsymbol{y}+\boldsymbol{e}_i), z_\tau(\boldsymbol{y}+\boldsymbol{e}_j)\} = V_\tau(\boldsymbol{y}+\boldsymbol{e}_i)$ as $z_\tau(\boldsymbol{y}+\boldsymbol{e}_j) = z_\tau(\boldsymbol{y}) > V_\tau(\boldsymbol{y}+\boldsymbol{e}_i) = V_\tau(\boldsymbol{y}) - 1$. Given the inductive assumption of $V_i(\boldsymbol{y}) > z_i(\boldsymbol{y})$ and therefore $S_i(\boldsymbol{y}+\boldsymbol{e}_i) = S_i(\boldsymbol{y}) + 1$, we have $\sum\limits_{n=1}^{\tau} S_n(\boldsymbol{y}+\boldsymbol{e}_i) = \sum\limits_{n=1}^{\tau} S_n(\boldsymbol{y})$. Given this relationship and the state equations, the trajectories for the schedules $\boldsymbol{y}$ and $\boldsymbol{y}+\boldsymbol{e}_i$ are the same for periods $\tau + 1, \ldots t-H$. Therefore, we have $\sum\limits_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_i) = \sum\limits_{n=1}^{t-H} S_n(\boldsymbol{y})$, and thus $\mathbf{1}\{\sum\limits_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_i) > \sum\limits_{n=1}^{t-H} S_n(\boldsymbol{y})\} = \prod_{n=i}^{t-H} \mathbf{1}\{V_n(\boldsymbol{y}) > z_n(\boldsymbol{y})\} = 0$.

2. If $V_\tau(\boldsymbol{y}) > z_\tau(\boldsymbol{y})$, then $S_\tau(\boldsymbol{y}) = S_\tau(\boldsymbol{y}+\boldsymbol{e}_i) = z_\tau(\boldsymbol{y}) = z_\tau(\boldsymbol{y}+\boldsymbol{e}_i)$, and we maintain $\sum\limits_{n=1}^{\tau} S_n(\boldsymbol{y}+\boldsymbol{e}_i) = \sum\limits_{n=1}^{\tau} S_n(\boldsymbol{y}) + 1$. Thus, $\mathbf{1}\{\sum\limits_{n=1}^{t-H} S_n(\boldsymbol{y}+\boldsymbol{e}_i) > \sum\limits_{n=1}^{\tau} S_n(\boldsymbol{y})\} = \prod_{n=i}^{\tau} \mathbf{1}\{V_n(\boldsymbol{y}) > z_n(\boldsymbol{y})\} = 1$. This also implies that $V_{\tau+1}(\boldsymbol{y}+\boldsymbol{e}_i) = V_{\tau+1}(\boldsymbol{y}) - 1$.

Otherwise, if $\sum\limits_{n=1}^{\tau-1} S_n(\boldsymbol{y}+\boldsymbol{e}_i) = \sum\limits_{n=1}^{\tau-1} S_n(\boldsymbol{y})$ and $\prod_{n=i}^{\tau-1} \mathbf{1}\{V_n(\boldsymbol{y}) > z_n(\boldsymbol{y})\} = 0$, then $S_\tau(\boldsymbol{y}+\boldsymbol{e}_i) = S_\tau(\boldsymbol{y})$ because $V_\tau(\boldsymbol{y}+\boldsymbol{e}_i) = V_\tau(\boldsymbol{y})$. □

Proof of Lemma 4. Our proof is based on an argument by contradiction and divided into cases by values of i,j,k. Since $(L2[\mathbb{Z}])$ requires $k \notin \{i,j\}$, we divide our discussion into two cases of $i=j$ and $i \neq j$:

***Case:* $\mathbf{i=j}$.** For the first direction, i.e. $(L2[\mathbb{Z}])$ implies one of (12), (13) or (14) holds, we prove by contradiction. Assume $(L2[\mathbb{Z}])$ holds while none of (12), (13) or (14) holds. Under the case assumption, $i=j$, we have $f(x+\boldsymbol{e}_j+\boldsymbol{e}_k) + f(x+\boldsymbol{e}_i) = f(x+\boldsymbol{e}_i+\boldsymbol{e}_k) + f(x+\boldsymbol{e}_j)$. Therefore, (14) reduces to $f(x+\boldsymbol{e}_j+\boldsymbol{e}_k) + f(x+\boldsymbol{e}_i) = f(x+\boldsymbol{e}_i+\boldsymbol{e}_k) + f(x+\boldsymbol{e}_j) \geq f(x+\boldsymbol{e}_i+\boldsymbol{e}_j) + f(x+\boldsymbol{e}_k)$. If (14) is assumed to be not true, we have that $\forall i,j,k \in N$ with $k \notin \{i,j\}$, $f(x+\boldsymbol{e}_j+\boldsymbol{e}_k) + f(x+\boldsymbol{e}_i) = f(x+\boldsymbol{e}_i+\boldsymbol{e}_k) + f(x+\boldsymbol{e}_j) < f(x+\boldsymbol{e}_i+\boldsymbol{e}_j) + f(x+\boldsymbol{e}_k)$, which contradicts the $(L2[\mathbb{Z}])$. For the other direction, we want to show that if one of (12), (13) or (14) holds, then $(L2[\mathbb{Z}])$ holds. Suppose (12) or (13) holds, then $f(x+\boldsymbol{e}_i+\boldsymbol{e}_j) + f(x+\boldsymbol{e}_k) = f(x+\boldsymbol{e}_i+\boldsymbol{e}_k) + f(x+\boldsymbol{e}_j) = f(x+\boldsymbol{e}_j+\boldsymbol{e}_k) + f(x+\boldsymbol{e}_i)$ under the case assumption $i=j$, which directly satisfies $(L2[\mathbb{Z}])$. Supposing Equation (14) holds, we have $f(x+\boldsymbol{e}_i+\boldsymbol{e}_j) + f(x+\boldsymbol{e}_k) \leq f(x+\boldsymbol{e}_i+\boldsymbol{e}_k) + f(x+\boldsymbol{e}_j)$ and $f(x+\boldsymbol{e}_i+\boldsymbol{e}_j) + f(x+\boldsymbol{e}_k) \leq f(x+\boldsymbol{e}_j+\boldsymbol{e}_k) + f(x+\boldsymbol{e}_i)$, which satisfies $(L2[\mathbb{Z}])$ because $f(x+\boldsymbol{e}_i+\boldsymbol{e}_j) + f(x+\boldsymbol{e}_k)$ is the smallest term among three terms.

***Case:* $\mathbf{i \neq j}$.** In this case i,j,k takes three distinct values. For the first direction, i.e. $(L2[\mathbb{Z}])$ implies one of (12), (13) or (14) holds, we again prove by contradiction. Assume $(L2[\mathbb{Z}])$ holds while none of (12), (13) or (14) holds. We have 4 sub-cases:

Case A:* $\mathbf{f(x+e_i+e_j)+f(x+e_k), f(x+e_i+e_k)+f(x+e_j), f(x+e_j+e_k)+f(x+e_i)}$ *take three different values.

Since $(L2[\mathbb{Z}])$ only requires that $\forall i,j,k \in N$ with $k \notin \{i,j\}$, i,j,k indices needs to be fully interchangeable. Specifically, suppose $(L2[\mathbb{Z}])$ and $f(x+\boldsymbol{e}_i+\boldsymbol{e}_j) + f(x+\boldsymbol{e}_k) < f(x+\boldsymbol{e}_i+\boldsymbol{e}_k) + f(x+\boldsymbol{e}_j) < f(x+\boldsymbol{e}_j+\boldsymbol{e}_k) + f(x+\boldsymbol{e}_i)$ both holds. By interchanging the values of i and k, $(L2[\mathbb{Z}])$ and $f(x+\boldsymbol{e}_k+\boldsymbol{e}_j) + f(x+\boldsymbol{e}_i) < f(x+\boldsymbol{e}_k+\boldsymbol{e}_i) + f(x+\boldsymbol{e}_j) < f(x+\boldsymbol{e}_j+\boldsymbol{e}_i) + f(x+\boldsymbol{e}_k)$ should both hold, too. However, this inequality contradicts the $(L2[\mathbb{Z}])$ since $f(x+\boldsymbol{e}_k+\boldsymbol{e}_i) + f(x+\boldsymbol{e}_j) < f(x+\boldsymbol{e}_j+\boldsymbol{e}_i) + f(x+\boldsymbol{e}_k)$ and $f(x+\boldsymbol{e}_k+\boldsymbol{e}_j) + f(x+\boldsymbol{e}_i) < f(x+\boldsymbol{e}_j+\boldsymbol{e}_i) + f(x+\boldsymbol{e}_k)$ while satisfying that $\forall i,j,k \in N$ with $k \notin \{i,j\}$, since after the interchange, i,j,k still takes three different values. The same exchange can be applied when $(L2[\mathbb{Z}])$ and $f(x+\boldsymbol{e}_i+\boldsymbol{e}_k) + f(x+\boldsymbol{e}_j) < f(x+\boldsymbol{e}_i+\boldsymbol{e}_j) + f(x+\boldsymbol{e}_k) < f(x+\boldsymbol{e}_j+\boldsymbol{e}_k) + f(x+\boldsymbol{e}_i)$ both hold. By interchanging the values of k and i, $(L2[\mathbb{Z}])$ and $f(x+\boldsymbol{e}_k+\boldsymbol{e}_i) + f(x+\boldsymbol{e}_j) < f(x+\boldsymbol{e}_k+\boldsymbol{e}_j) + f(x+\boldsymbol{e}_i) < f(x+\boldsymbol{e}_j+\boldsymbol{e}_i) + f(x+\boldsymbol{e}_k)$ should both hold, too. However, this inequality contradicts the ($L2[\mathbb{Z}]$) since $f(x+\boldsymbol{e}_k+\boldsymbol{e}_i)+f(x+\boldsymbol{e}_j) < f(x+\boldsymbol{e}_j+\boldsymbol{e}_i)+f(x+\boldsymbol{e}_k)$ and $f(x+\boldsymbol{e}_k+\boldsymbol{e}_j)+f(x+\boldsymbol{e}_i) < f(x+\boldsymbol{e}_j+\boldsymbol{e}_i)+f(x+\boldsymbol{e}_k)$ while satisfying that $\forall i,j,k \in N$ with $k \notin \{i,j\}$, since after the interchange, i,j,k still takes three different values.

Suppose ($L2[\mathbb{Z}]$) and $f(x+\boldsymbol{e}_j+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_i) < f(x+\boldsymbol{e}_i+\boldsymbol{e}_j)+f(x+\boldsymbol{e}_k) < f(x+\boldsymbol{e}_i+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_j)$ both holds, by interchanging he value of j and k, ($L2[\mathbb{Z}]$) and $f(x+\boldsymbol{e}_j+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_i) < f(x+\boldsymbol{e}_i+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_j) < f(x+\boldsymbol{e}_i+\boldsymbol{e}_j)+f(x+\boldsymbol{e}_k)$ should both hold, too. However, this inequality contradicts the ($L2[\mathbb{Z}]$) since $f(x+\boldsymbol{e}_k+\boldsymbol{e}_i)+f(x+\boldsymbol{e}_j) < f(x+\boldsymbol{e}_j+\boldsymbol{e}_i)+f(x+\boldsymbol{e}_k)$ and $f(x+\boldsymbol{e}_k+\boldsymbol{e}_j)+f(x+\boldsymbol{e}_i) < f(x+\boldsymbol{e}_j+\boldsymbol{e}_i)+f(x+\boldsymbol{e}_k)$ while satisfying that $\forall i,j,k \in N$ with $k \notin \{i,j\}$, since after the interchange, i,j,k still takes three different values. The same exchange can be applied to when ($L2[\mathbb{Z}]$) and $f(x+\boldsymbol{e}_j+\boldsymbol{e}_i)+f(x+\boldsymbol{e}_k) < f(x+\boldsymbol{e}_k+\boldsymbol{e}_j)+f(x+\boldsymbol{e}_i) < f(x+\boldsymbol{e}_i+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_j)$ both hold, by interchanging the value of j and k, ($L2[\mathbb{Z}]$) and $f(x+\boldsymbol{e}_j+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_i) < f(x+\boldsymbol{e}_i+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_j) < f(x+\boldsymbol{e}_i+\boldsymbol{e}_j)+f(x+\boldsymbol{e}_k)$ should both hold, too. However, this inequality contradicts the ($L2[\mathbb{Z}]$) since $f(x+\boldsymbol{e}_k+\boldsymbol{e}_i)+f(x+\boldsymbol{e}_j) < f(x+\boldsymbol{e}_j+\boldsymbol{e}_i)+f(x+\boldsymbol{e}_k)$ and $f(x+\boldsymbol{e}_k+\boldsymbol{e}_j)+f(x+\boldsymbol{e}_i) < f(x+\boldsymbol{e}_j+\boldsymbol{e}_i)+f(x+\boldsymbol{e}_k)$ while satisfying that $\forall i,j,k \in N$ with $k \notin \{i,j\}$, since after the interchange, i,j,k still takes three different values.

Suppose ($L2[\mathbb{Z}]$) holds, then $f(x+\boldsymbol{e}_i+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_j) < f(x+\boldsymbol{e}_j+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_i) < f(x+\boldsymbol{e}_i+\boldsymbol{e}_j)+f(x+\boldsymbol{e}_k)$ cannot hold, because the latter is a direct contradiction with ($L2[\mathbb{Z}]$). Therefore, we do not need to consider this case. The same argument applies to when we suppose ($L2[\mathbb{Z}]$) holds, then $f(x+\boldsymbol{e}_i+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_i) < f(x+\boldsymbol{e}_k+\boldsymbol{e}_i)+f(x+\boldsymbol{e}_j) < f(x+\boldsymbol{e}_i+\boldsymbol{e}_j)+f(x+\boldsymbol{e}_k)$ cannot hold by direct contradiction to ($L2[\mathbb{Z}]$).

Case B: $\mathbf{f(x+e_i+e_j)+f(x+e_k)=f(x+e_i+e_k)+f(x+e_j)<f(x+e_j+e_k)+f(x+e_i)}$**.** Using the similar interchanging arguments, we assume $i'=k$ and $k'=i$; then, we have $f(x+\boldsymbol{e}_{i'}+\boldsymbol{e}_j)+f(x+\boldsymbol{e}_{k'}) > \max\{f(x+\boldsymbol{e}_{k'}+\boldsymbol{e}_{i'})+f(x+\boldsymbol{e}_j), f(x+\boldsymbol{e}_j+\boldsymbol{e}_{i'})+f(x+\boldsymbol{e}_{k'})\}$, where $i',j,k' \in N$ with $k' \notin \{i',j\}$, which contradicts the definition of ($L2[\mathbb{Z}]$).

Case C: $\mathbf{f(x+e_i+e_j)+f(x+e_k)=f(x+e_j+e_k)+f(x+e_i)<f(x+e_i+e_k)+f(x+e_j)}$**.** Using the similar interchanging arguments, we assume $j'=k$ and $k'=j$ then we have $f(x+\boldsymbol{e}_i+\boldsymbol{e}_{k'})+f(x+\boldsymbol{e}_{j'}) > \max\{f(x+\boldsymbol{e}_i+\boldsymbol{e}_{j'})+f(x+\boldsymbol{e}_{k'}), f(x+\boldsymbol{e}'_k+\boldsymbol{e}_{j'})+f(x+\boldsymbol{e}_i)\}$ where $i,j',k' \in N$ with $k' \notin \{i,j'\}$, which contradicts the definition of ($L2[\mathbb{Z}]$).

Case D: $\mathbf{f(x+e_i+e_k)+f(x+e_j)=f(x+e_j+e_k)+f(x+e_i)<f(x+e_i+e_j)+f(x+e_k)}$**.** This case is a direct contradiction of the L2[$\mathbb{Z}$] condition.

For the reverse direction, each of (12), (13), or (14) implies L2[$\mathbb{Z}$], the proof is straightforward and intuitive. Supposing (12) holds, because (12) states that $f(x+\boldsymbol{e}_i+\boldsymbol{e}_j)+f(x+\boldsymbol{e}_k) = f(x+\boldsymbol{e}_i+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_j) \geq f(x+\boldsymbol{e}_j+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_i)$, we have

$$\begin{aligned} & f(x+\boldsymbol{e}_i+\boldsymbol{e}_j)+f(x+\boldsymbol{e}_k) \\ & \qquad = \max\left[f(x+\boldsymbol{e}_i+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_j), f(x+\boldsymbol{e}_j+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_i)\right], \end{aligned}$$

which satisfies L2[ℤ]. Suppose (13) holds, because (13) states that $f(x+\boldsymbol{e}_i+\boldsymbol{e}_j)+f(x+\boldsymbol{e}_k) = f(x+\boldsymbol{e}_j+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_i) \geq f(x+\boldsymbol{e}_i+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_j)$, we have

$$\begin{aligned}&f(x+\boldsymbol{e}_i+\boldsymbol{e}_j)+f(x+\boldsymbol{e}_k)\\&\qquad = \max\left[f(x+\boldsymbol{e}_i+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_j), f(x+\boldsymbol{e}_j+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_i)\right],\end{aligned}$$

which satisfies L2[ℤ]. Suppose (14) holds, because (14) states that $f(x+\boldsymbol{e}_j+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_i) = f(x+\boldsymbol{e}_i+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_j) \geq f(x+\boldsymbol{e}_i+\boldsymbol{e}_j)+f(x+\boldsymbol{e}_k)$, we have

$$\begin{aligned}&f(x+\boldsymbol{e}_i+\boldsymbol{e}_j)+f(x+\boldsymbol{e}_k)\\&\qquad \leq \max\left[f(x+\boldsymbol{e}_i+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_j), f(x+\boldsymbol{e}_j+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_i)\right],\end{aligned}$$

which satisfies L2[ℤ]. □

Proof of Proposition 3. By Theorem 4.5 of Murota and Shioura (2018), $-W_t(\boldsymbol{y})$ is $M^\natural$-concave if and only if it satisfies conditions $(L1[\mathbb{Z}])$ and $(L2[\mathbb{Z}])$. We first observe that $(L1[\mathbb{Z}])$ holds by Lemma 2.

To satisfy the conditions of $(L2[\mathbb{Z}])$ as presented in Lemma 4, we assume without loss of generality that $i \leq j \leq k$. Under that assumption, we prove four relationships to fulfill the conditions of Lemma 4:

$$f(x+\boldsymbol{e}_i+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_j) \geq f(x+\boldsymbol{e}_i+\boldsymbol{e}_j)+f(x+\boldsymbol{e}_k); \tag{27}$$

$$f(x+\boldsymbol{e}_i+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_j) \geq f(x+\boldsymbol{e}_j+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_i); \tag{28}$$

$$\begin{aligned}&f(x+\boldsymbol{e}_i+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_j) > f(x+\boldsymbol{e}_i+\boldsymbol{e}_j)+f(x+\boldsymbol{e}_k) \text{ implies}\\&\qquad f(x+\boldsymbol{e}_i+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_j) = f(x+\boldsymbol{e}_j+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_i);\end{aligned} \tag{29}$$

$$\begin{aligned}&f(x+\boldsymbol{e}_i+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_j) > f(x+\boldsymbol{e}_j+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_i) \text{ implies}\\&\qquad f(x+\boldsymbol{e}_i+\boldsymbol{e}_k)+f(x+\boldsymbol{e}_j) = f(x+\boldsymbol{e}_i+\boldsymbol{e}_j)+f(x+\boldsymbol{e}_k).\end{aligned} \tag{30}$$

Condition (27): To satisfy (27) above with $f(x)$ represented by $-W_t(\boldsymbol{y})$, we show that

$$W_t(\boldsymbol{y}+\boldsymbol{e}_j) - W_t(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j) \leq W_t(\boldsymbol{y}+\boldsymbol{e}_k) - W_t(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_k) \tag{31}$$

for $t = 1,\ldots,T$. In other words, it is less valuable (in terms of reducing waiting) to delay employing the extra unit of capacity from period j to k when there is an extra unit of capacity in period i. Using Lemma 3, this condition reduces to

$$\begin{aligned}&\mathbf{1}\{W_t(\boldsymbol{y}+\boldsymbol{e}_k)>0\}\prod_{n=i}^{t-H}\mathbf{1}\{V_n(\boldsymbol{y}+\boldsymbol{e}_k)>z_n(\boldsymbol{y}+\boldsymbol{e}_k)\}\\ \geq&\mathbf{1}\{W_t(\boldsymbol{y}+\boldsymbol{e}_j)>0\}\prod_{n=i}^{t-H}\mathbf{1}\{V_n(\boldsymbol{y}+\boldsymbol{e}_j)>z_n(\boldsymbol{y}+\boldsymbol{e}_j)\}.\end{aligned} \tag{32}$$

Now, we show (32) in two subcases based on the time period t:

Case $\boldsymbol{t < k + H}$***:*** In this case, the extra capacity starting in period k is not active yet. In accordance with this intuition, (32) holds because the events in the indicator functions on the right-hand side (RHS) imply the events in indicator functions on the left-hand side (LHS) as $V_n(\boldsymbol{y}+\boldsymbol{e}_k) \geq V_n(\boldsymbol{y}+\boldsymbol{e}_j)$ and $z_n(\boldsymbol{y}+\boldsymbol{e}_j) \geq z_n(\boldsymbol{y}+\boldsymbol{e}_k)$ when $t < k+h$ and $n = i, i+1 \ldots, k$, and $W_t(\boldsymbol{y}+\boldsymbol{e}_k) \geq W_t(\boldsymbol{y}+\boldsymbol{e}_j)$ for $t < k+H$.

Case $\boldsymbol{t \geq k + H}$ ***:*** If $\prod_{n=i}^{k} \mathbf{1}\{V_n(\boldsymbol{y}+\boldsymbol{e}_j) > z_n(\boldsymbol{y}+\boldsymbol{e}_j)\} = 0$, then (32) holds trivially. If $\prod_{n=i}^{k} \mathbf{1}\{V_n(\boldsymbol{y}+\boldsymbol{e}_j) > z_n(\boldsymbol{y}+\boldsymbol{e}_j)\} = 1$, then extra capacity added in period k must have started a job; i.e., $V_k(\boldsymbol{y}+\boldsymbol{e}_k) > z_k(\boldsymbol{y}+\boldsymbol{e}_k)$. Therefore, $V_t(\boldsymbol{y}+\boldsymbol{e}_k) = V_t(\boldsymbol{y}+\boldsymbol{e}_j)$ and $z_t(\boldsymbol{y}+\boldsymbol{e}_j) = z_t(\boldsymbol{y}+\boldsymbol{e}_k)$ for $t > k$. Furthermore, if $\prod_{n=i}^{t-H} \mathbf{1}\{V_n(\boldsymbol{y}+\boldsymbol{e}_j) > z_n(\boldsymbol{y}+\boldsymbol{e}_j)\} = 1$ for $t \geq k+H$, we must have that $W_t(\boldsymbol{y}+\boldsymbol{e}_k) = W_t(\boldsymbol{y}+\boldsymbol{e}_j)$ as the extra capacity added in period k must have resulted in an extra cleaned resource available to allocate in period $k+H$. Thus, the two schedules will have turned around the same number of resources when $t \geq k+H$, and the condition holds.

Condition (28): To satisfy (28) above, we show that

$$W_t(\boldsymbol{y}+\boldsymbol{e}_j) - W_t(\boldsymbol{y}+\boldsymbol{e}_j+\boldsymbol{e}_k) \leq W_t(\boldsymbol{y}+\boldsymbol{e}_i) - W_t(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_k) \tag{33}$$

for $t = 1, \ldots, T$. In other words, the extra unit of capacity in period k is more valuable (in terms of reducing waiting) if the other extra unit of capacity is employed in period i instead of period j. Applying Lemma 3, the condition reduces to

$$\begin{aligned} &\mathbf{1}\{W_t(\boldsymbol{y}+\boldsymbol{e}_i) > 0\} \prod_{n=k}^{t-H} \mathbf{1}\{V_n(\boldsymbol{y}+\boldsymbol{e}_i) > z_n(\boldsymbol{y}+\boldsymbol{e}_i)\} \\ \geq &\mathbf{1}\{W_t(\boldsymbol{y}+\boldsymbol{e}_j) > 0\} \prod_{n=k}^{t-H} \mathbf{1}\{V_n(\boldsymbol{y}+\boldsymbol{e}_j) > z_n(\boldsymbol{y}+\boldsymbol{e}_j)\}. \end{aligned} \tag{34}$$

To show that (34) holds, we divide our discussion into two cases based on t:

Case $\boldsymbol{t < k + H}$***:*** In this case, the condition trivially holds with equality. Intuitively, it is because the extra capacity added in period k has not turned around any resources yet. Therefore, (33) reduces to

$$W_t(\boldsymbol{y}+\boldsymbol{e}_j) - W_t(\boldsymbol{y}+\boldsymbol{e}_i) \leq W_t(\boldsymbol{y}+\boldsymbol{e}_j) - W_t(\boldsymbol{y}+\boldsymbol{e}_i), \tag{35}$$

which holds with equality.

Case $\boldsymbol{t \geq k + H}$***:*** In this case, we first note that $W_t(\boldsymbol{y}+\boldsymbol{e}_j) \leq W_t(\boldsymbol{y}+\boldsymbol{e}_i)$ because schedule $\boldsymbol{y}+\boldsymbol{e}_j$ has potentially started one more job than schedule $\boldsymbol{y}+\boldsymbol{e}_i$ because $j \geq i$. Therefore, in Equation (34), the first indicator function on the RHS implies the first indicator function on the LHS. For the second indicators on both sides, we note that $z_n(\boldsymbol{y}+\boldsymbol{e}_i) = z_n(\boldsymbol{y}+\boldsymbol{e}_j)$ and $V_n(\boldsymbol{y}+\boldsymbol{e}_i) \geq V_n(\boldsymbol{y}+\boldsymbol{e}_j)$ when $t > k$; therefore, (34) holds.

Condition (29)**:** Now, we show that (29) is satisfied, as specified by the following:

$$W_t(x+\boldsymbol{e}_i+\boldsymbol{e}_k)+W_t(x+\boldsymbol{e}_j)<W_t(x+\boldsymbol{e}_i+\boldsymbol{e}_j)+W_t(x+\boldsymbol{e}_k) \text{ implies}$$
$$W_t(x+\boldsymbol{e}_i+\boldsymbol{e}_k)+W_t(x+\boldsymbol{e}_j)=W_t(x+\boldsymbol{e}_j+\boldsymbol{e}_k)+W_t(x+\boldsymbol{e}_i);$$

We want to show that the inequality part in (29), which can be expressed using indicator functions as

$$\begin{aligned}&\mathbf{1}\{W_t(\boldsymbol{y}+\boldsymbol{e}_k)>0\}\prod_{n=i}^{t-H}\mathbf{1}\{V_n(\boldsymbol{y}+\boldsymbol{e}_k)>z_n(\boldsymbol{y}+\boldsymbol{e}_k)\}\\&\quad>\mathbf{1}\{W_t(\boldsymbol{y}+\boldsymbol{e}_j)>0\}\prod_{n=i}^{t-H}\mathbf{1}\{V_n(\boldsymbol{y}+\boldsymbol{e}_j)>z_n(\boldsymbol{y}+\boldsymbol{e}_j)\},\end{aligned}\tag{36}$$

implies the equality part of (29), which is equivalent to:

$$\begin{aligned}&\mathbf{1}\{W_t(\boldsymbol{y}+\boldsymbol{e}_i)>0\}\prod_{n=k}^{t-H}\mathbf{1}\{V_n(\boldsymbol{y}+\boldsymbol{e}_i)>z_n(\boldsymbol{y}+\boldsymbol{e}_i)\}\\&=\mathbf{1}\{W_t(\boldsymbol{y}+\boldsymbol{e}_j)>0\}\prod_{n=k}^{t-H}\mathbf{1}\{V_n(\boldsymbol{y}+\boldsymbol{e}_j)>z_n(\boldsymbol{y}+\boldsymbol{e}_j)\}\end{aligned}\tag{37}$$

These expressions consider the scenario in which the extra unit of capacity is utilized if added in period k but wasted if added in period j; we must show it is also wasted if added in period i instead of j. Again, we divide our discussion into two cases based on the value of t

Case $\boldsymbol{t<k+H}$***:*** Similar to the previous case, the equality holds trivially as the extra server started in period k has not returned any resources yet; i.e. $W_t(\boldsymbol{y}+\boldsymbol{e}_j+\boldsymbol{e}_k)=W_t(\boldsymbol{y}+\boldsymbol{e}_j)$ and $W_t(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_k)=W_t(\boldsymbol{y}+\boldsymbol{e}_i)$. Condition (29) becomes

$$W_t(x+\boldsymbol{e}_i+\boldsymbol{e}_k)+W_t(x+\boldsymbol{e}_j)<W_t(x+\boldsymbol{e}_i+\boldsymbol{e}_j)+W_t(x+\boldsymbol{e}_k) \text{ implies}$$
$$W_t(x+\boldsymbol{e}_i)+W_t(x+\boldsymbol{e}_j)=W_t(x+\boldsymbol{e}_j)+W_t(x+\boldsymbol{e}_i);$$

which holds true because of the equality of the implied condition.

Case $\boldsymbol{t\geq k+H}$***:*** In this case, suppose the inequality in (29) holds strictly. Then, by equation (36), we have $\mathbf{1}\{W_t(\boldsymbol{y}+\boldsymbol{e}_k)>0\}\prod_{n=i}^{t-H}\mathbf{1}\{V_n(\boldsymbol{y}+\boldsymbol{e}_k)>z_n(\boldsymbol{y}+\boldsymbol{e}_k)\}>\mathbf{1}\{W_t(\boldsymbol{y}+\boldsymbol{e}_j)>0\}\prod_{n=i}^{t-H}\mathbf{1}\{V_n(\boldsymbol{y}+\boldsymbol{e}_j)>z_k(\boldsymbol{y}+\boldsymbol{e}_j)\}$. Thus, $\prod_{n=i}^{t-H}\mathbf{1}\{V_n(\boldsymbol{y}+\boldsymbol{e}_k)>z_k(\boldsymbol{y}+\boldsymbol{e}_k)\}=1$. That is, for schedule $\boldsymbol{y}+\boldsymbol{e}_k$, we have $V_n(\boldsymbol{y}+\boldsymbol{e}_k)>z_n(\boldsymbol{y}+\boldsymbol{e}_k)$ for any $n=i,i+1,\ldots,k\}$. Since $z_n(\boldsymbol{y}+\boldsymbol{e}_k)$ and $z_n(\boldsymbol{y}+\boldsymbol{e}_j)$ or $z_n(\boldsymbol{y}+\boldsymbol{e}_i)$ differ by at most 1 by Lemma 3, we have $V_n(\boldsymbol{y}+\boldsymbol{e}_j)\geq z_n(\boldsymbol{y}+\boldsymbol{e}_j)$ and $V_n(\boldsymbol{y}+\boldsymbol{e}_i)\geq z_n(\boldsymbol{y}+\boldsymbol{e}_i)$ for all $n=i,\ldots,t-H$. Therefore, we know that both the schedule $\boldsymbol{y}+\boldsymbol{e}_i$ and the schedule $\boldsymbol{y}+\boldsymbol{e}_j$ are fully utilized from period i to period $t-H$, which implies that $\sum_{n=1}^{t}S_n(\boldsymbol{y}+\boldsymbol{e}_i)=\sum_{n=1}^{t}S_n(\boldsymbol{y}+\boldsymbol{e}_j)$ for $t\geq j$. Furthermore, schedule $\boldsymbol{y}+\boldsymbol{e}_i$ and schedule $\boldsymbol{y}+\boldsymbol{e}_j$ have the same trajectories after period j as they started the same amount of work by period j and have the same capacity after period j. Therefore, we have $V_n(\boldsymbol{y}+\boldsymbol{e}_i)=V_n(\boldsymbol{y}+\boldsymbol{e}_j)$ and $z_n(\boldsymbol{y}+\boldsymbol{e}_i)=z_n(\boldsymbol{y}+\boldsymbol{e}_j)$ for $n=k,k+1\ldots t-H$, and $W_t(\boldsymbol{y}+\boldsymbol{e}_i)=W_t(\boldsymbol{y}+\boldsymbol{e}_j)$ for $t\geq k+H$. Thus, these relationships imply that equation (37) holds with equality, which implies that (29) holds.

Condition (30): Finally, we show that Condition (30) is satisfied, as specified by the following:

$W_t(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_k)+W_t(\boldsymbol{y}+\boldsymbol{e}_j)<W_t(\boldsymbol{y}+\boldsymbol{e}_j+\boldsymbol{e}_k)+W_t(\boldsymbol{y}+\boldsymbol{e}_i)$ implies

$$W_t(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_k)+W_t(\boldsymbol{y}+\boldsymbol{e}_j)=W_t(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j)+W_t(\boldsymbol{y}+\boldsymbol{e}_k).$$

We want to show that the inequality part in conditions (30), which can be expressed using indicator functions as

$$\begin{aligned}&\mathbf{1}\{W_t(\boldsymbol{y}+\boldsymbol{e}_i)>0\}\prod_{n=k}^{t-H}\mathbf{1}\{V_n(\boldsymbol{y}+\boldsymbol{e}_i)>z_n(\boldsymbol{y}+\boldsymbol{e}_i)\}\\ &>\mathbf{1}\{W_t(\boldsymbol{y}+\boldsymbol{e}_j)>0\}\prod_{n=k}^{t-H}\mathbf{1}\{V_n(\boldsymbol{y}+\boldsymbol{e}_j)>z_n(\boldsymbol{y}+\boldsymbol{e}_j)\},\end{aligned}\tag{38}$$

implies the equality part in conditions (30), which is equivalent to

$$\begin{aligned}&\mathbf{1}\{W_t(\boldsymbol{y}+\boldsymbol{e}_k)>0\}\prod_{n=i}^{t-H}\mathbf{1}\{V_n(\boldsymbol{y}+\boldsymbol{e}_k)>z_n(\boldsymbol{y}+\boldsymbol{e}_k)\}\\ &=\mathbf{1}\{W_t(\boldsymbol{y}+\boldsymbol{e}_j)>0\}\prod_{n=i}^{t-H}\mathbf{1}\{V_n(\boldsymbol{y}+\boldsymbol{e}_j)>z_k(\boldsymbol{y}+\boldsymbol{e}_j)\}\end{aligned}\tag{39}$$

In words, (38) supposes a scenario in which an extra unit of capacity in period i is wasted but an extra unit of capacity in period j is utilized. In this case, we must show that an extra unit of capacity in period j has the same effect as an extra unit of capacity in period k.

Case $\boldsymbol{t<k+H}$: In this case, the inequality part in (30) is impossible as the extra server started in period k has not yet had the opportunity to service any resources. Therefore, for $t<k+H$, we have $W_t(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_k)=W_t(\boldsymbol{y}+\boldsymbol{e}_i)$ and $W_t(\boldsymbol{y}+\boldsymbol{e}_j+\boldsymbol{e}_k)=W_t(\boldsymbol{y}+\boldsymbol{e}_j)$. Thus the inequality part in (30) reduces to $W_t(\boldsymbol{y}+\boldsymbol{e}_i)+W_t(\boldsymbol{y}+\boldsymbol{e}_j)>W_t(\boldsymbol{y}+\boldsymbol{e}_j)+W_t(\boldsymbol{y}+\boldsymbol{e}_i)$, which is impossible. Therefore, we disregard this case.

Case $\boldsymbol{t\geq k+H}$: In this case, suppose the inequality in (30) above holds strictly: i.e., $\mathbf{1}\{W_t(\boldsymbol{y}+\boldsymbol{e}_i)>0\}\prod_{n=k}^{t-H}\mathbf{1}\{V_n(\boldsymbol{y}+\boldsymbol{e}_i)>z_n(\boldsymbol{y}+\boldsymbol{e}_i)\}>\mathbf{1}\{W_t(\boldsymbol{y}+\boldsymbol{e}_j)>0\}\prod_{n=k}^{t-H}\mathbf{1}\{V_n(\boldsymbol{y}+\boldsymbol{e}_j)>z_n(\boldsymbol{y}+\boldsymbol{e}_j)\}$. We consider two subcases: (1) $\mathbf{1}\{W_t(\boldsymbol{y}+\boldsymbol{e}_j)>0\}=0$; and (2) $\mathbf{1}\{W_t(\boldsymbol{y}+\boldsymbol{e}_j)>0\}=1$. For the first subcase, $\mathbf{1}\{W_t(\boldsymbol{y}+\boldsymbol{e}_j)>0\}=0$ implies $\mathbf{1}\{W_t(\boldsymbol{y}+\boldsymbol{e}_k)>0\}=0$ as the number of resources turned around by schedule $\boldsymbol{y}+\boldsymbol{e}_k$ is at least as many as the resources of schedule $\boldsymbol{y}+\boldsymbol{e}_j$ for $t\geq k+H$. Therefore, the equality holds as both sides are 0. For the second subcase, where $\mathbf{1}\{W_t(\boldsymbol{y}+\boldsymbol{e}_j)>0\}=1$, for the inequality to hold, $\prod_{n=k}^{t-H}\mathbf{1}\{V_n(\boldsymbol{y}+\boldsymbol{e}_i)>z_n(\boldsymbol{y}+\boldsymbol{e}_i)\}$ must take value 1 and $\prod_{n=k}^{t-H}\mathbf{1}\{V_n(\boldsymbol{y}+\boldsymbol{e}_j)>z_n(\boldsymbol{y}+\boldsymbol{e}_j)\}$ must take value 0. These two terms differ from each other, implying that the extra capacity added in period i is wasted, whereas the extra capacity added in period j is utilized. In other words, these two terms are equal to each other if and only if the extra capacity added in i and j is both utilized or wasted. Therefore, given that the extra capacity added in period i is wasted, the equality part in (30):

$$W_t(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_k)+W_t(\boldsymbol{y}+\boldsymbol{e}_j)=W_t(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j)+W_t(\boldsymbol{y}+\boldsymbol{e}_k)\tag{40}$$

reduces to

$$W_t(\boldsymbol{y}+\boldsymbol{e}_k)+W_t(\boldsymbol{y}+\boldsymbol{e}_j)=W_t(\boldsymbol{y}+\boldsymbol{e}_j)+W_t(\boldsymbol{y}+\boldsymbol{e}_k), \tag{41}$$

which holds with equality. □

Proof of Proposition 4. As in the previous proposition, we need to show that $(L1[\mathbb{Z}])$ and $(L2[\mathbb{Z}])$ continue to hold for $-C(\boldsymbol{y})$. By (7) we can re-write $(L1[\mathbb{Z}])$ as

$$C(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j)+C(\boldsymbol{y})\geq C(\boldsymbol{y}+\boldsymbol{e}_i)+C(\boldsymbol{y}+\boldsymbol{e}_j).$$

Because the staffing costs on each side of the inequality sum to $\left(2\sum_{t=1}^{T-1}k_t y_t\right)+k_i+k_j$, this condition reduces to

$$\sum_{t=1}^{T}b_tW_t(\boldsymbol{y}+\boldsymbol{e}_i+\boldsymbol{e}_j)+\sum_{t=1}^{T}b_tW_t(\boldsymbol{y})\geq\sum_{t=1}^{T}b_tW_t(\boldsymbol{y}+\boldsymbol{e}_i)+\sum_{t=1}^{T}b_tW_t(\boldsymbol{y}+\boldsymbol{e}_j),$$

which holds due to Proposition 3 with $b_t\geq 0$.

The same reduction to inequalities shown to be true in Proposition 3 occurs for the $(L2[\mathbb{Z}])$ conditions, except that the left-hand side and right-hand side of each inequality and equation have a total staffing cost of $\left(2\sum_{t=1}^{T-1}k_t y_t\right)+k_i+k_j+k_k$.

If $C(\boldsymbol{y})$ is M$^\natural$-convex for any realized sample path, then $\mathbb{E}\left[C(\boldsymbol{y})\right]$ is M$^\natural$-convex over multiple sample paths as the $(L1[\mathbb{Z}])$ and $(L2[\mathbb{Z}])$ conditions continue to hold summed over multiple sample paths. Thus, the expected total cost $\mathbb{E}\left[C(\boldsymbol{y})\right]$ is M-convex in $\boldsymbol{y}$. □

Proof of Proposition 5. We provide a counterexample with $L=3$ with $\boldsymbol{y}=(1,1,1)$, and $(i,j,k)=(1,2,3)$. The sample paths for the six configurations of $\boldsymbol{y}$ appearing in $(L2[\mathbb{Z}])$ are displayed in Table 8. For $t=3$ in this example, we observe

$$\begin{aligned}&\sum_{n=1}^{3}S_n((1,2,2))+\sum_{n=1}^{3}S_n((2,1,1))=18\\>&\sum_{n=1}^{3}S_n((2,1,2))+\sum_{n=1}^{3}S_n((1,2,1))=17\\>&\sum_{n=1}^{3}S_n((2,2,1))+\sum_{n=1}^{3}S_n((1,1,2))=16.\end{aligned}$$

If the number of arrivals by period 3 exceeds 18, $\sum_{t=1}^{3}a_t\geq 18$, and $I_0=0$, then $(L2[\mathbb{Z}])$ is violated for $-W_t(\boldsymbol{y})$. □

Proof of Lemma 5. The proof is by induction. In period $t=1$, (16) becomes

$$S_1(y)=\min\left\{d_0+d_1,y\right\},$$

Table 8 **Counterexample sample path with $L = 3$.**

Period	t	**1**	**2**	**3**		**1**	**2**	**3**
Departures	d_t	9	0	0		9	0	0
Shift Starts	y_t	2	1	2		1	2	1
Active	$z_t(\boldsymbol{y})$	2	3	5		1	3	4
Vacant Dirty	$V_t(\boldsymbol{y})$	9	7	4		9	8	5
Starts	$S_t(\boldsymbol{y})$	2	3	4		1	3	4
Total Starts	$\sum_{n=1}^{t} S_t(\boldsymbol{y})$	2	5	9		1	4	8
Shift Starts	y_t	2	2	1		1	1	2
Active	$z_t(\boldsymbol{y})$	2	4	5		1	2	4
Vacant Dirty	$V_t(\boldsymbol{y})$	9	7	3		9	8	6
Starts	$S_t(\boldsymbol{y})$	2	4	3		1	2	4
Total Starts	$\sum_{n=1}^{t} S_t(\boldsymbol{y})$	2	6	9		1	3	7
Shift Starts	y_t	1	2	2		2	1	1
Active	$z_t(\boldsymbol{y})$	1	3	5		2	3	4
Vacant Dirty	$V_t(\boldsymbol{y})$	9	8	5		9	7	4
Starts	$S_t(\boldsymbol{y})$	1	3	5		2	3	4
Total Starts	$\sum_{n=1}^{t} S_t(\boldsymbol{y})$	1	4	9		2	5	9

which is concave and non-decreasing in y. Assuming the inductive hypothesis, $\sum_{n=1}^{t'} S_n(y)$ is discretely concave and non-decreasing in y, for $t' < t$, we must prove $\sum_{n=1}^{t} S_n(y)$ is also discretely concave and non-decreasing in y. We begin by rewriting the cumulative sum using (16) as

$$\sum_{n=1}^{t} S_n(y) = \sum_{n=1}^{t-1} S_n(y) + S_t(y) = \sum_{n=1}^{t-1} S_n(y) + \min\left\{V_t(y), y - \sum_{n=t-H+1}^{t-1} S_n(y)\right\},$$

which can be rewritten using (1) as

$$\sum_{n=1}^{t} S_n(y) = \sum_{n=1}^{t-1} S_n(y) + \min\left\{\sum_{n=0}^{t} d_n - \sum_{n=1}^{t-1} S_n(y), y - \sum_{n=t-H+1}^{t-1} S_n(y)\right\},$$

which is equivalent to

$$\sum_{n=1}^{t} S_n(y) = \min\left\{\sum_{n=0}^{t} d_n, y + \sum_{n=1}^{t-H} S_n(y)\right\}.$$

Because $\sum_{n=1}^{t-H} S_n(y)$ is concave in y by the inductive assumption, $y + \sum_{n=1}^{t-H} S_n(y)$ is also concave in y. By the properties of concave functions and minimum operators, $\min\left\{\sum_{n=1}^{t} d_n, y + \sum_{n=1}^{t-H} S_n(y)\right\}$ is also concave in y.

We use a similar approach to show that $\sum_{n=1}^{t} S_n(y+1) \geq \sum_{n=1}^{t} S_n(y)$, which implies $\sum_{n=1}^{t} S_n(\cdot)$ is non-decreasing in y. Substituting $S_n(y+1)$ and $S_n(y)$ as above using (16) and (1), the inequality becomes

$$\begin{aligned}&\sum_{n=1}^{t-1} S_n(y+1) + \min\left\{\sum_{n=0}^{t} d_n - \sum_{n=1}^{t-1} S_n(y+1), y+1 - \sum_{n=t-H+1}^{t-1} S_n(y+1)\right\}\\ \geq &\sum_{n=1}^{t-1} S_n(y) + \min\left\{\sum_{n=0}^{t} d_n - \sum_{n=1}^{t-1} S_n(y), y - \sum_{n=t-H+1}^{t-1} S_n(y)\right\},\end{aligned}$$

Table 9 Example system with one arrival in period 1, one departure in period 1, one server working only in period 1, another server working only in period 6, and a cleaning duration of $H = 2$ periods ($d_0 = a_0 = I_0 = V_0 = 0$).

Time Period	t	1	2	3	4	5	6	7	8
Departures	d_t	1	0	0	0	0	0	0	0
Arrivals	a_t	1	0	0	0	0	0	0	0
Active Servers	z_t	1	0	0	0	0	1	0	0
Available Servers	R_t	1	-1	1	-1	1	0	0	0
Vacant Dirty Resources	V_t	1	0	1	0	1	0	0	0
Resources for which Service Starts	S_t	1	-1	1	-1	1	0	0	0
Inventory Position	I_t	-1	-1	0	-1	0	-1	0	0
Waiting Customers	W_t	1	1	0	1	0	1	0	0

which is equivalent to

$$\min\left\{\sum_{n=0}^{t} d_n, y+1+\sum_{n=1}^{t-H} S_n(y+1)\right\} \geq \min\left\{\sum_{n=0}^{t} d_n, y+\sum_{n=1}^{t-H} S_n(y)\right\}. \tag{42}$$

Next, we observe that $y+1+\sum_{n=1}^{t-H} S_n(y+1) \geq y+\sum_{n=1}^{t-H} S_n(y)$ because $y+1 > y$ and $\sum_{n=1}^{t-H} S_n(y+1) \geq \sum_{n=1}^{t-H} S_n(y)$ by the inductive hypothesis. Furthermore, we observe that if $y+1+\sum_{n=1}^{t-H} S_n(y+1) \geq y+\sum_{n=1}^{t-H} S_n(y)$, then we have $\min\left\{\sum_{n=0}^{t} d_n, y+1+\sum_{n=1}^{t-H} S_n(y+1)\right\} \geq \min\left\{\sum_{n=0}^{t} d_n, y+\sum_{n=1}^{t-H} S_n(y)\right\}$. Thus, $\sum_{n=1}^{t} S_n(y)$ is non-decreasing in y. □

Proof of Proposition 6. Using (3), we can express the number of customers waiting at the end of period t as

$$\begin{aligned} W_t(y) &= \left[\sum_{n=1}^{t-H} S_n(y) - \sum_{n=1}^{t} a_n\right]^- \\ &= \max\left\{0, \sum_{n=1}^{t} a_n - \sum_{n=1}^{t-H} S_n(y)\right\}. \end{aligned}$$

By Lemma 5, we know that $-\sum_{n=1}^{t-H} S_n(y)$ is concave and non-decreasing in y. Thus, $W_t(y)$ is convex and non-increasing in y because of the properties for a convex and non-increasing function with the addition of $\sum_{n=0}^{t} a_n$ and the maximum expression resulting and 0. □

Proof of Proposition 7. By Proposition 6, $\left(\sum_{t=1}^{T} b_t W_t(y)\right)$ is convex and non-increasing for any $d_1, \ldots, d_T$ and $a_1, \ldots, a_T$. Also, ky is convex in y. As the sum of two convex functions, (17) is also convex in y. Because this holds for any realization of departures and arrivals, it also holds in expectation.

□

2. Break and End-of-Shift Service Assumption

To avoid complicating our state equations to handle a server's break or shift end when no other servers are available, we assume (5) is the number of available servers in any period. Table 9 shows how this assumption works for a simple system with one arrival and one departure and a resource with a turnaround time of $H = 2$ periods. One server provides one of the two required time units of service to the resource in period 1 and is no longer available. Another server is available in period 6 and completes the service so that the resource can be allocated at the beginning of period 7. Our assumption means that the number of available servers oscillates between -1 and 1 from periods 2 to 5, and the number of waiting customers oscillates between 1 and 0. Thus, a waiting cost is assessed for half of the periods during which the resource is waiting for service to resume.

We offer an alternate model that more explicitly models server availability during breaks and at the end of a shift in Section 3 of the online supplement. While we do not have analytical results for this model, our numerical testing leads us to believe that the alternate model has the same properties as the model presented in this section. Also, only negligible differences exist for the two models in their estimation of policy performance.

This approach represents a compromise in which the key performance metrics are bounded by two policies: servers do not begin servicing a resource if they cannot complete it before a break or shift completion, and servers delay their break to complete servicing the resource or stay overtime and complete their work. In effect, any rooms not completed by the break are fractionally allocated to a guest until sufficient capacity is available again.

3. Alternate Model for End-of-Shift and Break Policies

We describe the state equations for an alternate model of room cleaning that avoids the assumption stated in (5). In particular, this model explicitly tracks the number of time periods of cleaning required for each vacant dirty resource and does not allow a resource to be allocated until it is fully cleaned. We assume that resources with the fewest number of periods of cleaning remaining preempt cleaning of any other resources; e.g., if the server cleaning a resource with only 1 period of cleaning remaining goes on break, a server working on a resource with many more periods remaining will switch to that resource. While this may not exactly reflect common practice — i.e., a hotel room attendant might just complete a room before going on break — this assumption allows for relatively simple state equations that avoid dynamic decision making about break timing. If the server were to wait to take a break until the resource is completely serviced, the model's estimate of resource availability would be slightly conservative.

Let $V_{tk}(\boldsymbol{y})$ represent the number of vacant dirty resources in period $t=0,\ldots,T$ with $k=1,\ldots,H$ periods of cleaning remaining until it can be allocated to a new customer. Also, we use $S_{tk}(\boldsymbol{y})$ to denote the number of resources with k servicing periods remaining that are serviced in period t. The $z_t(\boldsymbol{y})$ active servers may each perform one period of servicing; i.e., $\min\left\{z_t(\boldsymbol{y}), \sum_{k=1}^{H} V_{tk}(\boldsymbol{y})\right\}$ resources receive servicing so that their state advances from k to $k-1$ in period t. Resource service priority increases as k decreases so that, in each period $t=1,\ldots,T$ it can be defined recursively as

$$S_{t,k}(\boldsymbol{y}) := \min\left\{V_{t,k}(\boldsymbol{y}), z_t(\boldsymbol{y}) - \sum_{m=1}^{k-1} S_{t,m}(\boldsymbol{y})\right\} \tag{43}$$

for $k = 1, \ldots, H$. Furthermore, we also have the following recursive state equations for each period t:

$$V_{t,H}(\boldsymbol{y}) := V_{t-1,H}(\boldsymbol{y}) + D_t - S_{t-1,H}(\boldsymbol{y}), \tag{44}$$

$$V_{t,k}(\boldsymbol{y}) := V_{t-1,k}(\boldsymbol{y}) + S_{t-1,k+1}(\boldsymbol{y}) - S_{t-1,k}(\boldsymbol{y}), \qquad k = 1, \ldots, H-1, \tag{45}$$

$$I_t(\boldsymbol{y}) := I_{t-1}(\boldsymbol{y}) - A_t + S_{t-1,1}(\boldsymbol{y}). \tag{46}$$

All other system definitions from Section 3 continue to apply.

4. Algorithms

Algorithm 1 finds a solution that performs at most $\frac{1}{2}$ worse than the optimal solution. Algorithm 1 is based on the first algorithm in Soma and Yoshida (2017) but has two differences: first, we aim to minimize a DR-supermodular function instead of maximizing a DR-submodular function; second, the cardinality constraint $B \in \mathbb{Z}_+^N : \boldsymbol{y} < B$ need not to be binding in our case. To achieve this, one can set the cardinality constraint B to be a vector of large values, e.g. the total number of hotel rooms. For problems that have a specific cardinality constraint, one can set B accordingly and still obtain the guarantee of the 1/2 optimality rate. Let B' denote a sufficiently large vector of capacity constraint and let $g(\chi_e \mid x)$ denote $g(x + \chi_e) - g(x)$, we modify the algorithm in the following manner:

Algorithm 1: Pseudopolynomial-time algorithm for minimizing a DR-supermodular function

```
Input: DR-supermodular function $f : \mathbb{Z}_+^N \to \mathbb{R}$, large capacity upper bound $B' \in \mathbb{Z}_+^N$ or actual
       capacity upper bound $B \in \mathbb{Z}_+^N$
Output: Approximate minimizer $x \in \mathbb{Z}_+^N$
Define $g(x) = -f(x)$;
Initialize $x \leftarrow 0$, $y \leftarrow B'$;
foreach $t \in \beta$ do
    while $x_t < y_t$ do
        $\alpha \leftarrow g(\boldsymbol{e}_t \mid x)$;
        $\gamma \leftarrow g(-\boldsymbol{e}_t \mid y)$;
        if $\gamma < 0$ then
            $x_t \leftarrow x_t + 1$;
        else
            if $\alpha < 0$ then
                $y_t \leftarrow y_t - 1$;
            else
                With probability $\frac{\alpha}{\alpha+\gamma}$, set $x_t \leftarrow x_t + 1$;
                Otherwise, set $y_t \leftarrow y_t - 1$;
            end
        end
    end
end
return $x$;
```

Algorithm 2 implements the Steepest Descent Algorithm in Shioura (2022) to find the minimizer of an M-convex function. However, rather than starting from the vector $\mathbf{0}$, we specify an input $\boldsymbol{y}_0$ as the initial search point. Note that this algorithm searches within a constrained hyperplane; therefore, it requires sufficiently large capacity $Y \in \mathbb{Z}_+$ as an input. Let $||\cdot||_1$ denote the $L1-$norm of a vector. For any initial input $\boldsymbol{y}^0$, we add $Y - ||\boldsymbol{y}^0||_1$ as the slack element for $\boldsymbol{y}^0$ to ensure that the search is within a constrained hyperplane. The value of Y — as long as it is sufficiently large — does not affect the search itself, as the value for the slack element, $Y - ||\boldsymbol{y}^0||_1$, does not enter the objective function.

Algorithm 2: Steepest Descent for an M-Convex Function

```
Input: $f : \mathbb{Z}_+^N \to \mathbb{R}_+$, $\boldsymbol{y}^0 \in \mathbb{Z}_+^N$, sufficiently large number $Y \in \mathbb{Z}_+$
Output: $\boldsymbol{y} \in \mathbb{Z}_+^N$
$\tilde{\boldsymbol{y}} \leftarrow (\boldsymbol{y}^0, Y - ||\boldsymbol{y}_0||_1)$;
Find the $i, j$ pair that maximizes $f(\tilde{\boldsymbol{y}}) - f(\tilde{\boldsymbol{y}} + \boldsymbol{e}_i - \boldsymbol{e}_j)$;
while $f(\tilde{\boldsymbol{y}}) - f(\tilde{\boldsymbol{y}} + \boldsymbol{e}_i - \boldsymbol{e}_j) > 0$ do
    Find the $i, j$ pair that maximizes $f(\tilde{\boldsymbol{y}}) - f(\tilde{\boldsymbol{y}} + \boldsymbol{e}_i - \boldsymbol{e}_j)$;
    $\tilde{\boldsymbol{y}} \leftarrow \tilde{\boldsymbol{y}} + \boldsymbol{e}_i - \boldsymbol{e}_j$;
end
return $\boldsymbol{y}$ ($\tilde{\boldsymbol{y}}$ without the slack element);
```

5. Additional Results for Workers with Short Shifts

In our conversations with hotel workforce strategists, they identified workers interested in shifts lasting less than 8 hours as a labor pool that hotels might need to utilize. For example, they envision high school or college students as labor sources for working shorter shifts at hotels in urban markets. Alternatively, part-time workers in this model could represent cross-trained workers who clean rooms for half of their shift and fulfill some other role for the remainder of their shift. In theory, using part-time workers should be advantageous for hotels due to the ability to schedule capacity when it is most needed. For our comparison, part-time room attendants have a shift cost and timing structure that maintains the same cost per room cleaned as full-time workers when fully utilized: $k_p = \$120$ for a four-hour shift, which includes 12.5 minutes of start-up time before cleaning the first room and no breaks during the four-hour shift. Thus, two part-time room attendants have the same pay and cleaning capacity as one full-time room attendant.

Table 10 **Optimal schedules with part-time room attendants working 4-hour shifts and 1,300 rooms to clean.**

		Shift Start Time											Total
		8 AM	9 AM	10 AM	11 AM	12 PM	1 PM	2 PM	3 PM	4 PM	5 PM	6 PM	
Current Practice		0	90	0	0	0	0	0	0	10	0	0	100
Optimal Full-time		34	23	22	3	5	5	5	0	0	1	1	100
Optimal with Part-time	Part-Time	5	74	9	22	19	0	1	2	51	2	15	200
	Full-Time	0	0	0	0	0	0	0	0	0	0	0	0

Table 10 shows the optimal schedule with the option to choose full-time or part-time room attendants and 1,300 rooms to clean. Highlighting the value of flexible capacity that can be employed with more precision, the optimal solution chooses 200 part-time room attendants and 0 full-time

Table 11 Optimal schedules with a maximum of 50 part-time room attendants available to work 4-hour shifts.

Number of Rooms to Clean		Schedule											Total
		8 AM	9 AM	10 AM	11 AM	12 AM	1 PM	2 PM	3 PM	4 PM	5 PM	6 PM	
1600	Part-Time	9	0	0	0	0	0	0	0	21	3	17	50
	Full-Time	15	26	42	8	6	4	0	0	0	0	0	101
1900	Part-Time	3	0	0	0	0	0	0	7	18	1	21	50
	Full-Time	17	44	42	13	9	0	0	0	0	0	0	125

room attendants. This result is expected as two part-time workers have the same cleaning capacity and cost as one full-time worker but allow for added flexibility. Because it is possible to achieve a solution with negligible waiting using only full-time workers, the benefits of part-time workers are not immediately obvious for this specific scenario. However, under a more realistic scenario with a maximum of 50 part-time workers, part-time workers can sometimes help the hotel reducing the number of full-time equivalent workers through better targeting capacity allocation. In particular, Table 11 shows a decrease in the required staffing level for the scenarios with a high volume of rooms to clean. For the scenarios with 1,600 and 1,900 rooms to clean, using part-time workers can reduce the staffing level needed by the equivalent of 3 and 6 full-time workers, respectively. In both cases, the waiting costs remain less than $5. Part-time workers appear to be particularly valuable in the late afternoon and evening. This analysis shows the usefulness of part-time workers or alternate sources of shorter-term capacity in helping a hotel to clean all rooms in time to avoid customer waiting.

6. Sample Average Approximation Accuracy Test

We numerically determine the size of the sample paths required to solve our problem using sample average approximation. method numerically. For different sizes of solving sample path, we generate 100 sets of solving and testing sample paths. Then, we find the optimal solution for the solving sample paths and evaluate the resulting solution on the testing sample paths. Finally, we solve the testing sample path optimally to check if it corresponds to the solution given by the solving sample path. Table 12 confirms that 100 solving sample paths and 100 testing sample paths are sufficient to solve the problem: the in-sample solution is 100% optimal when evaluated on the testing sample paths. Also, the in-sample average total cost is nearly identical to that of the testing sample paths. Therefore, we set the solving and testing sample-paths size to be 100 for our analysis.

Table 12 Solution accuracy based on the number of sample paths (solving and testing) and 100 replications.

Evaluation Design	Performance Measure		
Sample Paths	MSE	MAPE	Optimal Rate
50	\$0.79	0.0006%	98%
100	\$0.26	0.0003%	100%
200	\$0.25	0.0003%	100%