Design of Patient Visit Itineraries in Tandem Systems

Nan Liu
Carroll School of Management, Boston College, Chestnut Hill, MA 02467, USA, nan.liu@bc.edu

Guohua Wan
Antai College of Economics and Management, Shanghai Jiao Tong University, Shanghai 200030, China, ghwan@sjtu.edu.cn

Shan Wang
School of Business, Sun Yat-sen University, Guangzhou, 510275, China, wangsh337@mail.sysu.edu.cn

Problem definition: Multi-stage service is common in healthcare. One widely adopted approach to manage patient visits in multi-stage service is to provide patients with visit itineraries, which specify personalized appointment time for each patient at each service stage. We study how to design such visit itineraries. **Methodology/results:** We develop the first optimization modeling framework to provide each patient a personalized visit itinerary in a tandem (healthcare) service system. Due to interdependencies among stages, our model loses those elegant properties (e.g., L-convexity and submodularity) often utilized to solve the classic single-stage models. To address these challenges, we develop two original reformulations. One is directly amenable to off-the-shelf optimization software and the other is a concave minimization problem over a polyhedron shown to have neat structural properties, based on which we develop efficient solution algorithms. In addition to these exact solution approaches, we propose an approximation approach with a provable optimality bound and numerically validated performance to serve as an easy-to-implement heuristic. A case study populated by data from the Dana-Farber Cancer Institute shows that our approach makes a remarkable 28% cost reduction over practice on average. **Managerial implications:** Common approaches used in practice are based on simple adjustments to schedules generated by single-stage models, often assuming deterministic service times. Whereas such approaches are intuitive and take advantage of existing knowledge on single-stage models, they can lead to significant loss of operational efficiency in managing multi-stage services. A well-designed patient visit itinerary which carefully addresses the interdependencies among stages can significantly improve patient experience and provider utilization.

Key words: healthcare operations management, patient scheduling, tandem systems, optimization

1. Introduction

Appointment scheduling is commonly used in service industries to match provider capacity with customer demand. How to schedule appointments for customers is a classic problem in service operations management (OM). Extensive research efforts have been devoted in this area since the seminal work by Bailey (1952). However, most, if not all, existing studies only focus on a single-stage system, where customers are assumed to receive service from one stage and then leave. Whereas a single-stage system is representative for some settings with one main service stage, it does not fully capture system dynamics and managerial challenges in more complex service environments.

One industry that often sees multi-stage services is healthcare. (In the rest of the article, we use "patients" and "customers" interchangeably.) For instance, in an infusion center, patients usually go through three stages of services—blood draw, physician exam (to verify the treatment plan), and finally, infusion treatment. In an orthopedic clinic, a common care path involves an X-ray exam, followed by physician consultation, and finally, service from orthopedic technicians. A common feature in these settings is that patients receive sequential services and each service stage requires a substantial amount of time with significant variability. As a result, patients who receive service in such tandem systems often spend a long time, if not a full day, in the facility. As healthcare is becoming more patient-centered, it is crucial for healthcare providers to better manage patient care experience, especially in clinical environments with lengthy visit durations. Patients might have to take a whole day off work and have considerable anxieties about their treatments. The ramifications of prolonged, unproductive, and uncomfortable patient stays within healthcare facilities extend beyond mere irritation. Healthcare providers can suffer significantly from loss of goodwill among patients, who may eventually choose to seek care elsewhere due to negative perceptions.

To manage such multi-stage care processes, many providers tell patients the appointment time at the first service stage, and then ask them to check in at the next stage and wait until they can be seen. However, this is clearly not desirable from the patients' perspectives because they are not well informed about what to expect during their visits. A more appealing approach to manage patient experience in such complex environments is to provide patients with visit *itineraries*, which specify appointment time for *each* service stage. This approach is adopted by several large healthcare organizations that we know. For instance, a large healthcare organization in Florida prepares visit itineraries to specify appointment times for patients to receive sequential mammography tests, e.g., the first digital tomosynthesis test scheduled at 8:30am followed by the second bilateral whole-breast ultrasonography test at 8:45am. To design patient visit itineraries, the-state-of-the-art practice leverages the understanding of well-studied single-stage appointment scheduling systems. Practitioners of a large infusion center in the Greater Boston area inform us that a common practice is to first determine patient inter-appointment intervals based on one particular stage of service, usually the one considered as "bottleneck" (e.g., physician exam). Then, the appointment times for preceding/following stages are provided by subtracting/adding expected service time in those stages, respectively (more on this in Section 6.3).

Under this scheduling paradigm, one can leverage classic single-stage models to easily derive inter-appointment intervals for the bottleneck stage (while ignoring other stages), and then deduce the appointment times in other stages. Whereas this is an easy-to-implement heuristic idea taking advantage of existing knowledge, it only uses expected service times to capture the interdependencies of patient services across multiple stages, and may lead to significant loss of operational efficiencies. Our research is motivated by the design issues of patient visit itinerary in such tandem service systems. Most extant literature develops models and insights that can only be applied to single-stage services; managing multi-stage services requires fundamentally different tools and guidelines. In particular, we seek to understand how to schedule appointments in tandem service systems with the objective to achieve a fine balance between patient waiting and provider utilization.

We model daily operations of the tandem service process described above as a transient tandem queueing system. Customers are scheduled to arrive at different stages of service, and their scheduled arrival times are the decision variables of our model. Service stations are sequentially connected. (In the rest of the article, we use "stations" and "stages" interchangeably.) We assume that one service provider works at each station in our theoretical analysis, based on which we also develop heuristic ideas to deal with settings with multiple providers staffed in a single station. Customers visit these stations consecutively before leaving. Similar to a single-stage system, a customer's service starting time at one station cannot be earlier than his appointment time for service at this stage, nor the service finishing time of his immediate preceding counterpart at this stage. In addition, his service starting time can neither be earlier than his own service finishing time in the prior stage—this is due to the nature of sequential services and actually makes our model fundamentally different from the classic single-stage model. Due to such interdependencies across stages, the evaluation and properties of customer waiting time and provider idle time also deviate significantly from those in single-stage systems.

Indeed, one interesting and distinct feature of our model is how we capture customer waiting and provider idling in a multi-stage system. In particular, we consider two types of customer waiting and two types of provider idling. The first type of waiting is called *regular* waiting, which occurs when a patient has finished service in the prior stage and for the current stage his appointment time has already come, but the provider of the current stage is still working on other patients. The second type of waiting is what we call *leisure* waiting, which takes place when a customer finishes his service in the prior stage earlier than his appointment time for the current stage. We assume that leisure waiting costs less than regular waiting per unit of time, because leisure waiting is fully anticipated by patients and patients do not have to stand by but may spend that time in some leisure activities (e.g., taking a walk) to relieve some of the discomfort due to waiting. For provider idling, the first type is called *recoverable* idling which occurs when the provider finishes the service of her current patient but her next patient's appointment time has not come yet, so she could use the time for some ancillary activities, e.g., making phone calls (Chen and Robinson 2014). The second type of provider idling is called *regular* idling, which takes place when the provider is ready, the patient's appointment time has arrived, but the patient is not present. We assume that recoverable idling costs no greater than regular idling per unit of time, because during regular idling the provider has to stay put, waiting for patients who may arrive anytime.

By specifying patient appointment times at each service stage, a carefully designed visit itinerary replaces patient regular waiting by less costly leisure waiting and provider regular idling by less costly recoverable idling, thereby improving overall utilities. The objective of our model is to identify such a visit itinerary which minimizes the total expected patient waiting costs (both types), provider idling costs (both types) and overtime costs. In addition to accounting for dynamics in a multi-stage system, our model is also able to handle general forms of well-known uncertainties that affect appointment scheduling decisions, such as patient-specific non-punctuality, stage-dependent random service times, heterogeneous patient no-show behavior and non-identical service sequences.

Our contributions can be summarized as follows. To the best of our knowledge, we develop the first optimization modeling framework to create personalized visit itineraries for customers in a tandem service system. We disprove those elegant properties of the objective function (e.g., L-convexity and submodularity), which usually hold in classic single-stage appointment scheduling problems. We find that our problem does not yield an equivalent convex relaxation as observed in single-stage models. To address these challenges, we propose two original reformulations: a mixed binary linear stochastic program and a concave minimization problem over a polyhedron. Leveraging these, we develop efficient solution approaches. In addition, we present an approximation solution approach with a provable optimality bound and numerically validated performance. This approach offers a practical heuristic scheduling policy, optimizing appointment times for all patients at the first stage and the first patient in subsequent stages. In a case study using data from an infusion center, we address scenarios with multiple providers in one stage by approximating a multi-server queue as multiple single-server queues. This multi-server heuristic makes a remarkable 28% cost reduction over current practices on average. Overall, our research underscores the importance of considering interdependencies among different stages in tandem service systems.

The remainder of this paper is organized as follows. Section 2 briefly reviews the relevant literature. Section 3 develops and analyzes our scheduling model for tandem systems. In Sections 4 and 5, we present our exact and approximation solution approaches, respectively. Section 6 conducts numerical studies, and Section 7 discusses several extensions of our model. Finally, Section 8 makes concluding remarks. The proofs of all analytical results are available in the Online Appendix.

2. Literature Review

Our work is closely related to the appointment scheduling literature that investigates how to schedule patients over time in a day. This literature focuses on developing math programming models to optimize the tradeoff between patient waiting and provider utilization (Gupta and Denton 2008, Ahmadi-Javid et al. 2017). Previous studies have considered a variety of uncertainties in practice that can affect the design of appointment scheduling systems (such as patient no-shows, random service times, and walk-in behavior). Representative work includes, but is not limited to, Jiang et al. (2017), Wang et al. (2020), Kong et al. (2020) and Zacharias and Yunes (2020).

We draw attention to a few articles that are most related to our work from the modeling perspective. All these studies, like ours, treat patient appointment times as decision variables. (The other modeling approach is to determine the number of patients to schedule in each discretized time slot.) Denton and Gupta (2003) formulate the appointment scheduling problem as a two-stage stochastic linear program and then exploit the problem structure to derive upper bounds and a variation of the standard L-shaped algorithm to obtain optimal solutions. Begen and Queyranne (2011) consider a similar problem where each job has a random processing duration given by a joint discrete probability distribution. They show that the objective function is submodular and L-convex. Hassin and Mendel (2008) study how to schedule appointments with patient no-shows and the impact of no-shows on the optimal schedule. Similar to most literature on appointment scheduling, all these three articles focus on single-stage systems. Our work fundamentally departs from them in that we consider multi-stage services and investigate the appointment time for each patient at each stage. We demonstrate that the resulting optimization problem has a completely different structure, and to tackle it we develop new solution approaches.

A rising stream of literature has started to investigate scheduling decisions in multi-stage services. Kuiper and Mandjes (2015), Alvarez-Oh et al. (2018) and Soltani et al. (2019) study how to set appointment times for patients at the first stage. Wang et al. (2018, 2019) propose dynamical policies to accept patients to the appointment book at the beginning stage in a network of services. All these studies focus on managing patient schedule at the *entry* stage. Another modeling approach taken by prior work follows the traditional job-scheduling framework, focusing on throughput maximization and/or makespan minimization via dynamic or static controls to regulate patient flows through a service network (Diamant et al. 2018, Zhang et al. 2019). But these models do not consider patient waiting or provider idling, factors that are essential in setting up daily appointment times for patients. Our work complements and advances this literature by developing an original modeling framework to optimize personalized visit itinerary (i.e., appointment times at all stages for each patient) in a multi-stage service system, taking into account a full set of decision impact factors including various cost components and uncertainty sources.

From a broader perspective, our work is related to the healthcare OM literature on patient flow management among different hospital units (e.g., surgical suites, inpatient wards, ICUs). In general, this body of work studies the control of patient admission, either via cyclic admission schedules or dynamic policies, to optimize strategic-level metrics such as throughput and resource utilization; see, e.g., Helm and Van Oyen (2014), Liu et al. (2019), Dai and Shi (2019) for some examples. These studies consider multi-day operations in inpatient care, whereas our model focuses on intra-day operations for outpatient care. Our work is also connected to the literature on admission control for tandem queues. See Zhang and Ayhan (2012) for a brief review. These admission control studies are typically concerned with whether to accept or reject an arriving customer, taking into account its impact on downstream services. These two streams of work tend to have different application contexts compared to ours. Therefore, though the service system we consider shares a similar tandem structure as theirs, the managerial levers to use and performance metrics to optimize are fundamentally different.

3. The Model

In this section, we develop an appointment scheduling model for tandem service systems. Consider a clinical practice with $N \geq 2$ sequentially connected service stations numbered $1, 2, \ldots, N$. Each station is staffed with a single provider, and we call the provider working at station n as provider n. During a day, $K \geq 1$ patients, indexed by $1, 2, \ldots, K$, need to visit these stations sequentially. For now, we assume that all patients will go to the next station $n+1$ after finishing the service at station n. (In Section 7, we will show that our model can incorporate settings where patients do not visit all stations and may leave the clinic at some station.) The time of a day is divided into discrete time slots, say 5 minutes per slot. The service duration of patient k at station n is a random variable δ_k^n for which the support is the set of nonnegative integers. Let $\boldsymbol{\delta}$ denote $(\boldsymbol{\delta}_1, \boldsymbol{\delta}_2, \ldots, \boldsymbol{\delta}_k, \ldots, \boldsymbol{\delta}_K)$, where $\boldsymbol{\delta}_k = (\delta_k^1, \delta_k^2, \ldots, \delta_k^N)$. We allow δ_k^n to be heterogeneous and correlated, i.e., $\boldsymbol{\delta}$ can follow any nonnegative discrete distribution.

Our decision variables are the appointment itineraries for all patients, denoted by $\boldsymbol{a}_k = (a_k^1, a_k^2, \ldots, a_k^N)$, where a_k^n is the scheduled arrival time of patient k at station n. We require that a_k^n to be an integer, meaning that no patients are scheduled in the middle of a time slot. This requirement conforms to scheduling practice which usually provides patients with appointment times at the beginning of a slot, rather than some arbitrary time say 8:17am. Let $\boldsymbol{a}$ denote the vector $(\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_K)$, which contains all decision variables. Without loss of generality, we set $a_1^1 = 0$, which can be viewed as the clinic open time. For now, we assume that all scheduled patients are punctual in arrival to the clinic (and we will relax this assumption later in Section 7). Patient service sequence is predetermined and remains unchanged at all stations. We thus require that $a_k^n \geq a_{k-1}^n$, $\forall k \geq 2, n \geq 1$. For regularity, we further require that for the same patient, his appointment time at an earlier stage must precede that at a later one, i.e., $a_k^n \geq a_k^{n-1}$, $\forall n \geq 2, k \geq 1$.

Let T^n be the length of regular working hours at station n. Denote $(T^1, T^2, \ldots, T^N)$ by $\boldsymbol{T}$. We require that all appointments for station n must be scheduled before T^n. If a provider cannot finish her work during regular working hours, patients will be served during overtime. In this section, we assume that all scheduled patients will show up for their appointments at each station. In Section 7, we will show that our model can be easily extended to incorporate patient no-show behavior.

A common optimization framework in the literature is to assign different cost rates to patient wait time and provider idle time and overtime—and—then to minimize the expected total weighted cost. We follow this framework, but note that this cost structure needs to be generalized in our study context because, as discussed below, we distinguish two different types of waiting for patients and two different types of idling for providers.

Let r_k^n denote the ready time of patient k at station n. A patient is ready for service at station n when his service at station $n-1$ is completed and his appointment time at station n has been reached. Similarly, we let v_k^n denote the available time of provider n to serve patient k. This happens when she finishes serving patient $k-1$ at station n and patient k's appointment time at station n has been reached. Finally, we let s_k^n denote the actual service start time of patient k at station n. A patient starts his service when he is ready and the provider is available. Then we have the following recursive equations for $\boldsymbol{r}$, $\boldsymbol{v}$, and $\boldsymbol{s}$, respectively.

$$r_k^n = \begin{cases} a_k^1 & \text{if } n = 1, \\ \max\{a_k^n, s_k^{n-1} + \delta_k^{n-1}\} & \text{if } n \geq 2, \end{cases} \tag{1}$$

$$v_k^n = \begin{cases} a_1^n & \text{if } k = 1, \\ \max\{a_k^n, s_{k-1}^n + \delta_{k-1}^n\} & \text{if } k \geq 2, \end{cases} \tag{2}$$

and

$$s_k^n = \max\{r_k^n, v_k^n\}. \tag{3}$$

The first type of waiting for patient k at station n, called *leisure* waiting, takes place from his service finish time at the prior station $n-1$, i.e., $s_k^{n-1} + \delta_k^{n-1}$, to his ready time r_k^n for station n. Note that a patient may not be ready even if his service from the prior station is done because his appointment time for the next station may have not arrived yet; see (1). The second type of waiting for patient k at station n, called *regular* waiting, is the time gap from his ready time r_k^n to the actual service start time s_k^n. During regular waiting, patients must be standing by and waiting in the facility because his next service may start at any moment. However, during leisure waiting, patients know that their services will not start until their appointment times. By definition, leisure waiting is known, finite and explained waiting, whereas regular waiting is uncertain and likely unexplained. As a result, patients are informed and less anxious during leisure waiting (Larson 1987, Maister et al. 1984). In addition to these psychological reliefs, patients can also be more physically comfortable during leisure waiting because they anticipate they need to wait, and hence they do not have to spend the time only in the waiting area but may choose to stretch their legs and relax somewhere more comfortable. Therefore, it is natural to expect that leisure waiting costs less than regular waiting. Now, let $\mathit{\Gamma}_{W,k}$ and $\mathit{\Gamma}_{L,k}$ denote the total regular wait time and leisure wait time of patient k, respectively. We have

$$\mathit{\Gamma}_{W,k} = \sum_{n=1}^{N} (s_k^n - r_k^n) \tag{4}$$

and

$$\mathit{\Gamma}_{L,k} = \sum_{n=2}^{N} (r_k^n - s_k^{n-1} - \delta_k^{n-1}). \tag{5}$$

Next, we discuss the two types of idling experienced by providers. A provider will be idle when she finishes the service of one patient and the next patient is not ready yet. The first type of idling that provider n experiences while waiting for patient k is the time gap from her service finish time of the previous patient, i.e., $s_{k-1}^n + \delta_{k-1}^n$, to her available time v_k^n for patient k; we term such idling as *recoverable* idling. Note that a provider may not be available to serve the next patient even if she is done with the current one because the next patient's appointment time may have not arrived yet; see (2). The second type of idling experienced by provider n when waiting to serve patient k starts from her available time v_k^n to the actual service start time s_k^n; this is called *regular* idling. During regular idling, providers must get ready for their patients and are unlikely to do anything too productive because patients may arrive anytime, However, providers are likely to be more productive during recoverable idling time by completing ancillary activities (Chen and Robinson 2014), such as renewing prescriptions, returning emails, and making phone calls, because patients are not expected to arrive yet. Let $\mathit{\Gamma}_I^n$ and $\mathit{\Gamma}_R^n$ denote the total regular idle time and total recoverable idle time of provider n, respectively. Then we have

$$\mathit{\Gamma}_I^n = \sum_{k=1}^{K} (s_k^n - v_k^n) \tag{6}$$

and

$$\mathit{\Gamma}_R^n = \sum_{k=2}^{K} (v_k^n - s_{k-1}^n - \delta_{k-1}^n). \tag{7}$$

The overtime at each station is evaluated in a conventional way. We first define the actual "off-duty" time of provider n as follows:

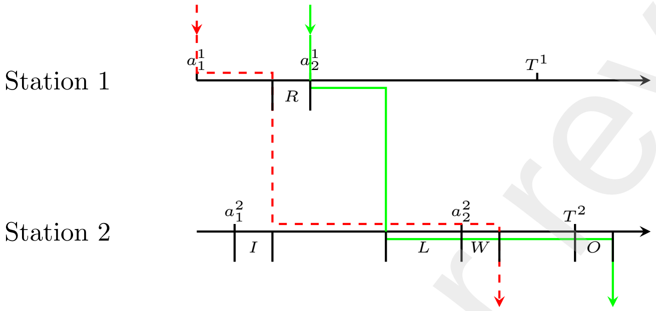
$$f^n = \max\{s_K^n + \delta_K^n, T^n\}. \tag{8}$$

That is, provider n needs to work until either T^n, her regular working hours end, or the last patient K completes his service, whichever occurs later. It then follows that the overtime of provider n is the difference between her actual off-duty time and her regular off-duty time, i.e.,

$$\mathit{\Gamma}_O^n = f^n - T^n - a_1^1 = f^n - T^n. \tag{9}$$

Note that these ready times, available times, start times and actual off-times depend on both of the random variables $\boldsymbol{\delta}$ and decision variables $\boldsymbol{a}$.

Before discussing the cost parameters and problem formulation, we use a two-station two-patient example in Figure 1 to illustrate the aforementioned patient flows and various cost components in a tandem service system. The red dashed arrow shows the details of patient 1's visit, including his arrival times, times to get service, and service completion times at both stations. The green solid arrow shows similar information for patient 2. We see that provider 1 only experiences recoverable idling when waiting to serve patient 2, whereas provider 2 encounters regular idling before serving patient 1 and endures overtime work in order to serve patient 2. Patient 1 has no waiting, but patient 2 enjoys leisure waiting at station 2 before his appointment time at that stage and then experiences regular waiting after his appointment time until his service at station 2 starts.



Red dashed arrow: flow of patient 1; Green solid arrow: flow of patient 2.

R: recoverable idle time; I: regular idle time; L: leisure wait time; W: regular wait time.; O: overtime.

Figure 1 Sample Patient Flows in a Tandem Service System with 2 Stations

Next, we introduce the cost parameters. Let $C_{W,k}$ and $C_{L,k}$ be the unit regular waiting cost and unit leisure waiting cost for patient k, respectively. Let C_I^n and C_R^n be the unit regular idling cost and unit recoverable idling cost for provider n, respectively. It is natural to impose that $0 \le C_{L,k} \le C_{W,k}$ and $0 \le C_R^n \le C_I^n$ because, as discussed above, patients have a more pleasant experience during leisure waiting compared to regular waiting and providers incur lower idling costs during recoverable idling time than regular idling time. Finally, let $C_O^n \ge 0$ be the unit overtime cost of provider n. Then we can formulate our optimization problem as follows.

$$\begin{aligned}
\min \quad & \mathbb{E}\Big[\sum_{k=1}^{K}\big(C_{W,k}\mathit{\Gamma}_{W,k}(\boldsymbol{a})+C_{L,k}\mathit{\Gamma}_{L,k}(\boldsymbol{a})\big)+\sum_{n=1}^{N}\big(C_I^n\mathit{\Gamma}_I^n(\boldsymbol{a})+C_R^n\mathit{\Gamma}_R^n(\boldsymbol{a})+C_O^n\mathit{\Gamma}_O^n(\boldsymbol{a})\big)\Big] && (\mathbf{P})\\
\text{s.t.} \quad & \mathit{\Gamma}_{W,k}(\boldsymbol{a}), \mathit{\Gamma}_{L,k}(\boldsymbol{a}), \mathit{\Gamma}_I^n(\boldsymbol{a}), \mathit{\Gamma}_R^n(\boldsymbol{a}), \mathit{\Gamma}_O^n(\boldsymbol{a}) \text{ are defined in } (4),(5),(6),(7),(9), \text{ respectively,}\\
& a_k^n \ge a_{k-1}^n, \ \forall\, k \ge 2, \ n \ge 1,\\
& a_k^n \ge a_k^{n-1}, \ \forall\, k \ge 1, \ n \ge 2,\\
& a_1^1 = 0,\\
& a_k^n \in \{0,1,2,\dots,T^n\}, \ \forall\, k,n. && (10)
\end{aligned}$$

Remark 1. *In* $(\mathbf{P})$*, the integer constraint* (10) *can be relaxed as* $0 \le a_k^n \le T^n, \ \forall\, k,n$.

Remark 2. *This optimization model admits a fairly general cost structure. Specifically, it allows waiting cost to be patient-dependent and idling and overtime costs to be provider-dependent.*

Before proceeding to the analysis of the model, we briefly discuss how one may estimate these cost parameters in our model. One classic way to evaluate the value of time is to use its opportunity cost, which depends on what activities are being traded off (Cesario 1976). Patient regular waiting and provider regular idling are often traded off for work time, and hence these two time components may be valued at individual wage rates (Becker 1965). In particular, the previous literature suggests that the provider unit regular idling cost is around 10 times of the patient unit regular waiting cost (Zacharias and Pinedo 2014, Robinson and Chen 2010), and the provider unit overtime cost is around 15 times of that (because federal laws require that overtime pay rate should not be less than time and one-half regular pay rate). Now, patient leisure waiting and provider recoverable idling are not pure waste and they carry certain utilities for individuals; hence they are less costly than their counterparts. To elicit the marginal values of these time components, one may resort to empirical strategies. For instance, Liu et al. (2018) conduct discrete choice experiments to elicit patient willingness to pay for waiting. Hathaway et al. (2021) employ structural estimation to quantify how callers trade off between waiting for callbacks (akin to leisure waiting) and waiting in queue (similar to regular waiting). Finally, it is important to note that estimating the exact costs of waiting may be more challenging than those of idling because measuring provider productivity is generally more straightforward for management. Whereas we will not address these empirical questions in this work, we conduct an extensive sensitivity analysis based on a wide range of cost parameters to test our model (more on this in Section 6).

3.1. Structural Properties

When $N = 1$, problem (**P**) becomes a traditional single-stage appointment scheduling problem. The objective function can be shown to be L-convex or multimodular, and there exist polynomial local search algorithms to find the global optimal solution (Begen and Queyranne 2011, Zacharias and Yunes 2020, Wang et al. 2020). A well-known technique to tackle this classic problem is to reformulate it via an exact convex relaxation by rewriting $s_k = \max\{a_k, s_{k-1} + \delta_{k-1}\}$ as $s_k \geq a_k$ and $s_k \geq s_{k-1} + \delta_{k-1}$ (Denton and Gupta 2003). When $N \geq 2$, however, problem (**P**) becomes much more complicated. We start by introducing some definitions before our analysis.

Definition 1 (Hajek 1985). *A function $f : \mathbb{R}^k \to \mathbb{R}$ is submodular if and only if $f(\boldsymbol{x}) + f(\boldsymbol{x} + \boldsymbol{e}^i + \boldsymbol{e}^j) \leq f(\boldsymbol{x} + \boldsymbol{e}^i) + f(\boldsymbol{x} + \boldsymbol{e}^j)$ for all $x \in \mathbb{R}^k$ and for all $i \neq j$, where $\boldsymbol{e}^i$ be a unit vector whose i^{th} element is 1.*

Definition 2 (Fujishige and Murota 1997). *A function $f : \mathbb{R}^k \to \mathbb{R}$ is L-convex, if and only if $f(\boldsymbol{x})$ is submodular and $f(\boldsymbol{x} + \mathbf{1}) = f(\boldsymbol{x}) + Q$ for a constant Q, where $\mathbf{1}$ is a vector of ones.*

Lemma 1 (Murota 2004). *L-convexity guarantees local optimality and exact convex relaxation.*

Built upon these preliminaries, the following proposition examines the discrete convex properties of problem (**P**).

Proposition 1 (Discrete Convexity)**.**

1. For any given N, K, $\boldsymbol{T}$ and strictly positive cost coefficients ($C_{W,k}$, $C_{L,k}$, C_I^n, $C_R^n > 0$), there exist service time distributions such that the total regular waiting cost $\sum_{k=1}^K C_{W,k}\Gamma_{W,k}(\boldsymbol{a})$, the total leisure waiting cost $\sum_{k=1}^K C_{L,k}\Gamma_{L,k}(\boldsymbol{a})$, the total regular idling cost $\sum_{n=1}^N C_I^n\Gamma_I^n(\boldsymbol{a})$, and the total recoverable idling cost $\sum_{n=1}^N C_R^n\Gamma_R^n(\boldsymbol{a})$ are neither submodular nor L-convex on $\boldsymbol{a}$.

2. For any given N, K, $\boldsymbol{T}$ and C_O^n, the total overtime cost $\sum_{n=1}^N C_O^n\Gamma_O^n(\boldsymbol{a})$ is submodular and L-convex on $\boldsymbol{a}$ under any service time distribution.

3. For any given N, K and $\boldsymbol{T}$, there exist cost parameters $(C_{W,k}, C_{L,k}, C_I^n, C_R^n, C_O^n | k = 1,\dots,K, n=1,\dots,N)$ and a service time distribution such that the total cost $\sum_{k=1}^K \big(C_{W,k}\Gamma_{W,k}(\boldsymbol{a}) + C_{L,k}\Gamma_{L,k}(\boldsymbol{a})\big) + \sum_{n=1}^N \big(C_I^n\Gamma_I^n(\boldsymbol{a}) + C_R^n\Gamma_R^n(\boldsymbol{a}) + C_O^n\Gamma_O^n(\boldsymbol{a})\big)$ is neither submodular nor L-convex on $\boldsymbol{a}$.

Proposition 1 shows that the total overtime cost is submodular and L-convex, whereas the total waiting cost and the total idling cost are not necessarily so. Therefore, the objective function of (**P**) needs not to be submodular or L-convex. In the next lemma, however, we identify a special case where the discrete convex properties are guaranteed for the objective function of (**P**).

Lemma 2. *If $C_{L,k} = C_{W,k}$, $\forall k$ and $C_R^n = C_I^n$, $\forall n$, then the total cost $\sum_{k=1}^K \big(C_{W,k}\Gamma_{W,k}(\boldsymbol{a}) + C_{L,k}\Gamma_{L,k}(\boldsymbol{a})\big) + \sum_{n=1}^N \big(C_I^n\Gamma_I^n(\boldsymbol{a}) + C_R^n\Gamma_R^n(\boldsymbol{a}) + C_O^n\Gamma_O^n(\boldsymbol{a})\big)$ is submodular and L-convex on $\boldsymbol{a}$ for any given N, K, $\boldsymbol{T}$ and under any given service time distribution.*

These results indicate that, unless $C_{L,k} = C_{W,k}$ and $C_R^n = C_I^n$, problem (**P**) does not possess those useful discrete convex properties as the classic single-stage problem. The key driver for losing these properties is that we include the ready times $\boldsymbol{r}$ of patients and the available times $\boldsymbol{v}$ of providers into the model. After reorganizing the waiting cost and the idling cost, we will have $-(C_{W,k} - C_{L,k})r_k^n$ and $-(C_I^n - C_R^n)v_k^n$ in the objective function. When $C_{L,k} < C_{W,k}$, the sign of r_k^n is strictly negative in the objective function. Hence the common trick of relaxing $r_k^n = \max\{a_k^n, s_k^{n-1} + \delta_k^{n-1}\}$ to $r_k^n \geq a_k^n$ and $r_k^n \geq s_k^{n-1} + \delta_k^{n-1}$ does not work. Similarly, we cannot do such a relaxation for v_k^n when $C_R^n < C_I^n$.

Without those useful discrete convexity properties, solving (**P**) is challenging. One naive idea is enumeration, via which we need to recursively calculate the distributions of $\boldsymbol{r}$, $\boldsymbol{v}$, $\boldsymbol{s}$ and $\boldsymbol{f}$, and then evaluate the expected total cost for each schedule $\boldsymbol{a}$. This process can be very time-consuming if not infeasible. In the next section, we will introduce two reformulations to tackle the problem and develop efficient solution approaches.

4. Reformulation and Exact Solution Approach

Both reformulations in this section are based on sample path representation. We use Ω to denote the set of all possible sample scenarios for $\boldsymbol{\delta}$. Let $\omega \in \Omega$ be an arbitrary sample scenario, and $\boldsymbol{\delta}(\omega)$ be the vector of service durations under scenario ω. When solving the problem practically, we adopt the commonly-used Sample Average Approximation (SAA) approach to randomly generate a sufficient number of samples to represent Ω and then to minimize the average cost of these samples.

4.1. Mixed Binary Integer Linear Programming

Recall that the major obstacle to reformulate problem (**P**) as an equivalent linear program is that we cannot relax $r_k^n = \max\{a_k^n, s_k^{n-1} + \delta_k^{n-1}\}$ and $v_k^n = \max\{a_k^n, s_{k-1}^n + \delta_{k-1}^n\}$ using simple linear constraints. But these two constraints can be linearized using the classic "Big M" method via binary variables (Lieberman and Hillier 2005). The following proposition presents an equivalent Mixed Binary Integer Linear Programming (MBILP) formulation to problem (**P**).

Proposition 2. *The following MBILP is equivalent to* (**P**)*.*

$$\min \; \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \Big[\sum_{k=1}^{K} \big(C_{W,k} \mathit{\Gamma}_{W,k}(\omega) + C_{L,k} \mathit{\Gamma}_{L,k}(\omega) \big) + \sum_{n=1}^{N} \big(C_I^n \mathit{\Gamma}_I^n(\omega) + C_R^n \mathit{\Gamma}_R^n(\omega) + C_O^n \mathit{\Gamma}_O^n(\omega) \big) \Big] \tag{MILP}$$

$$\text{s.t. } s_k^n(\omega) \ge r_k^n(\omega), \quad \forall\, k,\ n,\ \omega, \tag{11}$$

$$s_k^n(\omega) \ge v_k^n(\omega), \quad \forall\, k,\ n,\ \omega, \tag{12}$$

$$r_k^1(\omega) = a_k^1, \quad \forall\, k,\ \omega,$$

$$r_k^n(\omega) \ge a_k^n, \quad \forall\, k,\ n \ge 2,\ \omega, \tag{13}$$

$$r_k^n(\omega) \ge s_k^{n-1}(\omega) + \delta_k^{n-1}(\omega), \quad \forall\, k,\ n \ge 2,\ \omega, \tag{14}$$

$$r_k^n(\omega) \le a_k^n + M(1 - z_k^n(\omega)), \quad \forall\, k,\ n \ge 2,\ \omega, \tag{15}$$

$$r_k^n(\omega) \le s_k^{n-1}(\omega) + \delta_k^{n-1}(\omega) + M z_k^n(\omega), \quad \forall\, k,\ n \ge 2,\ \omega, \tag{16}$$

$$v_1^n(\omega) = a_1^n, \quad \forall\, n,\ \omega, \tag{17}$$

$$v_k^n(\omega) \ge a_k^n, \quad \forall\, k \ge 2,\ n,\ \omega, \tag{18}$$

$$v_k^n(\omega) \ge s_{k-1}^n(\omega) + \delta_{k-1}^n(\omega), \quad \forall\, k \ge 2,\ n,\ \omega, \tag{19}$$

$$v_k^n(\omega) \le a_k^n + M(1 - y_k^n(\omega)), \quad \forall\, k \ge 2,\ n,\ \omega, \tag{20}$$

$$v_k^n(\omega) \le s_{k-1}^n(\omega) + \delta_{k-1}^n(\omega) + M y_k^n(\omega), \quad \forall\, k \ge 2,\ n,\ \omega, \tag{21}$$

$$f^n(\omega) \ge T^n, \quad \forall\, n,\ \omega, \tag{22}$$

$$f^n(\omega) \ge s_K^n(\omega) + \delta_K^n(\omega), \quad \forall\, n,\ \omega, \tag{23}$$

$$a_k^n \ge a_{k-1}^n, \quad \forall\, k \ge 2,\ n, \tag{24}$$

$$a_k^n \ge a_k^{n-1}, \quad \forall\, k,\ n \ge 2, \tag{25}$$

$$\mathit{\Gamma}_{W,k}(\omega),\ \mathit{\Gamma}_{L,k}(\omega),\ \mathit{\Gamma}_I^n(\omega),\ \mathit{\Gamma}_R^n(\omega),\ \mathit{\Gamma}_O^n(\omega) \text{ are defined in } (4),(5),(6),(7),(9), \text{ respectively, } \forall\, \omega,$$

$$a_1^1 = 0;\ 0 \le a_k^n \le T^n;\ s_k^n(\omega),\ r_k^n(\omega),\ v_k^n(\omega),\ f^n(\omega) \ge 0;\ z_k^n(\omega),\ y_k^n(\omega) \in \{0,1\}, \quad \forall\, k,\ n,\ \omega.$$

In problem (**MILP**), the decision variables are $\boldsymbol{a}$, $\boldsymbol{s}$, $\boldsymbol{v}$, $\boldsymbol{r}$, $\boldsymbol{f}$, $\boldsymbol{y}$ and $\boldsymbol{z}$; constraints (11)-(12) relax the max operator in (3); constraints (13)-(14) relax the max operator in (1); constraints (15)-(16) enforce that $r_k^n(\omega)$ cannot exceed both a_k^n and $s_k^{n-1} + \delta_k^{n-1}$; constraints (18)-(19) relax the max operator in (2); constraints (20)-(21) enforce that $v_k^n(\omega)$ cannot exceed both a_k^n and $s_{k-1}^n + \delta_{k-1}^n$; constraints (22)-(23) relax the max operator in (8).

By reformulating the original problem (**P**) as (**MILP**), we make a challenging problem amenable by many off-the-shelf optimization software packages such as Gurobi. Directly solving (**MILP**) via optimization software is clearly one solution approach, but this method does not take advantage of the structure of the original problem (**P**). Furthermore, we have $2NK|\Omega|$ binary variables in problem (**MILP**), suggesting that it can be time-consuming to solve large-scale instances.

4.2. Concave Minimization over Polyhedron

In the second reformulation, we get rid of integer constraints and make the objective function and the constraint set simpler.

Proposition 3. *Problem* (**P**) *is equivalent to the following concave programming problem.*

$$\min\ \frac{1}{|\Omega|}\sum_{\omega\in\Omega}\Big\{\sum_{k=1}^{K}\Big[C_{W,k}\big(s_k^N(\omega)-a_k^1-\sum_{n=1}^{N-1}\delta_k^n(\omega)\big)-(C_{W,k}-C_{L,k})\sum_{n=2}^{N}\max\{a_k^n-s_k^{n-1}(\omega)-\delta_k^{n-1}(\omega),0\}\Big]$$

$$+\sum_{n=1}^{N}\Big[C_I^n\big(s_K^n(\omega)-a_1^n-\sum_{k=1}^{K-1}\delta_k^n(\omega)\big)-(C_I^n-C_R^n)\sum_{k=2}^{K}\max\{a_k^n-s_{k-1}^n(\omega)-\delta_{k-1}^n(\omega),0\}+C_O^n(f^n(\omega)-T^n)\Big]\Big\} \tag{\textbf{CP}}$$

$$\text{s.t. } s_k^n(\omega)\ge a_k^n,\quad \forall\ k,\ n,\ \omega, \tag{26}$$

$$s_k^n(\omega)\ge s_{k-1}^n(\omega)+\delta_{k-1}^n(\omega),\quad \forall\ k\ge 2,\ n,\ \omega, \tag{27}$$

$$s_k^n(\omega)\ge s_k^{n-1}(\omega)+\delta_k^{n-1}(\omega),\quad \forall\ k,\ n\ge 2,\ \omega, \tag{28}$$

$$(22),(23),(24),(25),$$

$$0\le a_k^n\le T^n;\ s_k^n(\omega),\ f^n(\omega)\ge 0,\quad \forall\ k,\ n,\ \omega. \tag{29}$$

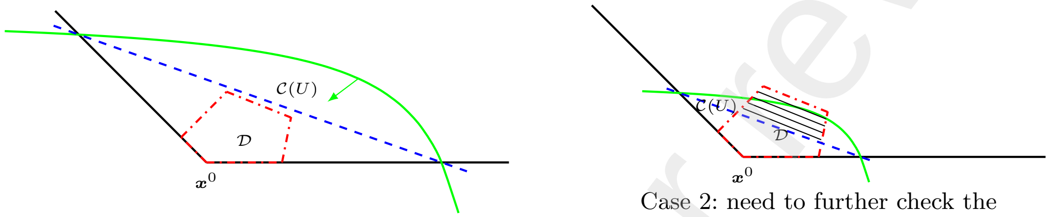
In this reformulation (**CP**), the decision variables are $\boldsymbol{a}$, $\boldsymbol{s}$ and $\boldsymbol{f}$, the objective function is a concave function, and all constraints are linear ones which construct a polyhedron. Without loss of generality, we can impose upper bounds such that $s_k^n(\omega)\le\overline{S}$ and $f^n(\omega)\le\overline{F}$ for some sufficiently large $\overline{S}$ and $\overline{F}$ to make this polyhedron a polytope (which is bounded). In the remainder of this section, we will use the matrix form of the problem (**CP**) for notational convenience. Specifically, let $\boldsymbol{x}$ denote any feasible solution $(\boldsymbol{a},\boldsymbol{s},\boldsymbol{f})$ of the problem (**CP**). Let $\mathcal{D}=\{\boldsymbol{x}|\boldsymbol{A}\boldsymbol{x}\ge\boldsymbol{b}_1,\boldsymbol{x}\le\boldsymbol{b}_2\}$ denote its feasible region, i.e., the polytope. Finally, let $\varGamma(\boldsymbol{x})$, i.e., $\varGamma(\boldsymbol{a},\boldsymbol{s},\boldsymbol{f})$, denote the objective function. Minimizing a concave function $\varGamma(\boldsymbol{x})$ over a polytope $\mathcal{D}$ has been studied previously (Tuy 2016). The following lemma shows some important properties of such problems, based on which solution approaches can be developed.

Lemma 3 (Tuy 2016). *Consider minimizing a concave objective function $\varGamma(\boldsymbol{x})$ over a polytope $\mathcal{D}$.*

1. *The global minimum of $\varGamma(\boldsymbol{x})$ over $\mathcal{D}$ must be attained at one vertex of $\mathcal{D}$.*
2. *For any real number U, denote the level set $\{\boldsymbol{x}|\varGamma(\boldsymbol{x})\ge U\}$ as $\mathcal{C}(U)$. Then $\mathcal{C}(U)$ is convex.*

To find the optimal solution, Lemma 3.1 guarantees that we only need to check the basic feasible solutions in the simplex of $\mathcal{D}$. Denote the optimal objective value of (**CP**) as $\varGamma^*$. Lemma 3.2

implies that, given the current best solution $\boldsymbol{x}$ which yields an upper bound $U = \mathit{\Gamma}(\boldsymbol{x})$ for $\mathit{\Gamma}^*$, we can check whether $\mathit{\Gamma}^* = U$ by developing a cut. Figure 2 illustrates two possible situations. The convex region in the red dash-dotted polytope is the feasible region $\mathcal{D}$; the region defined by the green solid curve is the upper level set $\mathcal{C}(U)$ which is typically not fully known; and the blue dashed line is a hyperplane which can be a cut. In case 1 (left), $\boldsymbol{x}$ is confirmed to be the optimal solution, as the hyperplane cuts all feasible solutions, i.e., the cut establishes that the halfspace below the blue dashed line is surely a subset of the upper level set $\mathcal{C}(U)$. In case 2 (right), only the shadow region need to be further checked and the white region in $\mathcal{C}(U)$ is cut.



Case 1: $\mathit{\Gamma}^* = U$.

Case 2: need to further check the shadow region.

Figure 2 Separation Illustration - I

Given a solution $\boldsymbol{x}$, we need to either show $\mathcal{D} \subseteq \mathcal{C}(U)$ (and thus establish the optimality) or find another solution $\boldsymbol{x}'$ such that $\boldsymbol{x}' \in \mathcal{D} \setminus \mathcal{C}(U)$ and $\Gamma(\boldsymbol{x}') < \Gamma(\boldsymbol{x})$ (and thus a better solution is found). Proper cuts (i.e., blue dashed lines in Figure 2) are helpful to achieve these goals. If one can continue generating such cuts, eventually an optimal solution will be found. Inspired by Tuy (2016), we develop an algorithm to solve the problem (**CP**). The idea is to sequentially develop cuts, partition the feasible region into sub-regions if necessary, and check optimality therein. Leveraging the structure of the problem, we further augment the algorithm by providing an efficient way to generate tighter bounds on the objective value.

We first explain how to generate these cuts. We start with a cone vertexed at a vertex of $\mathcal{D}$, having exactly $|\boldsymbol{x}|$ edges and containing $\mathcal{D}$. Lemma 4 shows that how to initialize such a cone.

Lemma 4. *Let $\boldsymbol{x}^0$ be a vertex of $\mathcal{D}$. Let $\boldsymbol{w}^0 = (\boldsymbol{w}_1^0, \boldsymbol{w}_2^0)$ be the slack variables such that $\boldsymbol{A}\boldsymbol{x}^0 - \boldsymbol{w}_1^0 = \boldsymbol{b}_1$ and $\boldsymbol{x}^0 + \boldsymbol{w}_2^0 = \boldsymbol{b}_2$. Let $\boldsymbol{y}^0 = (\boldsymbol{x}^0, \boldsymbol{w}^0)$ denote the corresponding basic solution. Let $[\boldsymbol{I}, \boldsymbol{W}]$ denote the associated simplex tableau, yielding $\boldsymbol{y}_B^0 = \boldsymbol{y}_B + \boldsymbol{W}\boldsymbol{y}_N$, where $\boldsymbol{B}$ and $\boldsymbol{N}$ denote the basis and non-basic columns, respectively. Let $\boldsymbol{U}$ be a $|\boldsymbol{x}| \times |\boldsymbol{x}|$ matrix. For $i = 1, 2, \cdots, |\boldsymbol{x}|$ and $j \in \boldsymbol{N}$, define*

$$\boldsymbol{U}_{ij} = \begin{cases} -\boldsymbol{W}_{ij} & \text{if } i \in \boldsymbol{B}, \\ 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

Then the cone $\mathcal{M} = \{\boldsymbol{x} | \boldsymbol{x} = \boldsymbol{x}^0 + \boldsymbol{U}\boldsymbol{t}, \boldsymbol{t} \geq \boldsymbol{0}\}$ is vertexed at $\boldsymbol{x}^0$ and has $|\boldsymbol{x}|$ edges. There must be one vertex of $\mathcal{D}$ on each edge of $\mathcal{M}$, and $\mathcal{M}$ must contain $\mathcal{D}$.

Note that each column of $\boldsymbol{U}$ is the direction of an edge. Let $\boldsymbol{u}^i$ denote the direction of edge i and $\partial\mathcal{C}(U)$ denote the boundary of $\mathcal{C}(U)$. Let θ_i be such that $\boldsymbol{x}^0+\theta_i\boldsymbol{u}^i\in\partial\mathcal{C}(U)$ for each $i=1,\dots,|\boldsymbol{x}|$. Then $\boldsymbol{x}^0+\theta_i\boldsymbol{u}^i$ is the point where $\mathcal{C}(U)$ meets edges i of the cone. This observation is formalized in the following lemma.

Lemma 5. *For any cone $\mathcal{M}=\{\boldsymbol{x}|\boldsymbol{x}=\boldsymbol{x}^0+\boldsymbol{U}\boldsymbol{t},\boldsymbol{t}\geq\boldsymbol{0}\}$ with $|\boldsymbol{x}|$ edges and $\boldsymbol{x}^0\in\mathcal{C}(U)$, let θ_i be such that $\boldsymbol{x}^0+\theta_i\boldsymbol{u}^i\in\partial\mathcal{C}(U-\epsilon)$ where ϵ is a small number. Then the following equality*

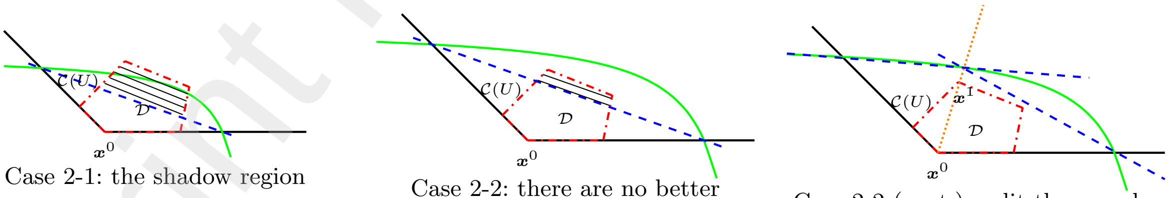
$$\sum_{i=1}^{|\boldsymbol{x}|}\frac{t_i}{\theta_i}=1,$$

where $\boldsymbol{t}=\boldsymbol{U}^{-1}(\boldsymbol{x}-\boldsymbol{x}^0)$, defines the hyperplane passing through all points where the edges of cone $\mathcal{M}$ meet the boundary of $\mathcal{C}(U-\epsilon)$. And any point $\boldsymbol{x}\in\mathcal{D}$ such that $\Gamma(\boldsymbol{x})<U-\epsilon$ must lie in the halfspace

$$\sum_{i=1}^{|\boldsymbol{x}|}\frac{t_i}{\theta_i}\geq 1,$$

where $\boldsymbol{t}=\boldsymbol{U}^{-1}(\boldsymbol{x}-\boldsymbol{x}^0)$.

Recall that we want to check whether $\mathcal{D}\subseteq\mathcal{C}(U)$ or find another solution $\boldsymbol{x}'$ such that $\boldsymbol{x}'\in\mathcal{D}\setminus\mathcal{C}(U)$. If $\boldsymbol{x}^0$ is the current solution, one idea is to find a vertex of $\mathcal{D}$ which is as far away from $\boldsymbol{x}^0$ as possible, i.e., to maximize $\sum_{i=1}^{|\boldsymbol{x}|}\frac{t_i}{\theta_i}$. We will illustrate how to obtain such a vertex later. Then we may have two cases shown in Figure 3 below. In case 2-1 (left), the solution which gives the maximum $\sum_{i=1}^{|\boldsymbol{x}|}\frac{t_i}{\theta_i}$ is one of the upper right corners of $\mathcal{D}$ (and outside of $\mathcal{C}(U)$); this solution is better than the current one and we can then update the current best solution and bound. In case 2-2 (middle & right), this solution is not better, and we need to split the cone to two sub-cones via a vertex in the shadow region. In the following, we illustrate how to obtain such a vertex.



Case 2-1: the shadow region contains better solutions.

Case 2-2: there are no better solutions in the shadow region.

Case 2-2 (cont.): split the cone by vertex $\boldsymbol{x}^1$ and check each sub-cone.

Figure 3 Separation Illustration - II

For any cone $\mathcal{M}=\{\boldsymbol{x}|\boldsymbol{x}=\boldsymbol{x}^0+\boldsymbol{U}\boldsymbol{t},\boldsymbol{t}\geq\boldsymbol{0}\}$ with $|\boldsymbol{x}|$ edges, solving the following linear programming (**LP-$\mathcal{DM}$**) yields the vertex of $\mathcal{D}\cap\mathcal{M}$ which is the farthest away from $\boldsymbol{x}^0$.

$$\max\{\sum_{i=1}^{|\boldsymbol{x}|}\frac{t_i}{\theta_i}|\boldsymbol{A}\boldsymbol{U}\boldsymbol{t}\geq\boldsymbol{b}_1-\boldsymbol{A}\boldsymbol{x}^0,\boldsymbol{U}\boldsymbol{t}\leq\boldsymbol{b}_2-\boldsymbol{x}^0,-\boldsymbol{U}\boldsymbol{t}\leq\boldsymbol{x}^0,\boldsymbol{t}\geq 0\}.\qquad\textbf{(LP-}\mathcal{DM}\textbf{)}$$

The first three constraints are derived from the original constraints in (**CP**) by replacing $\boldsymbol{x}$ with $\boldsymbol{x}^0 + \boldsymbol{U}\boldsymbol{t}$. Denote the optimal solution of (**LP-**$\mathcal{DM}$) by $\boldsymbol{t}^{\mathcal{M}}$ and the optimal objective value by $\psi^{\mathcal{M}}$. Let $\boldsymbol{x}^{\mathcal{M}} = \boldsymbol{x}^0 + \boldsymbol{U}\boldsymbol{t}^{\mathcal{M}}$. Then we have the following lemma, of which the different cases are illustrated in Figures 2 and 3.

Lemma 6. *$\boldsymbol{x}^{\mathcal{M}}$ must be a vertex of $\mathcal{D} \cap \mathcal{M}$. We have the following three cases:*

Case 1: $\psi^{\mathcal{M}} \leq 1$.

Case 2-1: $\psi^{\mathcal{M}} > 1$ and $\Gamma(\boldsymbol{x}^{\mathcal{M}}) < U$.

Case 2-2: $\psi^{\mathcal{M}} > 1$ and $\Gamma(\boldsymbol{x}^{\mathcal{M}}) \geq U$. In this case, $\boldsymbol{x}^{\mathcal{M}}$ does not lie on any edge of $\mathcal{M}$.

In Case 1, since $\psi^{\mathcal{M}} \leq 1$, then $\mathcal{D} \cap \mathcal{M} \subset \mathcal{C}(U - \epsilon)$, i.e., $\Gamma(\boldsymbol{x}) \geq U - \epsilon$ for all points $\boldsymbol{x} \in \mathcal{D} \cap \mathcal{M}$. The cone $\mathcal{M}$ can be removed from consideration, since all feasible solutions in $\mathcal{D} \cap \mathcal{M}$ are sub-optimal. In Case 2-1, we find a better solution $\boldsymbol{x}^{\mathcal{M}}$ which is a vertex in $\mathcal{D} \setminus \mathcal{C}(U)$, then we can update the current best solution and bound, and restart with this new vertex. These two cases give rise to a valid cut to (**CP**), explained by the following corollary.

Corollary 1. *If $\psi^{\mathcal{M}} > 1$, then*

$$\boldsymbol{\theta}^{-1}\boldsymbol{U}^{-1}\boldsymbol{x} \geq \boldsymbol{\theta}^{-1}\boldsymbol{U}^{-1}x^0 + 1, \tag{30}$$

where $\boldsymbol{\theta}^{-1} = (\frac{1}{\theta_1}, \frac{1}{\theta_2}, \ldots, \frac{1}{\theta_{|\boldsymbol{x}|}})$, is a valid cut to (**CP**).

Finally, if Case 2-2 happens, we may split the cone $\mathcal{M}$ to sub-cones $\mathcal{M}^1$, $\mathcal{M}^2$, ..., $\mathcal{M}^{|\boldsymbol{x}|}$ at the vertex $\boldsymbol{x}^{\mathcal{M}}$, and divide the problem to a set of sub-problems. Lemma 7 shows how to split a cone.

Lemma 7. *Given a cone $\mathcal{M} = \{\boldsymbol{x} | \boldsymbol{x} = \boldsymbol{x}^0 + \boldsymbol{U}\boldsymbol{t}, \boldsymbol{t} \geq \boldsymbol{0}\}$ and a vertex $\boldsymbol{x}^{\mathcal{M}}$ which does not lie on any edge of $\mathcal{M}$, the new direction of the ray from $\boldsymbol{x}^0$ through $\boldsymbol{x}^{\mathcal{M}}$ is $\boldsymbol{x}^{\mathcal{M}} - \boldsymbol{x}^0 = \boldsymbol{U}\boldsymbol{t}^{\mathcal{M}}$. The original $|\boldsymbol{x}|$ directions are columns of $\boldsymbol{U}$, i.e., $\boldsymbol{u}^1$, $\boldsymbol{u}^2$, ...,$\boldsymbol{u}^{|\boldsymbol{x}|}$. Combine any $|\boldsymbol{x}| - 1$ of them and $\boldsymbol{U}\boldsymbol{t}^{\mathcal{M}}$, we can get $|\boldsymbol{x}|$ new matrices $\boldsymbol{U}^1$, $\boldsymbol{U}^2$, ..., $\boldsymbol{U}^{|\boldsymbol{x}|}$. These $|\boldsymbol{x}|$ new matrices define $|\boldsymbol{x}|$ sub-cones $\mathcal{M}^1$, $\mathcal{M}^2$, ..., $\mathcal{M}^{|\boldsymbol{x}|}$. We call the set of $\mathcal{M}^1$, $\mathcal{M}^2$, ..., $\mathcal{M}^{|\boldsymbol{x}|}$ as a partition of $\mathcal{M}$ by $\boldsymbol{x}^{\mathcal{M}}$, denoted by $\mathcal{P}^{\mathcal{M}}$.*

Remark 3. *The cut* (30) *remains valid when investigating these sub-cones.*

If we fall into Case 2-1 more often but Case 2-2 less often, then we may find the global optimal solution faster. Having a tighter bound U is helpful. Thanks to the structure of the original problem (**P**), we may easily obtain such a bound for a given feasible solution by solving a linear program.

Proposition 4. *Given any feasible solution $\boldsymbol{x} = (\boldsymbol{a}, \boldsymbol{s}, \boldsymbol{f})$ of* (**CP**)*, solving the following linear program yields $(\boldsymbol{s}', \boldsymbol{f}')$ such that $(\boldsymbol{a}, \boldsymbol{s}', \boldsymbol{f}')$ is feasible and $\Gamma(\boldsymbol{a}, \boldsymbol{s}', \boldsymbol{f}') \leq \Gamma(\boldsymbol{x})$.*

$$\min_{\boldsymbol{s}(\omega)\geq 0, \boldsymbol{f}(\omega)\geq 0} \left\{ \frac{1}{|\Omega|} \sum_{\omega} \sum_{n=1}^{N} [f^n(\omega) + \sum_{k=1}^{K} s_k^n(\omega)] \middle| (26), (27), (28), (22), (23) \right\}. \tag{\textbf{Sub-LP}}$$

For a given $\boldsymbol{x}^{\mathcal{M}}$, solving the corresponding problem (**Sub-LP**) may yield a better solution and a smaller objective value, making it more likely to have Case 2-1 than Case 2-2. This also makes the updated upper bound tighter.

Following the discussion above, we now present the details of our algorithm, called the Cone Split and Cut Algorithm, to solve (**CP**). (See Algorithm 1 in the Online Appendix B for its pseudocode.) In Step 1, we use $\boldsymbol{A}^o$ and $\boldsymbol{b}_1^o$ to denote the original constraints without valid cuts. In Step 2, we obtain an initial vertex of $\mathcal{D}$ as well as the corresponding objective value denoted by U. It can be obtained by relaxing the objective function in (**CP**) to a linear function and solving the corresponding linear program. In Step 3, we obtain the associated cone $\mathcal{M}^0$ with $\boldsymbol{x}^0$ by Lemma 4 and initialize $\mathcal{P}$ with it ($\mathcal{P}$ is the current set of cones yet to be investigated). Then for each cone in $\mathcal{P}$, we update the constraints with its corresponding cuts, if available, in Step 6. Steps 7 and 8 find the level set $\mathcal{C}(U-\epsilon)$ and get a new vertex which is as far away from the original vertex as possible. If the new vertex is in the level set, then we have case 1 in Lemma 6, i.e., the feasible region is a subset of the level set and we will remove this cone from $\mathcal{P}$ (Step 11). If the new vertex is outside the level set, in Step 13 we add a cut, and then we check whether the new vertex is better: if it is better, we update the original vertex with the new one (Step 15) and restart from Step 3; otherwise, we move to the next cone in $\mathcal{P}$. After all cones in $\mathcal{P}$ are examined, we check whether $\mathcal{P}$ is empty. An empty $\mathcal{P}$ means that we have considered all potential cones and the current best solution is optimal. The algorithm terminates (Step 21). If $\mathcal{P}$ is not empty, we pick one to split, mark the current cuts, add the sub-cones into $\mathcal{P}$ in Steps 23 and 24. Then we restart from Step 4.

Proposition 5. *The Cone Split and Cut Algorithm (CSCA) terminates after finitely many steps, yielding an ϵ-optimal solution of* (**CP**).

In Section 6, we will examine the performance of our proposed CSCA by comparing it against several benchmarks including some of the best off-the-shelf optimization packages.

5. Approximation Solution Approach

We discuss exact solution approaches in the last section. In this section, we propose an approximation solution approach. Recall that $r_k^n = \max\{a_k^n, s_k^{n-1}+\delta_k^{n-1}\}$, $\forall n \geq 2$ and $v_k^n = \max\{a_k^n, s_{k-1}^n + \delta_{k-1}^n\}$, $\forall k \geq 2$ are the key challenging terms to deal with. Instead of using the Big M method to obtain an exact reformulation (**MILP**), we just use $s_k^{n-1}+\delta_k^{n-1}$ to approximate r_k^n for $n \geq 2$ and use $s_{k-1}^n + \delta_{k-1}^n$ to approximate v_k^n for $k \geq 2$. That is, this approximation asserts that a patient is ready for service in the next station once he is done with the prior station and the provider is available to serve the next patient once she is done with the prior patient. Then the original problem (**CP**) can be relaxed to the following linear program (**APR**), which is more tractable. Furthermore, this simple approximation approach leads to an easily implementable scheduling policy, and it also has provable performance guarantees.

$$\min\ \frac{1}{|\Omega|}\sum_{\omega\in\Omega}\Big\{\sum_{k=1}^{K}\Big[C_{W,k}\big(s_k^N(\omega)-a_k^1-\sum_{n=1}^{N-1}\delta_k^n(\omega)\big)\Big]+\sum_{n=1}^{N}\Big[C_I^n\big(s_K^n(\omega)-a_1^n-\sum_{k=1}^{K-1}\delta_k^n(\omega)\big)+C_O^n\big(f^n(\omega)-T^n\big)\Big]\Big\} \tag{\textbf{APR}}$$

$$\text{s.t.}\quad (26), (27), (28), (22), (23), (24), (25), (29).$$

The following proposition outlines the simple structure of the optimal solution to (**APR**), which also explains why this solution can be useful for practice.

Proposition 6. *There exists an optimal schedule $\boldsymbol{a}_R^*$ to* (**APR**) *such that* $a_{R,k}^{n*} = \max\{a_{R,k-1}^{n*}, a_{R,k}^{n-1*}\}$ *for all* $k \geq 2,\ n \geq 2$.

Proposition 6 indicates that to solve (**APR**), one only needs to specify the appointment times for all patients at the first stage—and—the appointment times for the first patient in all subsequent stages. Indeed, the appointment time for the first patient in all stages can be equivalently viewed as the service start times for those stages. So, for each patient other than the first one, his appointment time for a later stage on his visit itinerary can be provided as the maximum of his appointment time at the first stage and the start time of that later stage. It is possible that certain patients are assigned the same appointment time at some stages. To deal with this, one can design personalized appointment notices accordingly for both providers and patients. For providers at different service stages, they would be informed the times at which patients are scheduled. For patients who receive the same appointment time for some stages, they would be informed the appointment time to start this whole "phase" of services and told that services will continue throughout the whole phase; they would be informed another appointment time to start the next phase of services if any.

While this scheduling policy seems simple, it staggers the service start times of subsequent stages and partially captures the interdependencies across multiple stages. For simplicity, we call this policy the *stage-staggering* policy. In addition, because patients are given different appointment times at the first stage, they are unlikely to start service at the same time (and hence they are spaced out) in subsequent stages. Therefore, we can expect this policy to perform well. The next proposition shows that this policy has a provable performance bound. We state our results using the sample average formulations (**CP**) and (**APR**).

Proposition 7. *Consider a given* Ω. *Let* $\Gamma(\boldsymbol{a})$ *and* $\Gamma_R(\boldsymbol{a})$ *be the objective values of* (**CP**) *and* (**APR**), *respectively, under a given schedule* $\boldsymbol{a}$ *and its realized service start times and finished times for all patients at all stages in each sample scenario* $\omega \in \Omega$. *Let* $\boldsymbol{a}^*$ *and* $\boldsymbol{a}_R^*$ *be the optimal schedules to* (**CP**) *and* (**APR**), *respectively. Then we have*

$$\Gamma(\boldsymbol{a}^*) \leq \Gamma(\boldsymbol{a}_R^*) \leq \Gamma_R(\boldsymbol{a}_R^*) \leq \Gamma_R(\boldsymbol{a}^*).$$

Furthermore, the performance loss of $\boldsymbol{a}_R^*$, *i.e.,* $\Gamma(\boldsymbol{a}_R^*) - \Gamma(\boldsymbol{a}^*)$, *is bounded above by*

$$\frac{1}{|\Omega|}\sum_{\omega\in\Omega}\Big[\sum_{k=1}^{K}(C_{W,k}-C_{L,k})\sum_{n=2}^{N}(a_k^{n*}-a_k^{n-1*}-\delta_k^{n-1}(\omega))^+ + \sum_{n=1}^{N}(C_I^n - C_R^n)\sum_{k=2}^{K}(a_k^{n*}-a_{k-1}^{n*}-\delta_{k-1}^{n}(\omega))^+\Big],$$

which is no larger than $\max_{k,n}\{C_{W,k}-C_{L,k}, C_I^n - C_R^n\}(N+K-2)\max_n\{T^n\}$.

Proposition 7 means that the optimality gap of the stage-staggering policy is bounded. And the bound is independent of the overtime cost but shrinks when $(C_{W,k} - C_{L,k})$ and $(C_I^n - C_R^n)$ are smaller. In particular, we have the following corollary.

Corollary 2. *If* $C_{W,k} = C_{L,k}$ *and* $C_I^n = C_R^n$, $\boldsymbol{a}_R^*$ *is optimal.*

If leisure waiting has the same cost rate as regular waiting, and recoverable idling has the same cost rate as regular idling, then this policy is optimal. This result can be explained in the following way. When $C_{L,k}$ and C_R^n increase, the difference between leisure waiting and regular waiting and the difference between recoverable idling and regular idling decrease. Then a_k^n for $k \geq 2$ and $n \geq 2$ becomes less important because its role is to differentiate leisure waiting from regular waiting, and recoverable idling from regular idling. Thus the optimal schedule will choose a_k^n to be closer to $\max\{a_k^{n-1}, a_{k-1}^n\}$. When $C_{L,k} = C_{W,k}$ and $C_I^n = C_R^n$, the optimal schedule will choose to simply minimize total patient waiting and provider idling, thus $a_k^n = \max\{a_k^{n-1}, a_{k-1}^n\}$. To put it another way, the reason why giving appointments in subsequent stages benefits the system is because of the difference between leisure waiting and regular waiting and the difference between recoverable idling and regular idling. Therefore, our tandem scheduling model uncovers the hidden utility of staged appointments by taking into account heterogeneous patient waiting and provider idling costs. If these cost rate differences diminish, there is less benefit of giving subsequent appointments. Hence the approximation approach is close to exact.

Our stage-staggering policy is related to and complements the strategic idling idea of Baron et al. (2014, 2017). They focus on customer experience in queueing networks, and show that strategic idling can decrease wait time and the probability of long waits in a tandem system with two single-server queues or in an open shop. We study how to improve overall efficiency in appointment-based services, and suggest that postponing the service start times of providers may help a tandem service system achieve a better tradeoff between customer waiting and provider utilization.

6. Numerical Study

Our numerical study has three purposes. First, we evaluate the performance of our solution approaches. Specifically, in Section 6.1.1 we test the computational efficiency of our Cone Split and Cut Algorithm and compare it with directly solving the formulation (**MILP**) by the-state-of-the-art optimization software; in Section 6.1.2, we investigate the performance of our approximation solution approach and benchmark it against the exact approaches and those used in practice. Then, we study how different model parameters affect the structure of the optimal patient visit itinerary to reveal managerial insights. Finally, leveraging our tandem scheduling model, we develop a heuristic policy to handle the settings with multiple servers in one stage and test this heuristic in a case study populated by real data obtained from a large infusion center.

6.1. Performance Evaluation of Solution Approaches

6.1.1. Computational Efficiency of CSCA We use a variety of model parameters in our numerical study to capture various settings. Here we test the instances with 2 and 3 stations, i.e.,

$N = 2$ and $N = 3$. We set the number of patients $K = 5$ for $N = 2$ and $K = 6$ for $N = 3$. The service duration follows the two-point discrete uniform distribution with mean 1.5 slots for all stations. Recall that T^n is the length of regular working hours at station n. We set $(T^1, T^2) = (10, 12)$ or $(12, 14)$ for $N = 2$ and $(T^1, T^2, T^3) = (13, 15, 18)$ or $(15, 18, 20)$ for $N = 3$, respectively. The larger the T^n is, the lighter the workload is because the total number of patients is fixed.

For cost parameters, we consider multiple scenarios, and a scenario is defined by a specific mix of mean cost parameters. In order to have a more robust test, we draw random samples of cost parameters for each scenario. Following the discussion on our model parameters in Section 3, we set the mean of unit overtime cost $C_O^n = 15$, the mean of unit regular idling cost $C_I^n = 10$, and the mean of unit regular waiting cost $C_{W,k}$ to be 1 or 3. A larger $C_{W,k}$ means that patient waiting is relatively more important. The difference between C_I^n and C_R^n and that between $C_{W,k}$ and $C_{L,k}$ have an important influence on the problem's computational complexity. When the gap between the two types of waiting/idling costs gets smaller, the problem is closer to its convex approximation and is likely easier to solve. In particular, when $C_{W,k} = C_{L,k}$ and $C_I^n = C_R^n$, the problem can be treated as a linear program (see Proposition 1 and Corollary 2). Here we set the average C_R^n ($C_{L,k}$) to be 1/4, 1/2 or 3/4 of the average C_I^n ($C_{W,k}$). In total, we have 24 scenarios (see Table C.1 in the Online Appendix for a summary of the cost parameters). To evaluate computational times in a robust way, for each scenario, we generate 10 different instances by drawing a random sample of C_O^n, C_I^n, C_R^n, $C_{W,k}$ and $C_{L,k}$ from their respective uniform distributions (note that realized values of a specific cost parameter for different patients/stations may be different although the distribution of this specific cost parameter is the same). For each instance, we solve the formulation (**MILP**) directly via Gurobi 9.0.2 and by our CSCA. Then, we compare the average computational times of these two approaches for each scenario. All computations were conducted on a desktop computer equipped with Intel Core i7 3.0 GHz CPU, 32.00 GB RAM, and 64-bit Windows 10 OS.

Our computational experiments show that, in general, solving (**MILP**) directly quadruples the computational time needed by our proposed Cone Split and Cut Algorithm. The detailed computational times are illustrated in Figure A.1 of Appendix A. When the difference between two types of waiting/idling costs gets smaller, i.e., the problem is closer to its convex approximation, both methods take less time as expected. Whereas patient regular waiting cost does not seem to have a strong influence on computation times of both methods, the workload does. In problem instances with heavier workload, i.e., shorter regular working hours, both methods reach optimality faster likely due to the fact that we have fewer feasible schedules in these instances.

6.1.2. Performance Evaluation of Approximation Solution Approach In this section, we study the performance of our approximation solution approach and how different model parameters affect its performance. We use two other solution approaches as benchmarks. The first one is to solve (**MILP**) directly, like in Section 6.1.1. However, for large-scale problem instances, it is impossible to reach optimality within a reasonable amount of time. So we use the solution output by Gurobi 9.0.2 with 1-hour computational time limit. The second solution approach is the one commonly adopted by practice—solve the optimal schedule for the bottleneck station (which has the largest traffic intensity) while ignoring all other stations and then calculate the appointment times in other stations by adding/subtracting the corresponding mean service time.

Here, we consider 2 stations and 15 patients to schedule. The service time follows the discrete uniform distribution with 1 being the minimum of the support, and the mean service time $(\bar{\delta}_k^1, \bar{\delta}_k^2)$ is $(2, 1.5)$ or $(1.5, 2)$, corresponding to the first or the second station being the bottleneck, respectively. The regular working hours (T^1, T^2) are set as $(25, 30)$ or $(30, 35)$. To fairly and conveniently investigate the impact of different cost parameters on solution performance, we consider deterministic and homogeneous unit costs. Following discussions above, we set $C_O^n = 15$ and $C_I^n = 10$. The unit regular waiting cost $C_{W,k}$ is set to be 1 or 3. For the difference between two types of idling/waiting costs, we set C_R^n/C_I^n to be 0.25 or 0.75, and $C_{L,k}/C_{W,k}$ to be 0, 0.25, 0.75 or 1. We use a wider range for $C_{L,k}/C_{W,k}$ due to the greater challenge in quantifying waiting costs compared to idling costs, as previously discussed. To make discussions more intuitive, we draw upon the following interpretations on these cost ratios. If the clinic has a better *practice environment* for providers (e.g., equipped with better health IT infrastructure and has more supportive administration), then providers can utilize recoverable idling time more efficiently to perform ancillary tasks; that is, the unit recoverable idling cost C_R^n is smaller. Thus, a smaller C_R^n/C_I^n ratio suggests a better practice environment and vice versa. If the clinic has a better *service environment* for patients (e.g., providing patients a pager and having more comfortable space for patients while they wait), then informing patients about leisure waiting is more valuable and meaningful to them as patients can get more out of leisure waiting—they can choose a place to stretch their legs, enjoy a walk, or get a coffee before their next appointments. In this case, the unit leisure waiting cost $C_{L,k}$ is smaller, and we say that a smaller $C_{L,k}/C_{W,k}$ ratio implies a better service environment and vice versa. Table 1 summarizes the managerial implications of different parameters.

Table 2 shows the performance gaps between our approximation solution approach and the two benchmark approaches when the service environment is exceptional or poor. (The results for the cases where the service environment is superior or mediocre reveal similar insights, and are presented in Appendix C.) A negative (positive) percentage gap means that our approximation approach works better (worse) than the corresponding benchmark. On average, our approximation solution approach generates schedules with similar quality as those obtained by solving (**MILP**) directly; the average percentage gap is 1.8%. Diving into specific scenarios, one may expect our approximation approach to perform similarly or even better compared to the MILP approach when the practice environment is mediocre and the service environment is poor, because the bound of

Table 1 Parameter Settings to Test Approximation Solution Approach

	Parameters	Implications
service time	$(\bar{\delta}_k^1, \bar{\delta}_k^2) = (2, 1.5)$	bottleneck station: 1
	$(\bar{\delta}_k^1, \bar{\delta}_k^2) = (1.5, 2)$	bottleneck station: 2
regular working hours	$(T^1, T^2) = (25, 30)$	workload: heavy
	$(T^1, T^2) = (30, 35)$	workload: light
regular waiting cost	$C_{W,k} = 3$	patient waiting: important
	$C_{W,k} = 1$	patient waiting: less important
idling cost ratio	$C_R^n / C_I^n = 0.25$	practice environment: superior
	$C_R^n / C_I^n = 0.75$	practice environment: mediocre
waiting cost ratio	$C_{L,k}/C_{W,k} = 0$	service environment: exceptional
	$C_{L,k}/C_{W,k} = 0.25$	service environment: superior
	$C_{L,k}/C_{W,k} = 0.75$	service environment: mediocre
	$C_{L,k}/C_{W,k} = 1$	service environment: poor

the performance loss shrinks when $C_{W,k} - C_{L,k}$ and $C_I^n - C_R^n$ are smaller (see Proposition 7). The numerical results confirm that, indeed, in such environments our approximation solution approach achieves an average performance gap of -2%.

Nevertheless, the MILP approach can still provide the best schedule even under the computation time constraint. For instance, the MILP approach does outperform our approximation approach by a sizable average gap of 18.5% when the practice environment is superior, the service environment is exceptional, the second station is the bottleneck, and patient waiting is important. Recall that our approximation approach is equivalent to a policy that staggers the service start time of the second station, but does not provide appointments for the second station except for the first patient. Hence, compared to the MILP approach which gives a full visit itinerary, our approximation solution approach may not be as effective in managing patient wait time in the second stage. Thus, the performance gap becomes more significant when the second stage is the bottleneck.

Next, we draw attention to the comparison against the approach adopted in practice. Our approximation solution approach outperforms the one in practice by a remarkable average gap of 20%. The approach in practice yields more efficient schedules only in a very limited set of scenarios where the bottleneck is the second station, the workload is light, patients waiting is less important, and providers have a superior practice environment. One reason is that the schedules generated by our approximation approach in these scenarios are quite compact, whereas the approach in practice generates relatively loose schedules (in the second station) which benefits the provider considerably. This only occurs in 2 out of 32 scenarios we tested. In sum, these comparison studies provide convincing evidence that our approximation solution approach has robust performance and can serve as a reasonable scheduling policy for practical use; however, the MILP approach can still achieve considerable improvement in operational efficiency if the practice has resources to compute and implement that.

Table 2 Performance of Approximation Solution Approach

Bottleneck	Workload	Patient Waiting	Practice Environment	Service Environment	% Gap to MILP	% Gap to Practice
1	heavy	important	superior	exceptional	1.65%	-25.08%
1	heavy	important	superior	poor	0.65%	-22.69%
1	heavy	important	mediocre	exceptional	9.96%	-24.07%
1	heavy	important	mediocre	poor	0.00%	-23.94%
1	heavy	less important	superior	exceptional	-3.96%	-19.86%
1	heavy	less important	superior	poor	-1.33%	-14.90%
1	heavy	less important	mediocre	exceptional	2.83%	-16.76%
1	heavy	less important	mediocre	poor	0.04%	-14.35%
1	light	important	superior	exceptional	4.17%	-26.99%
1	light	important	superior	poor	0.02%	-28.39%
1	light	important	mediocre	exceptional	-0.50%	-29.30%
1	light	important	mediocre	poor	0.00%	-32.70%
1	light	less important	superior	exceptional	1.82%	-21.35%
1	light	less important	superior	poor	2.35%	-23.02%
1	light	less important	mediocre	exceptional	0.40%	-25.86%
1	light	less important	mediocre	poor	0.07%	-28.57%
2	heavy	important	superior	exceptional	22.24%	-23.79%
2	heavy	important	superior	poor	1.93%	-23.91%
2	heavy	important	mediocre	exceptional	-15.17%	-22.24%
2	heavy	important	mediocre	poor	0.26%	-23.21%
2	heavy	less important	superior	exceptional	43.06%	-24.02%
2	heavy	less important	superior	poor	-1.13%	-18.26%
2	heavy	less important	mediocre	exceptional	13.17%	-25.62%
2	heavy	less important	mediocre	poor	-13.06%	-18.91%
2	light	important	superior	exceptional	14.85%	-21.93%
2	light	important	superior	poor	-11.02%	-36.40%
2	light	important	mediocre	exceptional	0.39%	-17.10%
2	light	important	mediocre	poor	-3.13%	-22.08%
2	light	less important	superior	exceptional	-6.95%	20.92%
2	light	less important	superior	poor	-1.92%	7.89%
2	light	less important	mediocre	exceptional	-0.73%	-12.82%
2	light	less important	mediocre	poor	-0.61%	-15.59%

Notes. (1) % Gap to MILP is the performance gap between our approximation solution and the MILP solution with 1-hour computation time limit. It is computed by $\frac{\text{Approximation solution}-\text{MILP solution}}{\text{MILP solution}} \times 100\%$. (2) % Gap to Practice is the performance gap between our approximation solution and the approach in practice. It is computed by $\frac{\text{Approximation solution}-\text{Solution in practice}}{\text{Practical solution}} \times 100\%$.

6.2. Impact of Model Parameters on the Optimal Visit Itinerary

To examine how different model parameters impact the structure of the optimal patient visit itinerary, we conduct a range of sensitivity analysis. Specifically, we vary the location of the bottleneck stage, the relative importance of patient waiting and the difference between two types of idling/waiting costs. We focus on a system with $N = 2$ service stages and $K = 5$ patients to schedule. The regular working hours are set as $(T^1, T^2) = (10, 12)$. We make the same assumption on service time distribution as in Section 6.1.2. To make the pattern more visible, we consider the extreme cases for the provider idling and patient waiting cost rates. Table 3 shows the detailed parameter choices in our sensitivity analysis.

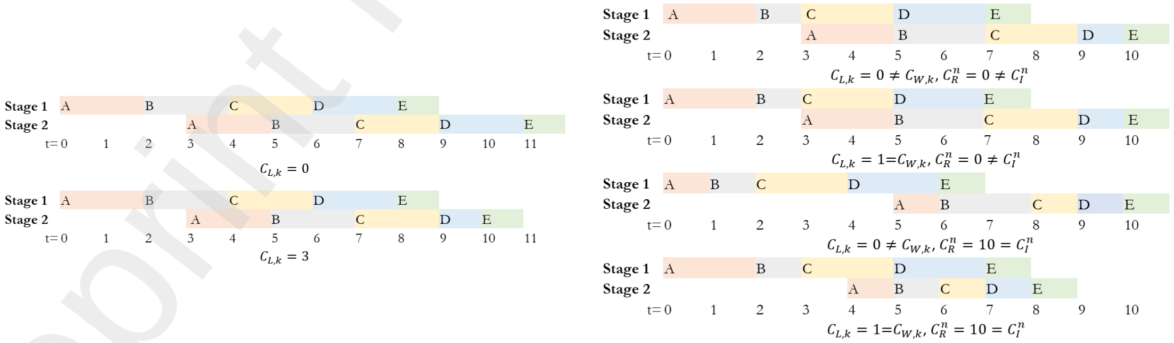
We start by discussing two intuitive observations. First, when the bottleneck moves from the first stage to the second one, the first stage schedule becomes more compact, whereas the second stage

Table 3 Parameter Settings for Sensitivity Analysis

	Parameters	Implications
service times	$(\bar{\delta}_k^1, \bar{\delta}_k^2) = (2,\ 1.5)$	bottleneck: stage 1
	$(\bar{\delta}_k^1, \bar{\delta}_k^2) = (1.5,\ 2)$	bottleneck: stage 2
regular waiting cost	$C_{W,k} = 3$	patient waiting: important
	$C_{W,k} = 1$	patient waiting: less important
recoverable idling cost	$C_R^n = 0$	recoverable idling costs 0
	$C_R^n = C_I^n$	recoverable idling is equivalent to regular idling
leisure waiting cost	$C_{L,k} = 0$	leisure waiting costs 0
	$C_{L,k} = C_{W,k}$	leisure waiting is equivalent to regular waiting

schedule loosens and starts earlier. The reason is that the bottleneck stage has longer service times, which drive the schedule to be less compact; at the same time, the provider at the second stage should start to work earlier to avoid excessive overtime. Second, when patient waiting cost becomes relatively more important than other costs (i.e., $C_{W,k}$ increases from 1 to 3), the optimal schedule loosens, i.e, patients tend to be spaced out. This pattern is consistent with what is observed in the classic single-stage scheduling models. That is, one should reserve longer appointment intervals when patients are more sensitive to waiting.

Furthermore, we observe that when *leisure* waiting becomes more costly (i.e., $C_{L,k}$ increases from 0 to $C_{W,k}$), the schedule tends to be tighter. Figure 4a shows one example by comparing two optimal schedules with different $C_{L,k}$'s while other parameters are kept the same: $(\bar{\delta}_k^1, \bar{\delta}_k^2) = (2,\ 1.5)$, $C_{W,k} = 3$, $C_R^n = 0$, and $C_I^n = 10$. For each stage, the optimal appointment times of patients A, B, C, D and E are illustrated by blocks A through E, respectively. As leisure waiting carries more weights, provider idling and overtime become less important, and thus one may expect the optimal schedule to stretch out. However, we observe the opposite. The explanation lies in the interdependencies between different stages. To reduce leisure waiting, one should set appointment times earlier in the second stage, leading to compact schedules in both stages.



(a) Optimal schedules w. different leisure waiting costs (b) Optimal schedules w. different waiting/idling costs

Figure 4 Optimal Schedules under Different Cost Parameters

Finally, we note that the optimal schedules with two distinct types of patient waiting costs or two distinct types of provider idling costs (i.e., $C_{L,k} \neq C_{W,k}$ or $C_R^n \neq C_I^n$) share similar structures, whereas the optimal schedule under only one type of patient waiting *and* one type of provider idling (i.e., $C_{L,k} = C_{W,k}$ *and* $C^n_R = C^n_I$) is quite different. For example, Figure 4b shows the optimal schedules assuming $(\bar{\delta}^1_k, \bar{\delta}^2_k) = (2, 1.5)$ but under different patient waiting and provider idling cost rates. Apparently, the optimal schedules of the top three scenarios are similar to each other, but the bottom one is quite different. This suggests that either the difference in two types of patient waiting or the difference in two types of provider idling has a significant impact on the optimal schedule. Simultaneously overlooking both differences can lead to inefficient management of tandem service systems. Recognizing at least one of them may generate a schedule closer to the optimal one.

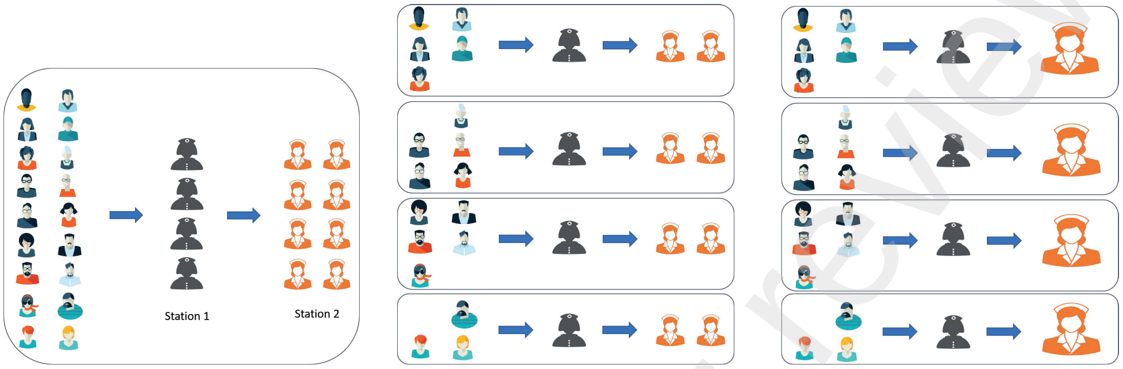
In sum, the optimal patient visit itinerary in a tandem service system is not simple. It exhibits a much more complex structure than what is observed in the single-stage scheduling context, where a "dome-shaped" schedule is often optimal and the impact of cost parameters is more straightforward (Hassin and Mendel 2008). The key drivers to design patient visit itineraries in tandem service systems include the tradeoffs among different costs and interdependencies across different stages.

6.3. Case Study in an Infusion Center

There can be multiple providers in a station of a healthcare tandem service system. Exact optimization of patient schedules in such systems, however, is very challenging, if not impossible, because patient service orders are no longer preserved. That is, patients with later appointment times in the previous station may get service earlier in the next station due to randomness in service times. Here we propose a simple heuristic leveraging our analysis of one-provider-per-station tandem systems above to come up with patient schedules in multi-server settings. Then we will test its performance by comparing it against scheduling approaches currently used in practice in a case study.

Our heuristic is inspired by the idea of approximating multi-server queues via multiple single-server queues (Arjas and Lehtonen 1978). We first identify the bottleneck of the original system. The bottleneck is the station with the highest traffic intensity, i.e., workload. Suppose that the bottleneck station has n_B providers. We then thin the demand process, i.e., the number of patients to schedule, appropriately and decompose the original tandem system into n_B simple tandem systems, where each station has only one provider. For non-bottleneck stations, we adjust service times accordingly so the workload is preserved. We derive the optimal schedule for each of these n_B systems. Finally, we superimpose these schedules to create an schedule for the original system.

The heuristic may be best explained by an example. Suppose there are 18 patients to schedule. The first station which is the bottleneck has 4 servers, and the second station has 8 servers (see Figure 5a). Then we will construct $n_B = 4$ simple tandem systems, where the first three systems have 5 patients to schedule and the last one has 3 patients (see Figure 5b). In the second station of the original system, the number of servers is 8, so there will be 2 servers in the second station for each simple system. We will use 1 super-server to represent these 2 servers (see Figure 5c). To do that, we divide the service time needed in the second station by 2 so the workload is unchanged for this station. The optimal schedule for each of these 4 simple tandem systems can be obtained using various approaches discussed above. Superimposing these schedules leads to an schedule that can be used for the original system.



(a) A Multi-server Tandem System (b) 4 Simple Tandem Systems (c) Super-server Approximation

Figure 5 Multiple Single-server Queues Approximation

We next test the heuristic in the setting of an infusion center, populated by data from the Dana-Farber Cancer Institute (Mandelbaum 2022, Mandelbaum et al. 2020). In this setting, there are two main service stations patients have to go through during their visits—examination rooms to see the physician and infusion beds to receive treatment. Typically, the daily number of patients to schedule is 90 and these patients are served by 25 infusion beds. The number of exam rooms used varies from day to day because they are shared by multiple departments. However, the ratio between the number of exam rooms and infusion beds is about 1:1.5 in Dana-Farber. So in our experiments we set the number of exam rooms to be 15. We set a time slot to be 15 minutes because in practice, the time interval between two consecutive appointments is 15 minutes, 30 minutes, or 60 minutes. The regular working time is 8-hour, i.e., 32 slots. For the service time distribution in exam rooms, we adopt the empirical distribution, which we recapitulate from Mandelbaum (2022) and uses 15-minute time slot as the unit; see Figure A.2 in Appendix A. In the second station, i.e., infusion beds, we follow Mandelbaum et al. (2020) and use the Gamma distribution to model service times, with mean 9 slots (2.25 hours) and standard deviation 5.58 slots (1.4 hours). In our model, time is discretized into slots. We round the service time generated by the Gamma distribution into multiples of 15-minute time slots. In our experiments, we adopt the same cost parameters as in Section 6.1.2 (see Table 1). Note that the number of stations and regular working times are fixed here, so in total we consider 16 different sets of cost parameters in the case study.

We compare our heuristic with the approach adopted by the infusion center. While we do not have exact schedules being used in practice, we learned their generic scheduling rules via personal communications. To come up with patient visit itineraries, the scheduler treats the service times in each station as deterministic and only use their mean values. Then, the schedule is determined for the bottleneck station assuming that it is the only service station in the system. That is, patient schedule is obtained by solving a single-stage appointment scheduling model for the bottleneck station with multiple servers and deterministic service times (see Zacharias and Pinedo 2017 for a general approach to solve such models). Then, the appointment times in other stations are adjusted by their corresponding mean service times. For example, if a patient's appointment time for infusion service is 11:15am, then his appointment time to see the physician is set at 9:00am, i.e., 2.25 hours before 11:15am. To implement our multi-server heuristic, in the experiments we use the approximation solution approach developed in Section 5 to derive the schedule for each resulting simple tandem system. This represents an easy way for practice to implement this heuristic. Indeed, this heuristic approach is computationally efficient, and takes less than 0.5 second to generate the patient visit itinerary for each instance.

We compare our multi-server heuristic with the approach used in practice by simulating their costs in the original multi-server setting. For each scheduling approach, we follow the first-come, first-served order to simulate patient service processes. We record the regular waiting and leisure waiting for each patient, as well as the regular idling, recoverable idling and overtime for each provider. We then evaluate and compare the average total costs for all sample scenarios simulated. Table 4 shows the detailed comparison results. Similar to Section 6.1.2, we focus on the exceptional and poor service environments here due to their practicality. In particular, the poor service environment with $C_L = C_W$ constitutes an operational philosophy where the aim is to not let patients wait at all, whereas the exceptional environment with $C_L = 0$ corresponds to the narrower aim to not have patients wait beyond the quoted appointment times. (Results for the mediocre and superior service environments exhibit similar patterns, and are presented in Appendix C.)

Table 4 **Improvement of Multi-server Heuristic over Approach in Practice**

Patient Waiting	Practice Environment	Service Environment	% Cost Reduction
important	mediocre	poor	34.12%
important	mediocre	exceptional	33.07%
important	superior	poor	19.14%
important	superior	exceptional	17.83%
less important	mediocre	poor	39.37%
less important	mediocre	exceptional	40.21%
less important	superior	poor	22.46%
less important	superior	exceptional	23.04%

Across these eight different cost parameter settings in Table 4, our multi-server heuristic makes a remarkable 28% cost reduction over practice on average. One set of scenarios where the approach in practice performs the closest to our heuristic is when patient waiting is important, the practice environment is superior and the service environment is exceptional. To explain this, we note that the approach in practice ignores interdependencies across stations and assumes deterministic service times. Therefore, it generates the same schedule across all cost parameter settings. Because the approach in practice builds the schedule based on the bottleneck stage (i.e., infusion service) which has a longer service time than the other stage, the schedule generated is relatively loose, meaning that patients are scheduled apart from each other. In such a loose schedule, the first stage service completion time of a patient tends to be earlier than his appointment time for service in the next stage, resulting in less regular waiting, less regular idling, more leisure waiting and more recoverable idling. When patient waiting is important, the practice environment is superior, and the service environment is exceptional, the schedule generated by the approach in practice is closer to one generated by our multi-server heuristic, because the latter schedule also tends to be loose in such a setting. Hence, the approach in practice is likely to achieve a reasonable performance. We should emphasize that even in such a setting, the multi-server heuristic still makes 18% improvement, highlighting that fully addressing uncertainties and interdependencies among stages is critical for managing tandem service systems.

7. Model Extension

Our model is motivated by specialized care environments such as infusion and orthopedic care. Appointments for such sophisticated care visits usually present high values to patients and also require efforts to arrange. Patients, therefore, respect these appointments and are likely to show up on time. However, given that no-shows and non-punctuality are commonly observed in outpatient care, it is valuable to extend our model to incorporate these uncertainties. This is the topic of this section. As it turns out, all the key technical results obtained above carry over with these important extensions.

Let binary random variable σ_k^n represent whether patient k shows up at station n (1 means show-up and 0 otherwise). We require that for each sample path, $\sigma_k^n \leq \sigma_k^{n-1}$, $\forall k \geq 1,\ n \geq 2$. This requirement means that patients will not receive service in subsequent stations unless they show up in all preceding stations. Let random variable ϵ_k denote the non-punctuality of patient k at the first station. If $\epsilon_k < 0$, it means that patient k arrives earlier than his appointment time by $|\epsilon_k|$ slots; otherwise, patient k is late by $|\epsilon_k|$ slots. Then we obtain the following recursive equations for patient ready times $\boldsymbol{r}$, provider available times $\boldsymbol{v}$, actual service start times $\boldsymbol{s}$, and service finish times $\boldsymbol{f}$, respectively.

$$r_k^n = \begin{cases} a_k^1 + \epsilon_k & \text{if } n = 1, \\ \max\{a_k^n, s_k^{n-1} + \sigma_k^{n-1}\delta_k^{n-1}\} & \text{if } n \geq 2, \end{cases} \tag{31}$$

$$v_k^n = \begin{cases} a_1^n & \text{if } k = 1, \\ \max\{a_k^n, s_{k-1}^n + \sigma_{k-1}^n\delta_{k-1}^n\} & \text{if } k \geq 2, \end{cases} \tag{32}$$

$$s_k^n = \max\{r_k^n, v_k^n\}, \tag{33}$$

and

$$f^n = \max\{s_K^n + \sigma_K^n \delta_K^n, T^n\}. \tag{34}$$

Note that (31), (32), (33) and (34) inherit a similar structure from (1), (2), (3), and (8), respectively, except that patient service time is counted as zero if he is a no-show and the ready time at the first station is perturbed by his non-punctuality accordingly. For regular wait time $\mathit{\Gamma}_{W,k}$ and leisure wait time $\mathit{\Gamma}_{L,k}$, we have, respectively,

$$\mathit{\Gamma}_{W,k} = \sum_{n=1}^{N} \sigma_k^n (s_k^n - r_k^n) \tag{35}$$

and

$$\mathit{\Gamma}_{L,k} = \sum_{n=2}^{N} \sigma_k^n (r_k^n - s_k^{n-1} - \sigma_k^{n-1} \delta_k^{n-1}). \tag{36}$$

The expression for regular idle time $\mathit{\Gamma}_I^n$ remains the same as (6), whereas the recoverable idle time $\mathit{\Gamma}_R^n$ changes to

$$\mathit{\Gamma}_R^n = \sum_{k=2}^{K} (v_k^n - s_{k-1}^n - \sigma_{k-1}^n \delta_{k-1}^n). \tag{37}$$

The expression for overtime $\mathit{\Gamma}_O^n$ remains the same as (9).

Remark 4. *When patient no-shows and non-punctuality are incorporated into the model, the same set of discrete convexity results as shown in Proposition 1 still hold.*

Remark 4 suggests that, as expected, the model with patient no-shows and non-punctuality remains challenging and the traditional solution approaches do not work. A convex relaxation is not exact. Without the submodularity of the objective function, local search is not guaranteed to reach optimum. However, our proposed solution approaches continue to be effective. One may reformulate the problem as an MILP (see Section 4.1). Or, one may reformulate the problem as a concave minimization problem over a polyhedron as done in Section 4.2—and even better—our proposed Cone Split and Cut Algorithm still works. The stage-staggering policy discussed in Section 5 can also be applied here with the same performance bound. To avoid repetitions, we leave these technical details to Appendix D.

It is important to note that this extension can handle fairly general patient no-show behavior and non-punctuality. Specifically, it allows patient no-show probabilities to depend on both patient and stage; the non-punctuality behavior can be patient-specific. In addition, it allows patient no-show behavior and non-punctuality to be correlated. For instance, this model extension can handle the situation where patients late by more than 3 slots are regarded as no-shows.

Leveraging this extension, our model can deal with settings where a subset of the patients leave the facility after visiting the first few stations. For instance, in an orthopedic clinic, patients first check in with the front desk and then see the physician. After that, depending on the need for cast adjustment or removal, patients may or may not see a cast technician. To use our model, one just need to set the no-show probability of those patients who do not need to see a cast technician to be one during that stage. This application is important, as it captures certain situations where some patients only need partial service. Or in other words, patients have non-identical service sequences. We note that, however, this extension does not apply to more complex settings where patients "skip" some stages in the middle and come back at later stages or patients visit service stations in a different order. If these happen, patient service order will not be preserved, and we need to redefine leisure and regular wait time for patients, resulting in a different model which requires new solution approaches. We do not focus on such settings here and leave them for future research.

8. Concluding Remarks

In this paper, we develop the first analytical model to design personalized visit itineraries for customers in a tandem (healthcare) service system. The itinerary specifies, for every customer, his appointment time at each stage of the service. Our flexible modeling framework can incorporate heterogeneous cost coefficients, random service times, patient no-show behavior and non-punctuality. However, due to interdependencies among stages, our model loses those elegant properties (e.g., L-convexity and submodularity) often observed in classic single-stage models, and neither does it yield an equivalent convex relaxation. To tackle these challenges we develop two original reformulations. One is directly amenable to off-the-shelf optimization packages, and the other has neat structural properties, enabling efficient solution algorithms. In addition to these exact solution approaches, we propose an approximation approach, which leads to an easy-to-implement stage-staggering policy for practical use. This policy has a provable performance guarantee, and is numerically validated to have robust performance. Nevertheless, exact solution approaches could still offer considerable improvement, and should be considered by practice when feasible.

Our study highlights the importance of carefully addressing stage interdependencies when managing patient appointment visits in tandem service systems. Current practices often rely on simple adjustments to schedules derived from single-stage models (assuming deterministic service times), and may lead to significant loss of operational efficiency. A carefully designed visit itinerary, considering different types of patient waiting and provider idling, can enhance customer experience and provider utilization.

Finally, our model is not without limitations, and there are several directions to extend our work. We assume waiting cost to be linear in time to capture the first-order difference between two types of waiting. This assumption appears to be reasonable in our setting where waiting is not too long (20-60 minutes as reported in Mandelbaum et al. 2020). However, in situations with extremely long waiting (e.g., multi-hour layover between flights), customers would be equally frustrated even if they are told in advance and hence the benefit of leisure waiting diminishes. To incorporate this effect and make the model more general, one may consider nonlinear waiting costs. In addition, future work could address a general service network where patients may receive an arbitrary subset of services in different orders, and delve deeper into multi-server settings. These extensions will result in problems with different structures which require new modeling and solution approaches. We shall leave them for future study.

Acknowledgments

This paper, initially presented at the 2022 MSOM Healthcare Operations Management SIG Day and later invited for a fast-track submission to the MSOM Journal, has greatly benefited from the support and feedback of many individuals. The authors are grateful to Editor-in-Chief Georgia Perakis for her pioneering fast-track initiative, SIG Day co-chairs Jonas Oddur Jonasson and Pengyi Shi, the participants, Department Editor Anita Carson, the associate editor, and three referees for their constructive feedback. The work of the second and last authors was supported in part by the National Science Foundation of China Grants 71931008 and 72001220, respectively.

References

Ahmadi-Javid, Amir, Zahra Jalali, Kenneth J Klassen. 2017. Outpatient appointment systems in healthcare: A review of optimization studies. *European Journal of Operational Research* **258**(1) 3–34.

Alvarez-Oh, Hyun-Jung, Hari Balasubramanian, Ekin Koker, Ana Muriel. 2018. Stochastic appointment scheduling in a team primary care practice with two flexible nurses and two dedicated providers. *Service Science* **10**(3) 241–260.

Arjas, Elja, Tapani Lehtonen. 1978. Approximating many server queues by means of single server queues. *Mathematics of Operations Research* **3**(3) 205–223.

Bailey, Norman TJ. 1952. A study of queues and appointment systems in hospital out-patient departments, with special reference to waiting-times. *Journal of the Royal Statistical Society: Series B (Methodological)* **14**(2) 185–199.

Baron, Opher, Oded Berman, Dmitry Krass, Jianfu Wang. 2014. Using strategic idleness to improve customer service experience in service networks. *Operations Research* **62**(1) 123–140.

Baron, Opher, Oded Berman, Dmitry Krass, Jianfu Wang. 2017. Strategic idleness and dynamic scheduling in an open-shop service network: Case study and analysis. *Manufacturing & Service Operations Management* **19**(1) 52–71.

Becker, Gary S. 1965. A theory of the allocation of time. *The Economic Journal* **75**(299) 493–517.

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

Cesario, Frank J. 1976. Value of time in recreation benefit studies. *Land Economics* **52**(1) 32–41.

Chen, Rachel R, Lawrence W Robinson. 2014. Sequencing and scheduling appointments with potential call-in patients. *Production and Operations Management* **23**(9) 1522–1538.

Dai, Jim G, Pengyi Shi. 2019. Inpatient overflow: An approximate dynamic programming approach. *Manufacturing & Service Operations Management* **21**(4) 894–911.

Denton, Brian, Diwakar Gupta. 2003. A sequential bounding approach for optimal appointment scheduling. *IIE Transactions* **35**(11) 1003–1016.

Diamant, Adam, Joseph Milner, Fayez Quereshy. 2018. Dynamic patient scheduling for multi-appointment health care programs. *Production and Operations Management* **27**(1) 58–79.

Fujishige, Satoru, Kazuo Murota. 1997. *On the relationship between L-convex functions and submodular integrally convex functions*. Kyoto University. Research Institute for Mathematical Sciences [RIMS].

Gupta, Diwakar, Brian Denton. 2008. Appointment scheduling in health care: Challenges and opportunities. *IIE Transactions* **40**(9) 800–819.

Hajek, Bruce. 1985. Extremal splittings of point processes. *Mathematics of Operations Research* **10**(4) 543–556.

Hassin, Refael, Sharon Mendel. 2008. Scheduling arrivals to queues: A single-server model with no-shows. *Management Science* **54**(3) 565–572.

Hathaway, Brett A, Seyed M Emadi, Vinayak Deshpande. 2021. Don't call us, we'll call you: An empirical study of caller behavior under a callback option. *Management Science* **67**(3) 1508–1526.

Helm, Jonathan E, Mark P Van Oyen. 2014. Design and optimization methods for elective hospital admissions. *Operations Research* **62**(6) 1265–1282.

Jiang, Ruiwei, Siqian Shen, Yiling Zhang. 2017. Integer programming approaches for appointment scheduling with random no-shows and service durations. *Operations Research* **65**(6) 1638–1656.

Kong, Qingxia, Shan Li, Nan Liu, Chung-Piaw Teo, Zhenzhen Yan. 2020. Appointment scheduling under time-dependent patient no-show behavior. *Management Science* **66**(8) 3480–3500.

Kuiper, Alex, Michel Mandjes. 2015. Appointment scheduling in tandem-type service systems. *Omega* **57** 145–156.

Larson, Richard C. 1987. Or forum—perspectives on queues: Social justice and the psychology of queueing. *Operations Research* **35**(6) 895–905.

Lieberman, Gerald J, Frederick S Hillier. 2005. *Introduction to operations research, 8th edition*. McGraw-Hill New York, NY, USA.

Liu, Nan, Stacey R Finkelstein, Margaret E Kruk, David Rosenthal. 2018. When waiting to see a doctor is less irritating: Understanding patient preferences and choice behavior in appointment scheduling. *Management Science* **64**(5) 1975–1996.

Liu, Nan, Van-Anh Truong, Xinshang Wang, Brett R Anderson. 2019. Integrated scheduling and capacity planning with considerations for patients' length-of-stays. *Production and Operations Management* **28**(7) 1735–1756.

Maister, David H, et al. 1984. *The psychology of waiting lines*. Harvard Business School.

Mandelbaum, Avishai. 2022. (How) Will RTLS transform healthcare delivery (research)? – Strengths & Limitations. `https://ie.technion.ac.il/serveng/References/0_0_Healthcare_INFORMS_Boston_300719_full_version_1080H.pdf`. [Online; accessed 4-January-2023].

Mandelbaum, Avishai, Petar Momčilović, Nikolaos Trichakis, Sarah Kadish, Ryan Leib, Craig A Bunnell. 2020. Data-driven appointment-scheduling under uncertainty: The case of an infusion unit in a cancer center. *Management Science* **66**(1) 243–270.

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

Robinson, Lawrence W, Rachel R Chen. 2010. A comparison of traditional and open-access policies for appointment scheduling. *Manufacturing & Service Operations Management* **12**(2) 330–346.

Soltani, Mohamad, Michele Samorani, Bora Kolfal. 2019. Appointment scheduling with multiple providers and stochastic service times. *European Journal of Operational Research* **277**(2) 667–683.

Tuy, Hoang. 2016. *Convex analysis and global optimization*. Springer International Publishing AG Switzerland.

Wang, Dongyang, Douglas J Morrice, Kumar Muthuraman, Jonathan F Bard, Luci K Leykum, Susan H Noorily. 2018. Coordinated scheduling for a multi-server network in outpatient pre-operative care. *Production and Operations Management* **27**(3) 458–479.

Wang, Dongyang, Kumar Muthuraman, Douglas Morrice. 2019. Coordinated patient appointment scheduling for a multistation healthcare network. *Operations Research* **67**(3) 599–618.

Wang, Shan, Nan Liu, Guohua Wan. 2020. Managing appointment-based services in the presence of walk-in customers. *Management Science* **66**(2) 667–686.

Zacharias, Christos, Michael Pinedo. 2014. Appointment scheduling with no-shows and overbooking. *Production and Operations Management* **23**(5) 788–801.

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

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

Zhang, Bo, Hayriye Ayhan. 2012. Optimal admission control for tandem queues with loss. *IEEE Transactions on Automatic Control* **58**(1) 163–167.

Zhang, Pengfei, Jonathan F Bard, Douglas J Morrice, Karl M Koenig. 2019. Extended open shop scheduling with resource constraints: Appointment scheduling for integrated practice units. *IISE Transactions* **51**(10) 1037–1060.

Online Appendix for "Design of Patient Visit Itineraries in Tandem Systems" by Nan Liu, Guohua Wan and Shan Wang

Appendix A: Additional Figures

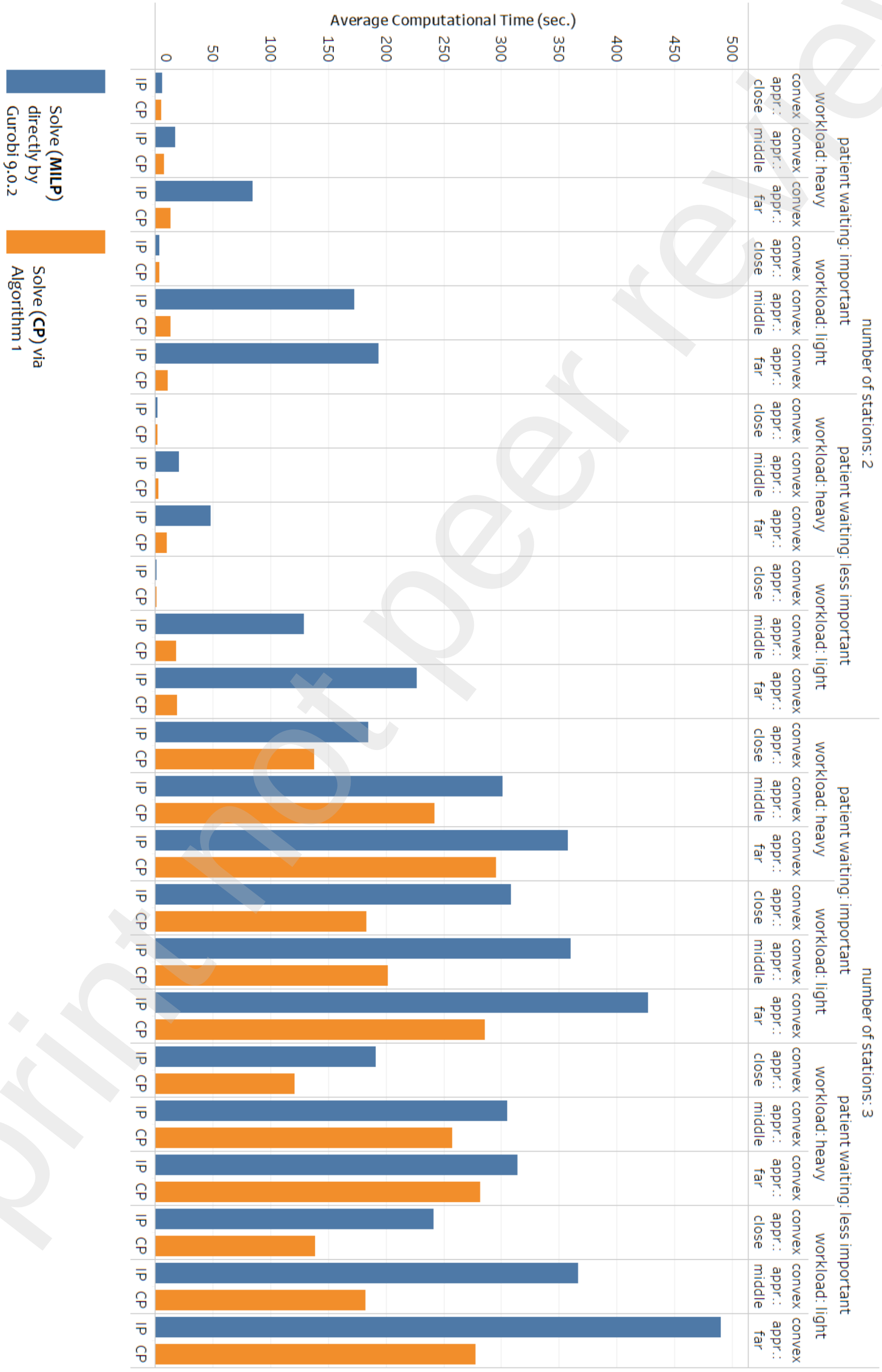


Figure A.1 Average Computational Times under Different Scenarios

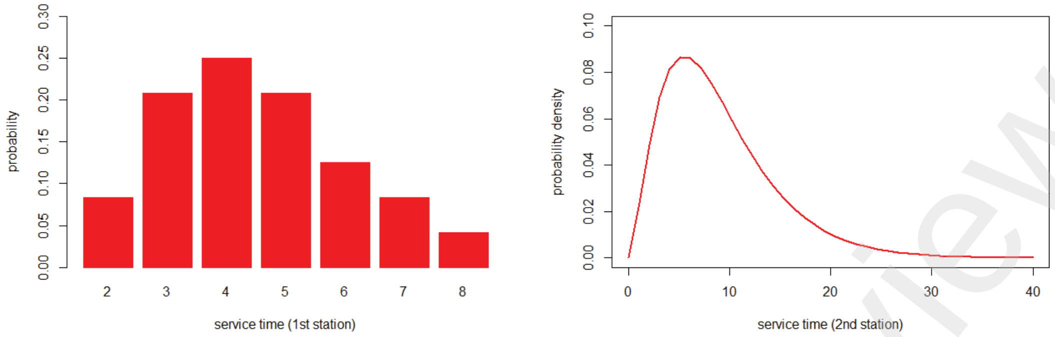


Figure A.2 Service Time Distributions

Appendix B: Cone Split and Cut Algorithm

Algorithm 1 Cone Split and Cut Algorithm (CSCA)

```
 1: Denote the original constraints in (CP) as $\boldsymbol{A}^o\boldsymbol{x} \geq \boldsymbol{b}_1^o$.
 2: Initialize a vertex $\boldsymbol{x}^0 = (\boldsymbol{a}^0, \boldsymbol{s}^0, \boldsymbol{f}^0)$ of $\mathcal{D}$, solve (Sub-LP) to obtain $\boldsymbol{s}'$ and $\boldsymbol{f}'$ for given $\boldsymbol{a}^0$, and let $U = \Gamma(\boldsymbol{a}^0, \boldsymbol{s}', \boldsymbol{f}')$
    denote the current upper bound.
 3: Calculate the cone $\mathcal{M}^0 = \{\boldsymbol{x}|\boldsymbol{x} = \boldsymbol{x}^0 + \boldsymbol{U}\boldsymbol{t}, \boldsymbol{t} \geq \boldsymbol{0}\}$ by Lemma 4. Set $\mathcal{P} = \{\mathcal{M}^0\}$.
 4: for all cones in $\mathcal{P}$ do
 5:   Denote the current cone as $\mathcal{M}$.
 6:   Denote the corresponding cuts as $\boldsymbol{A}_{cut}$ and $\boldsymbol{b}_{cut}$. Set $\boldsymbol{A} \leftarrow \begin{bmatrix} \boldsymbol{A}^o \\ \boldsymbol{A}_{cut} \end{bmatrix}$ and $\boldsymbol{b}_1 \leftarrow \begin{bmatrix} \boldsymbol{b}_1^o \\ \boldsymbol{b}_{cut} \end{bmatrix}$.
 7:   For each edge of $\mathcal{M}$, compute $\theta_i$ such that $\boldsymbol{x}^0 + \theta_i \boldsymbol{u}^i \in \partial\mathcal{C}(U - \epsilon)$.
 8:   Solve LP (LP-$\mathcal{DM}$) for $\mathcal{M}$, get the optimal solution $\boldsymbol{t}^{\mathcal{M}}$ and optimal objective value $\psi^{\mathcal{M}}$, and let $\boldsymbol{x}^{\mathcal{M}} =$
      $\boldsymbol{x}^0 + \boldsymbol{U}\boldsymbol{t}^{\mathcal{M}}$. Denote the first component in $\boldsymbol{x}^{\mathcal{M}}$ as $\boldsymbol{a}^{\mathcal{M}}$, solve (Sub-LP) to obtain $\boldsymbol{s}'$ and $\boldsymbol{f}'$ for given $\boldsymbol{a}^{\mathcal{M}}$, and
      set $\Gamma^{\mathcal{M}} = \Gamma(\boldsymbol{a}^{\mathcal{M}}, \boldsymbol{s}', \boldsymbol{f}')$.
 9:   if $\psi^{\mathcal{M}} \leq 1$ then
10:     Delete $\mathcal{M}$ from $\mathcal{P}$;
11:     Continue to next cone.
12:   else
13:     Set $\boldsymbol{A}_{cut} \leftarrow \begin{bmatrix} \boldsymbol{A}_{cut} \\ \boldsymbol{\theta}^{-1}\boldsymbol{U}^{-1} \end{bmatrix}$ and $\boldsymbol{b}_{cut} \leftarrow \begin{bmatrix} \boldsymbol{b}_{cut} \\ \boldsymbol{\theta}^{-1}\boldsymbol{U}^{-1}\boldsymbol{x}^0 + 1 \end{bmatrix}$.
14:     if $\Gamma^{\mathcal{M}} < U$ then
15:       Set $\boldsymbol{x}^{\boldsymbol{0}} = \boldsymbol{x}^{\mathcal{M}}$, $\boldsymbol{a}^0 = \boldsymbol{a}^{\mathcal{M}}$ and $U = \Gamma^{\mathcal{M}}$; clear cuts.
16:       Break and go to Step 3.
17:     end if
18:   end if
19: end for
20: if $\mathcal{P} = \emptyset$ then
21:   Terminate and return $\boldsymbol{a}^0$
22: else
23:   Select a cone $\mathcal{M}^*$ from $\mathcal{P}$ such that $\psi^{\mathcal{M}^*}$ is the maximum one among all cones in $\mathcal{P}$, and delete $\mathcal{M}^*$ from $\mathcal{P}$;
      split $\mathcal{M}^*$ to $|\boldsymbol{x}|$ sub-cones, and get a partition $\mathcal{P}^{\mathcal{M}^*}$ by Lemma 7.
24:   Let $\boldsymbol{A}_{cut}$ and $\boldsymbol{b}_{cut}$ be the corresponding cuts for cones in $\mathcal{P}^{\mathcal{M}^*}$. Set $\mathcal{P} \leftarrow \{\mathcal{P}^{\mathcal{M}^*} \cup \mathcal{P}\}$.
25:   Go to Step 4
26: end if
```

Notes: In Step 15, we set $\boldsymbol{x}^{\boldsymbol{0}} = \boldsymbol{x}^{\mathcal{M}}$, where $\boldsymbol{x}^{\mathcal{M}} = \boldsymbol{x}^0 + \boldsymbol{U}\boldsymbol{t}^{\mathcal{M}}$. Note that by solving (**LP-**$\mathcal{DM}$) we can only get the simplex tableau of $\boldsymbol{t}^{\mathcal{M}}$. In order to get the cone at $\boldsymbol{x}^{\mathcal{M}}$, we need to transform the associated simplex tableau of $\boldsymbol{t}^{\mathcal{M}}$ to the associated simplex tableau of $\boldsymbol{x}^{\mathcal{M}}$. To do that, we let $\boldsymbol{W}$ denote the associated simplex tableau of $\boldsymbol{t}^{\mathcal{M}}$. Then we use Lemma 4 to get the $\boldsymbol{U}^{\boldsymbol{t}^{\mathcal{M}}}$ corresponding to $\boldsymbol{t}^{\mathcal{M}}$ based on $\boldsymbol{W}$. Finally, the direction matrix $\boldsymbol{U}^{\boldsymbol{x}^{\mathcal{M}}} = \boldsymbol{U}\boldsymbol{U}^{\boldsymbol{t}^{\mathcal{M}}}$, where $\boldsymbol{U}$ is the direction matrix at the old $\boldsymbol{x}^0$.

Appendix C: Additional Tables and Results

Table C.1 Parameter Settings to Test CSCA

	Parameters	Implications
number of stations	$N = 2$	problem size: smaller
	$N = 3$	problem size: larger
regular working hours	$(T^1, T^2) = (10, 12)$ or $(T^1, T^2, T^3) = (13, 15, 18)$	workload: heavy
	$(T^1, T^2) = (12, 14)$ or $(T^1, T^2, T^3) = (15, 18, 20)$	workload: light
average regular waiting	$\overline{C}_{W,k} = 3$	waiting: important
	$\overline{C}_{W,k} = 1$	waiting: less important
difference between two types of idling/waiting costs	$\overline{C}_R^n = 0.25\overline{C}_I^n$ and $\overline{C}_{L,k} = 0.25\overline{C}_{W,k}$	far from convex approximation
	$\overline{C}_R^n = 0.5\overline{C}_I^n$ and $\overline{C}_{L,k} = 0.5\overline{C}_{W,k}$	$\updownarrow$
	$\overline{C}_R^n = 0.75\overline{C}_I^n$ and $\overline{C}_{L,k} = 0.75\overline{C}_{W,k}$	close to convex approximation

Notes. We use over-bar to represent the mean value of cost parameters. For instance, $\overline{C}_{W,k}$ represents the mean of unit regular waiting cost for patient k.

Appendix D: Additional Technical Results for Model Extension

D.1. Mixed Binary Integer Linear Programming

The mixed integer linear program reformulation for the extended model is similar to the one of the base model. With the extension, we just need to set the first-stage patient ready time in each scenario to be $a_k^1 + \epsilon_k(\omega)$, and modify the service time as $\sigma_k^n(\omega)\delta_k^n(\omega)$. For patient waiting, we include a factor of $\sigma_k^n(\omega)$ because we do not count waiting for no-shows in (35) and (36). Below is the formulation.

$$\min \ \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \Big[\sum_{k=1}^{K} \big(C_{W,k} \mathit{\Gamma}_{W,k}(\omega) + C_{L,k} \mathit{\Gamma}_{L,k}(\omega) \big) + \sum_{n=1}^{N} \big(C_I^n \mathit{\Gamma}_I^n(\omega) + C_R^n \mathit{\Gamma}_R^n(\omega) + C_O^n \mathit{\Gamma}_O^n(\omega) \big) \Big] \qquad \textbf{(MILP.EXT)}$$

$$\begin{aligned}
\text{s.t. } & (11), (12), (13), (15), (17), (18), (20), (22), \\
& r_k^1(\omega) = a_k^1 + \epsilon_k(\omega), \quad \forall\ k,\ \omega, \\
& r_k^n(\omega) \geq s_k^{n-1}(\omega) + \sigma_k^{n-1}(\omega)\delta_k^{n-1}(\omega), \quad \forall\ k,\ n \geq 2,\ \omega, \\
& r_k^n(\omega) \leq s_k^{n-1}(\omega) + \sigma_k^{n-1}(\omega)\delta_k^{n-1}(\omega) + M z_k^n(\omega), \quad \forall\ k,\ n \geq 2,\ \omega, \\
& v_k^n(\omega) \geq s_{k-1}^n(\omega) + \sigma_k^{n-1}(\omega)\delta_{k-1}^n(\omega), \quad \forall\ k \geq 2,\ n,\ \omega, \\
& v_k^n(\omega) \leq s_{k-1}^n(\omega) + \sigma_k^{n-1}(\omega)\delta_{k-1}^n(\omega) + M y_k^n(\omega), \quad \forall\ k \geq 2,\ n,\ \omega, \\
& f^n(\omega) \geq s_K^n(\omega) + \sigma_k^{n-1}(\omega)\delta_K^n(\omega), \quad \forall\ n,\ \omega, \\
& (24), (25), \\
& \mathit{\Gamma}_{W,k}(\omega), \mathit{\Gamma}_{L,k}(\omega), \mathit{\Gamma}_I^n(\omega), \mathit{\Gamma}_R^n(\omega) \text{ and } \mathit{\Gamma}_O^n(\omega) \text{ are defined in } (35), (36), (6), (37), (9), \text{ respectively}, \quad \forall\ \omega, \\
& a_1^1 = 0;\ 0 \leq a_k^n \leq T^n;\ s_k^n(\omega),\ r_k^n(\omega),\ v_k^n(\omega) \text{ and } f^n(\omega) \geq 0;\ z_k^n(\omega) \text{ and } y_k^n(\omega) \in \{0, 1\}, \quad \forall\ k,\ n,\ \omega.
\end{aligned}$$

D.2. Concave Minimization over Polyhedron

For the concave programming reformulation, we make similar modifications as above. The detailed formulation follows.

$$\min \ \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \Big\{ \sum_{k=1}^{K} \Big[C_{W,k} \big(\sum_{n=1}^{N} \sigma_k^n(\omega) s_k^n(\omega) - \sigma_k^1(\omega)(a_k^1 + \epsilon_k(\omega)) - \sum_{n=1}^{N-1} \sigma_k^{n+1}(\omega)(s_k^n(\omega) + \delta_k^n(\omega)) \big) - (C_{W,k} - C_{L,k})$$

Table C.2 Performance of Approximation Solution Approach when Service Environment is Superior or Mediocre

Bottleneck	Workload	Patient Waiting	Practice Environment	Service Environment	% Gap to MILP	% Gap to Practice
1	heavy	important	superior	superior	-1.31%	-11.90%
1	heavy	important	superior	mediocre	1.10%	-11.56%
1	heavy	important	mediocre	superior	4.86%	-19.72%
1	heavy	important	mediocre	mediocre	0.23%	-19.88%
1	heavy	less important	superior	superior	8.16%	-13.92%
1	heavy	less important	superior	mediocre	1.97%	-12.12%
1	heavy	less important	mediocre	superior	4.42%	-11.36%
1	heavy	less important	mediocre	mediocre	0.64%	-12.39%
1	light	important	superior	superior	-8.15%	-12.11%
1	light	important	superior	mediocre	4.74%	-14.21%
1	light	important	mediocre	superior	-22.31%	-15.87%
1	light	important	mediocre	mediocre	2.13%	-14.14%
1	light	less important	superior	superior	-20.27%	-20.65%
1	light	less important	superior	mediocre	4.46%	-17.30%
1	light	less important	mediocre	superior	10.37%	-21.79%
1	light	less important	mediocre	mediocre	2.47%	-19.57%
2	heavy	important	superior	superior	6.54%	-6.79%
2	heavy	important	superior	mediocre	-0.49%	-10.15%
2	heavy	important	mediocre	superior	2.51%	-19.66%
2	heavy	important	mediocre	mediocre	-0.66%	-20.20%
2	heavy	less important	superior	superior	0.34%	-12.53%
2	heavy	less important	superior	mediocre	1.17%	-17.48%
2	heavy	less important	mediocre	superior	-0.51%	-10.89%
2	heavy	less important	mediocre	mediocre	-0.11%	-26.04%
2	light	important	superior	superior	16.86%	-10.97%
2	light	important	superior	mediocre	12.66%	-2.11%
2	light	important	mediocre	superior	-3.47%	-15.36%
2	light	important	mediocre	mediocre	-1.82%	-16.22%
2	light	less important	superior	superior	-0.13%	8.61%
2	light	less important	superior	mediocre	-5.80%	7.45%
2	light	less important	mediocre	superior	-2.37%	-10.20%
2	light	less important	mediocre	mediocre	0.56%	-15.91%

Notes. The definition of % Gap is the same as in Table 2 in the main paper.

Table C.3 Improvement of Multi-server Heuristic over Approach in Practice when Service Environment is Superior or Mediocre

Patient Waiting	Practice Environment	Service Environment	% Cost Reduction
important	mediocre	mediocre	27.21%
important	mediocre	superior	27.29%
important	superior	mediocre	27.20%
important	superior	superior	12.98%
less important	mediocre	mediocre	38.59%
less important	mediocre	superior	39.00%
less important	superior	mediocre	21.99%
less important	superior	superior	22.28%

$$\sum_{n=2}^{N}\sigma_k^n(\omega)\max\{a_k^n - s_k^{n-1}(\omega) - \sigma_k^{n-1}(\omega)\delta_k^{n-1}(\omega),0\}\Big] + \sum_{n=1}^{N}\Big[C_I^n\big(s_K^n(\omega) - a_1^n - \sum_{k=1}^{K-1}\sigma_k^n(\omega)\delta_k^n(\omega)\big) - (C_I^n - C_R^n)$$

$$\sum_{k=2}^{K}\max\{a_k^n - s_{k-1}^n(\omega) - \sigma_{k-1}^n(\omega)\delta_{k-1}^n(\omega),0\} + C_O^n(f^n(\omega) - T^n)\Big]\Big\} \qquad \textbf{(CP.ext)}$$

$$
\begin{aligned}
\text{s.t. } & s_k^1(\omega) \geq a_k^1 + \epsilon_k(\omega), \quad \forall\, k,\ \omega, && \text{(D.1)}\\
& s_k^n(\omega) \geq a_k^n, \quad \forall\, k,\ n \geq 2,\ \omega, && \text{(D.2)}\\
& s_k^n(\omega) \geq s_{k-1}^n(\omega) + \sigma_{k-1}^n(\omega)\delta_{k-1}^n(\omega), \quad \forall\, k \geq 2,\ n,\ \omega, && \text{(D.3)}\\
& s_k^n(\omega) \geq s_k^{n-1}(\omega) + \sigma_k^{n-1}(\omega)\delta_k^{n-1}(\omega), \quad \forall\, k,\ n \geq 2,\ \omega, && \text{(D.4)}\\
& f^n(\omega) \geq s_K^n(\omega) + \sigma_K^n(\omega)\delta_K^n(\omega), \quad \forall\, n,\ \omega, && \text{(D.5)}\\
& (22), (24), (25), (29).
\end{aligned}
$$

Note that there are some changes in the objective function, because patients may become no-shows after visiting some stages. However, the objective function of (**CP.ext**) is still concave in the decision variables.

For the linear program used in the Cone Split and Cut Algorithm to generate a tighter bound for given a solution, we have the following formulation.

$$
\min_{\boldsymbol{s}(\omega)\geq 0, \boldsymbol{f}(\omega)\geq 0} \left\{ \frac{1}{|\Omega|} \sum_{\omega} \sum_{n=1}^{N} (f^n(\omega) + \sum_{k=1}^{K} s_k^n(\omega)) \Big| (\text{D.1}), (\text{D.2}), (\text{D.3}), (\text{D.4}), (22), (\text{D.5}) \right\}. \qquad \textbf{(Sub-LP.ext)}
$$

Based on (**CP.ext**) and (**Sub-LP.ext**), the Cone Split and Cut Algorithm still works for the extension.

D.3. Approximation Solution Approach

For the approximation solution, we have to revise the objective function and modify the service time as $\sigma_k^n \delta_k^n$. Below is the detailed formulation.

$$
\begin{aligned}
\min\ & \frac{1}{|\Omega|} \sum_{\omega\in\Omega} \Big\{ \sum_{k=1}^{K} \Big[C_{W,k} \big(\sum_{n=1}^{N} \sigma_k^n(\omega) s_k^n(\omega) - \sigma_k^1(\omega)(a_k^1 + \epsilon_k(\omega)) - \sum_{n=1}^{N-1} \sigma_k^{n+1}(\omega)(s_k^n(\omega) + \delta_k^n(\omega)) \big) \Big] \\
& + \sum_{n=1}^{N} \Big[C_I^n \big(s_K^n(\omega) - a_1^n - \sum_{k=1}^{K-} \sigma_k^n(\omega)\delta_k^n(\omega) \big) + C_O^n (f^n(\omega) - T^n) \Big] \Big\} \qquad \textbf{(APR.ext)}\\
\text{s.t. }\ & (\text{D.1}), (\text{D.2}), (\text{D.3}), (\text{D.4}), (22), (\text{D.5}), (24), (25), (29).
\end{aligned}
$$

Note that (**APR.ext**) is a linear program, so we still have a performance guarantee for it.

Corollary D.1. *Let $\boldsymbol{a}^*$ and $\boldsymbol{a}_R^*$ denote the optimal schedule to the model with patient no-shows and non-punctuality and that to the model with patient no-shows and non-punctuality but under the stage-staggering policy, respectively. Then, the performance loss of $\boldsymbol{a}_R^*$ is bounded from above as follows.*

$$
\Gamma(\boldsymbol{a}_R^*) - \Gamma(\boldsymbol{a}^*) \leq \max_{k,n} \{C_{W,k} - C_{L,k}, C_I^n - C_R^n\}(N + K - 2) \max_n \{T^n\}.
$$

Appendix E: Proofs of Analytical Results

Proof of Remark 1: Let us use ω to denote a sample path. By definition, $s_k^n(\omega) = \max\{a_k^n, s_{k-1}^n(\omega) + \delta_{k-1}^n(\omega), s_k^{n-1}(\omega) + \delta_k^{n-1}(\omega)\}$, $f^n(\omega) = \max\{T^n, s_K^n(\omega) + \delta_K^n(\omega)\}$, $r_k^n(\omega) = \max\{a_k^n, s_k^{n-1}(\omega) + \delta_k^{n-1}(\omega)\}$, $v_k^n(\omega) = \max\{a_k^n, s_{k-1}^n(\omega) + \delta_{k-1}^n(\omega)\}$. Expanding these terms recursively, we obtain $s_k^n(\omega) = \max\{a_k^n, a_{k-1}^n + \delta_{k-1}^n(\omega), a_k^{n-1} + \delta_k^{n-1}(\omega), a_{k-2}^n + \delta_{k-2}^n(\omega) + \delta_{k-1}^n(\omega), a_{k-1}^{n-1} + \delta_{k-1}^{n-1}(\omega) + \delta_{k-1}^n(\omega), a_{k-1}^{n-1} + \delta_{k-1}^{n-1}(\omega) + \delta_k^{n-1}(\omega), \ldots, \delta_1^1(\omega) + \cdots + \delta_{k-1}^n(\omega), \delta_1^1(\omega) + \cdots + \delta_k^{n-1}(\omega)\}$, $f^n(\omega) = \max\{T^n, a_K^n + \delta_K^n(\omega), a_{K-1}^n + \delta_{K-1}^n(\omega) + \delta_K^n(\omega), a_K^{n-1} + \delta_K^{n-1}(\omega) + \delta_K^n(\omega), \ldots, \delta_1^1(\omega) + \cdots + \delta_K^n(\omega)\}$, $r_k^n(\omega) = \max\{a_k^n, a_k^{n-1} + \delta_k^{n-1}(\omega), a_{k-1}^{n-1} + \delta_{k-1}^{n-1}(\omega) + \delta_k^{n-1}(\omega), \ldots, \delta_1^1(\omega) + \cdots + \delta_k^{n-1}(\omega)\}$, and $v_k^n(\omega) = \max\{a_k^n, a_{k-1}^n + \delta_{k-1}^n(\omega), a_{k-1}^{n-1} + \delta_{k-1}^{n-1}(\omega) + \delta_{k-1}^n(\omega), \ldots, \delta_1^1(\omega) + \cdots + \delta_{k-1}^n(\omega)\}$. Note that $\delta_k^n(\omega)$ must be integer.

Considering the above equations as constraints, the total cost

$$
\frac{1}{\Omega} \sum_{\omega\in\Omega} \left[\sum_{k=1}^{K} \left[C_{W,k} \sum_{n=1}^{N} (s_k^n(\omega) - r_k^n(\omega)) + C_{L,k} \sum_{n=2}^{N} \left(r_k^n(\omega) - s_k^{n-1}(\omega) - \delta_k^{n-1}(\omega) \right) \right] + \right.
$$

$$\sum_{n=1}^{N}\left[C_I^n\sum_{k=1}^{K}\left(s_k^n(\omega)-v_k^n(\omega)\right)+C_R^n\sum_{k=2}^{K}\left(v_k^n(\omega)-s_{k-1}^n(\omega)-\delta_{k-1}^n(\omega)\right)+C_O^n\left(f^n(\omega)-T^n\right)\right]\Bigg]$$

is a linear function of a_k^n, $s_k^n(\omega)$, $f^n(\omega)$, $r_k^n(\omega)$ and $v_k^n(\omega)$.

Suppose that a non-integer schedule $\boldsymbol{a}$ is optimal. We argue that we can obtain a schedule such that all non-integer entries in $\boldsymbol{a}$ become integers and all integer entries in $\boldsymbol{a}$ are unchanged yielding a non-higher cost.

Now we focus on some non-integer a_k^n with a fractional part δ, i.e., $a_k^{n\prime}-\lfloor a_k^n\rfloor=\delta$. We first show that there exists another schedule such that a_k^n becomes integer and all integer entries in $\boldsymbol{a}$ are unchanged yielding non-higher cost.

We first check whether there are any other entries in $\boldsymbol{a}$ having the same fractional part δ, i.e., there is a_i^j such that $a_i^j-\lfloor a_i^j\rfloor=\delta$. Denote the set of these entries (including a_k^n) as $\mathcal{A}_\delta$. Let us perturb these entries by a small number ϵ, which can be negative or positive, then $s_{k'}^{n'}(\omega)$, $f^{n'}(\omega)$, $r_{k'}^{n'}(\omega)$ and $v_{k'}^{n'}(\omega)$ may also change (we use n' and k' to differentiate from n and k). Let $\mathcal{P}_\epsilon$ denote the set of $s_{k'}^{n'}(\omega)$, $f^{n'}(\omega)$, $r_{k'}^{n'}(\omega)$ and $v_{k'}^{n'}(\omega)$ which will change with the entries in $\mathcal{A}_\delta$. For example, if $s_{k'}^{n'}(\omega)\in\mathcal{P}_\epsilon$, then $s_{k'}^{n'}(\omega|\mathcal{A}_\delta)\neq s_{k'}^{n'}(\omega|\mathcal{A}_\delta+\epsilon)$ for $\epsilon\neq 0$. (Note that $\mathcal{P}_\epsilon$ may be different for different ϵ because $s_{k'}^{n'}(\omega)$, $f^{n'}(\omega)$, $r_{k'}^{n'}(\omega)$ and $v_{k'}^{n'}(\omega)$ are also constrained by other appointment times.) Define $\underline{\epsilon}<0$ and $\bar{\epsilon}>0$ such that $\mathcal{P}_\epsilon=\lim_{\epsilon'\to 0}\mathcal{P}_{\epsilon'}$ if $\epsilon\in[\underline{\epsilon},0)\cup(0,\bar{\epsilon}]$, i.e., the set of the influenced $s_{k'}^{n'}(\omega)$, $f^{n'}(\omega)$, $r_{k'}^{n'}(\omega)$ and $v_{k'}^{n'}(\omega)$ will be the same if ϵ changes within $[\underline{\epsilon},0)\cup(0,\bar{\epsilon}]$. (Note that $\mathcal{A}_\delta$ contains all entries in $\boldsymbol{a}$ with the same fractional part, thus $\underline{\epsilon}<0$ and $\bar{\epsilon}>0$ must exist.) By the definitions of $s_k^n(\omega)$, $f^n(\omega)$, $r_k^n(\omega)$ and $v_k^n(\omega)$, we have $x(\omega|\mathcal{A}_\delta+\epsilon)=x(\omega|\mathcal{A}_\delta)+\epsilon$ if $x(\omega)\in\mathcal{P}_\epsilon$ and $\epsilon\in[\underline{\epsilon},\bar{\epsilon}]$; and $x(\omega|\mathcal{A}_\delta+\epsilon)=x(\omega|\mathcal{A}_\delta)$ if $x(\omega)\notin\mathcal{P}_\epsilon$ and $\epsilon\in[\underline{\epsilon},\bar{\epsilon}]$ (here x represents variables s_k^n, f_k^n, r_k^n, or v_k^n). Since the cost function is linear in a_k^n, $s_k^n(\omega)$, $f^n(\omega)$, $r_k^n(\omega)$ and $v_k^n(\omega)$, and $\mathcal{P}_\epsilon$ keeps same when $\epsilon\in[\underline{\epsilon},0)\cup(0,\bar{\epsilon}]$, then the cost is also linear in ϵ when $\epsilon\in[\underline{\epsilon},\bar{\epsilon}]$. So letting either $\epsilon\in\underline{\epsilon}$ or $\epsilon=\bar{\epsilon}$ must yield non-higher cost.

Now, suppose that $\epsilon=\underline{\epsilon}$ (or $\epsilon=\bar{\epsilon}$) yields non-higher cost.

If $-\underline{\epsilon}\geq\delta$ (or $\bar{\epsilon}\geq 1-\delta$), then decreasing all entries in $\mathcal{A}_\delta$ by δ (or increasing all entries in $\mathcal{A}_\delta$ by $1-\delta$) and keeping other entries in $\boldsymbol{a}$ unchanged yields a non-higher cost. Now we have a schedule with fewer non-integers yielding no higher cost.

If $-\underline{\epsilon}<\delta$ (or $\bar{\epsilon}<1-\delta$), then we decrease all entries in $\mathcal{A}_\delta$ by $-\underline{\epsilon}$ (or increase all entries in $\mathcal{A}_\delta$ by $\bar{\epsilon}$). Denote this new schedule as $\boldsymbol{a}'$. If we continue to decrease (or increase) the entries in $\mathcal{A}_\delta$, then some $s(\omega)$, $f(\omega)$, $r(\omega)$ or $v(\omega)$ which are not in $\mathcal{P}_\epsilon$ will be influenced. Because $\delta_k^n(\omega)$ must be an integer, there must exist at least one non-integer a_i^j such that $a_i^j-\lfloor a_i^j\rfloor=\delta+\underline{\epsilon}$ (or $a_i^j-\lfloor a_i^j\rfloor=\delta+\bar{\epsilon}$) by the definitions of $s_k^n(\omega)$, $f_k^n(\omega)$, $r_k^n(\omega)$ and $v_k^n(\omega)$.

Recall that the updated schedule is denoted as $\boldsymbol{a}'$. Now the fractional part of ${a'}_k^n$ becomes δ'. Denote the set of entries in $\boldsymbol{a}'$ with fractional part δ' as $\mathcal{A}_{\delta'}$, and continue to perturb the entries in $\mathcal{A}_{\delta'}$ by a small ϵ, and update $\underline{\epsilon}$, $\bar{\epsilon}$ and $\mathcal{P}_\epsilon$ accordingly. Similar to the previous arguments, we must have $x(\omega|\mathcal{A}_{\delta'}+\epsilon)=x(\omega|\mathcal{A}_{\delta'})+\epsilon$ if $x(\omega)\in\mathcal{P}_\epsilon$ and $\epsilon\in[\underline{\epsilon},\bar{\epsilon}]$; and $x(\omega|\mathcal{A}_{\delta'}+\epsilon)=x(\omega|\mathcal{A}_{\delta'})$ if $x(\omega)\notin\mathcal{P}_\epsilon$ and $\epsilon\in[\underline{\epsilon},\bar{\epsilon}]$. So letting either $\epsilon=\underline{\epsilon}$ (or $\epsilon=\bar{\epsilon}$) yields a non-higher cost. We will either decrease (or increase) the entries in $\mathcal{A}_{\delta'}$ by δ' (or $1-\delta'$) and make them become integers, or decrease (or increase) them by $-\underline{\epsilon}$ (or $\bar{\epsilon}$) and find another non-integer ${a'}_l^m$ such that ${a'}_l^m-\lfloor {a'}_l^m\rfloor=\delta'+\underline{\epsilon}$ if $\epsilon=\underline{\epsilon}$ yields non-higher cost (or ${a'}_l^m-\lfloor {a'}_l^m\rfloor=\delta'+\bar{\epsilon}$ if $\epsilon=\bar{\epsilon}$ yields a non-higher cost).

We can repeat this procedure till ${a'}_k^n$ becomes an integer. In these procedures, the cost is not increased and all integers in $\boldsymbol{a}$ are unchanged. Thus we prove that there exists another schedule such that a_k^n becomes an integer and all integers in $\boldsymbol{a}$ are unchanged yielding a non-higher cost. By induction, we can obtain an integer schedule yielding a non-higher cost. □

Proof of Proposition 1:

1. To show that the total regular waiting cost $\sum_{k=1}^{K} C_{W,k}\mathit{\Gamma}_{W,k}(\boldsymbol{a})$ is not necessarily submodular or L-convex on $\boldsymbol{a}$, it suffices to construct an example. Let us consider the following deterministic service times: $\delta_1^1 = 1$, $\delta_2^1 = 1$, $\delta_1^2 = 2$, $\delta_2^2 = 1$, and $\delta_k^n = 1$ for $k > 2$ or $n > 2$. Consider the following schedule: $a_1^1 = 0$, $a_2^1 = 1$, $a_1^2 = 1$, $a_2^2 = 2$, and $a_k^n \geq 6$ for $k > 2$ or $n > 2$. Also, consider the following perturbations: $\boldsymbol{a} + \boldsymbol{u}_i$ makes $a_2^1 = 2$; $\boldsymbol{a} + \boldsymbol{u}_j$ makes $a_2^2 = 3$; $\boldsymbol{a} + \boldsymbol{u}_i + \boldsymbol{u}_j$ makes $a_2^1 = 2$ and $a_2^2 = 3$.

Then, for $n \leq 2$ and $k \leq 2$, we obtain the follows.

With $\boldsymbol{a}$: $r_1^1 = 0$, $r_2^1 = 1$, $r_1^2 = 1$, $r_2^2 = 2$; $s_1^1 = 0$, $s_2^1 = 1$, $s_1^2 = 1$, $s_2^2 = 3$.

With $\boldsymbol{a} + \boldsymbol{u}_i$: $r_1^1 = 0$, $r_2^1 = 2$, $r_1^2 = 1$, $r_2^2 = 3$; $s_1^1 = 0$, $s_2^1 = 2$, $s_1^2 = 1$, $s_2^2 = 3$.

With $\boldsymbol{a} + \boldsymbol{u}_j$: $r_1^1 = 0$, $r_2^1 = 1$, $r_1^2 = 1$, $r_2^2 = 3$; $s_1^1 = 0$, $s_2^1 = 1$, $s_1^2 = 1$, $s_2^2 = 3$.

With $\boldsymbol{a} + \boldsymbol{u}_i + \boldsymbol{u}_j$: $r_1^1 = 0$, $r_2^1 = 2$, $r_1^2 = 1$, $r_2^2 = 3$; $s_1^1 = 0$, $s_2^1 = 2$, $s_1^2 = 1$, $s_2^2 = 3$.

For $n > 2$ or $k > 2$, since $a_k^n \geq 6$, then $r_k^n = r_k^n(\boldsymbol{u}_i) = r_k^n(\boldsymbol{u}_j) = r_k^n(\boldsymbol{u}_i + \boldsymbol{u}_j)$ and $s_k^n = s_k^n(\boldsymbol{u}_i) = s_k^n(\boldsymbol{u}_j) = s_k^n(\boldsymbol{u}_i + \boldsymbol{u}_j)$.

Recall that $\mathit{\Gamma}_{W,k} = \sum_{n=1}^{N}(s_k^n - r_k^n)$. Hence, we have $\mathit{\Gamma}_{W,2}(\boldsymbol{a} + \boldsymbol{u}_i) + \mathit{\Gamma}_{W,2}(\boldsymbol{a} + \boldsymbol{u}_j) = 0$, which is strictly smaller than $\mathit{\Gamma}_{W,2}(\boldsymbol{a}) + \mathit{\Gamma}_{W,2}(\boldsymbol{a} + \boldsymbol{u}_i + \boldsymbol{u}_j) = 1$; and for $k \neq 2$, $\mathit{\Gamma}_{W,k}(\boldsymbol{a} + \boldsymbol{u}_i) + \mathit{\Gamma}_{W,k}(\boldsymbol{a} + \boldsymbol{u}_j) = \mathit{\Gamma}_{W,k}(\boldsymbol{a}) + \mathit{\Gamma}_{W,k}(\boldsymbol{a} + \boldsymbol{u}_i + \boldsymbol{u}_j)$.

By the definitions of submodularity and L-convexity, we can conclude that, given any N, K, $\boldsymbol{T}$ and $C_{W,k} > 0$, there exists a service time distribution such that the total regular waiting cost $\sum_{k=1}^{K} C_{W,k}\mathit{\Gamma}_{W,k}(\boldsymbol{a})$ is not submodular or L-convex on $\boldsymbol{a}$.

2. To show that the total leisure waiting cost $\sum_{k=1}^{K} C_{L,k}\mathit{\Gamma}_{L,k}(\boldsymbol{a})$ is not necessarily submodular or L-convex on $\boldsymbol{a}$, it suffices to construct an example. Let us consider the same deterministic service times: $\delta_1^1 = 1$, $\delta_2^1 = 1$, $\delta_1^2 = 2$, $\delta_2^2 = 1$, and $\delta_k^n = 1$ for $k > 2$ or $n > 2$. Consider the following schedule: $a_1^1 = 0$, $a_2^1 = 1$, $a_1^2 = 1$, $a_2^2 = 3$, and $a_k^n \geq 6$ for $k > 2$ or $n > 2$. Also, consider the following perturbations: $\boldsymbol{a} + \boldsymbol{u}_i$ makes $a_2^1 = 2$; $\boldsymbol{a} + \boldsymbol{u}_j$ makes $a_1^1 = 1$; $\boldsymbol{a} + \boldsymbol{u}_i + \boldsymbol{u}_j$ makes $a_2^1 = 2$ and $a_1^1 = 1$.

Then, for $n \leq 2$ and $k \leq 2$, we obtain the follows.

With $\boldsymbol{a}$: $r_1^1 = 0$, $r_2^1 = 1$, $r_1^2 = 1$, $r_2^2 = 3$; $f_1^1 = 1$, $f_2^1 = 2$, $f_1^2 = 3$, $f_2^2 = 4$.

With $\boldsymbol{a} + \boldsymbol{u}_i$: $r_1^1 = 0$, $r_2^1 = 2$, $r_1^2 = 1$, $r_2^2 = 3$; $f_1^1 = 1$, $f_2^1 = 3$, $f_1^2 = 3$, $f_2^2 = 4$.

With $\boldsymbol{a} + \boldsymbol{u}_j$: $r_1^1 = 1$, $r_2^1 = 1$, $r_1^2 = 2$, $r_2^2 = 3$; $f_1^1 = 2$, $f_2^1 = 3$, $f_1^2 = 4$, $f_2^2 = 5$.

With $\boldsymbol{a} + \boldsymbol{u}_i + \boldsymbol{u}_j$: $r_1^1 = 1$, $r_2^1 = 2$, $r_1^2 = 2$, $r_2^2 = 3$; $f_1^1 = 2$, $f_2^1 = 3$, $f_1^2 = 4$, $f_2^2 = 5$.

For $n > 2$ or $k > 2$, since $a_k^n \geq 6$, then $r_k^n = r_k^n(\boldsymbol{u}_i) = r_k^n(\boldsymbol{u}_j) = r_k^n(\boldsymbol{u}_i + \boldsymbol{u}_j)$ and $f_k^n = f_k^n(\boldsymbol{u}_i) = f_k^n(\boldsymbol{u}_j) = f_k^n(\boldsymbol{u}_i + \boldsymbol{u}_j)$.

Recall that $\mathit{\Gamma}_{L,k} = \sum_{n=2}(r_k^n - f_k^{n-1})$, we have $\mathit{\Gamma}_{L,2}(\boldsymbol{a} + \boldsymbol{u}_i) + \mathit{\Gamma}_{L,2}(\boldsymbol{a} + \boldsymbol{u}_j) = 0$, which is strictly smaller than $\mathit{\Gamma}_{L,2}(\boldsymbol{a}) + \mathit{\Gamma}_{L,2}(\boldsymbol{a} + \boldsymbol{u}_i + \boldsymbol{u}_j) = 1$; and for $k \neq 2$, $\mathit{\Gamma}_{L,k}(\boldsymbol{a} + \boldsymbol{u}_i) + \mathit{\Gamma}_{L,k}(\boldsymbol{a} + \boldsymbol{u}_j) = \mathit{\Gamma}_{L,k}(\boldsymbol{a}) + \mathit{\Gamma}_{L,k}(\boldsymbol{a} + \boldsymbol{u}_i + \boldsymbol{u}_j)$.

By the definitions of submodularity and L-convexity, we can conclude that, given any N, K, $\boldsymbol{T}$ and $C_{L,k} > 0$, there exists a service time distribution such that the total leisure waiting cost $\sum_{k=1}^{K} C_{L,k}\mathit{\Gamma}_{L,k}(\boldsymbol{a})$ is not submodular or L-convex on $\boldsymbol{a}$.

3. To show that the total regular idling cost $\sum_{n=1}^{N} C_I^n \mathit{\Gamma}_I^n(\boldsymbol{a})$ is not necessarily submodular or L-convex on $\boldsymbol{a}$, it suffices to construct an example. Let us consider the following deterministic service times: $\delta_1^1 = 1$, $\delta_2^1 = 2$, $\delta_1^2 = 1$, $\delta_2^2 = 1$, and $\delta_k^n = 1$ for $k > 2$ or $n > 2$. Consider the following schedule: $a_1^1 = 0$, $a_2^1 = 1$, $a_1^2 = 1$, $a_2^2 = 2$, and $a_k^n \geq 6$ for $k > 2$ or $n > 2$. Also, consider the following perturbations: $\boldsymbol{a} + \boldsymbol{u}_i$ makes $a_1^2 = 2$; $\boldsymbol{a} + \boldsymbol{u}_j$ makes $a_2^2 = 3$; $\boldsymbol{a} + \boldsymbol{u}_i + \boldsymbol{u}_j$ makes $a_1^2 = 2$ and $a_2^2 = 3$.

Then, for $n \leq 2$ and $k \leq 2$, we obtain the follows.

With $\boldsymbol{a}$: $v_1^1 = 0$, $v_2^1 = 1$, $v_1^2 = 1$, $v_2^2 = 2$; $s_1^1 = 0$, $s_2^1 = 1$, $s_1^2 = 1$, $s_2^2 = 3$.

With $\boldsymbol{a} + \boldsymbol{u}_i$: $v_1^1 = 0$, $v_2^1 = 1$, $v_1^2 = 2$, $v_2^2 = 3$; $s_1^1 = 0$, $s_2^1 = 1$, $s_1^2 = 2$, $s_2^2 = 3$.

With $\boldsymbol{a} + \boldsymbol{u}_j$: $v_1^1 = 0$, $v_2^1 = 1$, $v_1^2 = 1$, $v_2^2 = 3$; $s_1^1 = 0$, $s_2^1 = 1$, $s_1^2 = 1$, $s_2^2 = 3$.

With $\boldsymbol{a}+\boldsymbol{u}_i+\boldsymbol{u}_j$: $v_1^1=0$, $v_2^1=1$, $v_1^2=2$, $v_2^2=3$; $s_1^1=0$, $s_2^1=1$, $s_1^2=2$, $s_2^2=3$.

For $n>2$ or $k>2$, since $a_k^n\geq 6$, then $v_k^n=v_k^n(\boldsymbol{u}_i)=v_k^n(\boldsymbol{u}_j)=v_k^n(\boldsymbol{u}_i+\boldsymbol{u}_j)$ and $s_k^n=s_k^n(\boldsymbol{u}_i)=s_k^n(\boldsymbol{u}_j)=s_k^n(\boldsymbol{u}_i+\boldsymbol{u}_j)$.

Recall that $\mathit{\Gamma}_I^n=\sum_{k=1}^K(s_k^n-v_k^n)$. Hence, we have $\mathit{\Gamma}_I^2(\boldsymbol{a}+\boldsymbol{u}_i)+\mathit{\Gamma}_I^2(\boldsymbol{a}+\boldsymbol{u}_j)=0$, which is strictly smaller than $\mathit{\Gamma}_I^2(\boldsymbol{a})+\mathit{\Gamma}_I^2(\boldsymbol{a}+\boldsymbol{u}_i+\boldsymbol{u}_j)=1$; and for $n\neq 2$, $\mathit{\Gamma}_I^n(\boldsymbol{a}+\boldsymbol{u}_i)+\mathit{\Gamma}_I^n(\boldsymbol{a}+\boldsymbol{u}_j)=\mathit{\Gamma}_I^n(\boldsymbol{a})+\mathit{\Gamma}_I^n(\boldsymbol{a}+\boldsymbol{u}_i+\boldsymbol{u}_j)$.

By the definitions of submodularity and L-convexity, we can conclude that, given any N, K, $\boldsymbol{T}$ and $C_I^n>0$, there exists a service time distribution such that the total regular idling cost $\sum_{n=1}^N C_I^n\mathit{\Gamma}_I^n(\boldsymbol{a})$ is not submodular or L-convex on $\boldsymbol{a}$.

4. To show that the total recoverable idling cost $\sum_{n=1}^N C_R^n\mathit{\Gamma}_R^n(\boldsymbol{a})$ is not necessarily submodular or L-convex on $\boldsymbol{a}$, it suffices to construct an example. Let us consider the same deterministic service times: $\delta_1^1=1$, $\delta_2^1=2$, $\delta_1^2=1$, $\delta_2^2=1$, and $\delta_k^n=1$ for $k>2$ or $n>2$. Consider the following schedule: $a_1^1=0$, $a_2^1=1$, $a_1^2=1$, $a_2^2=3$, and $a_k^n\geq 6$ for $k>2$ or $n>2$. Also, consider the following perturbations: $\boldsymbol{a}+\boldsymbol{u}_i$ makes $a_1^1=1$; $\boldsymbol{a}+\boldsymbol{u}_j$ makes $a_1^2=2$; $\boldsymbol{a}+\boldsymbol{u}_i+\boldsymbol{u}_j$ makes $a_1^1=1$ and $a_1^2=2$.

Then, for $n\leq 2$ and $k\leq 2$, we obtain the follows.

With $\boldsymbol{a}$: $v_1^1=0$, $v_2^1=1$, $v_1^2=1$, $v_2^2=3$; $f_1^1=1$, $f_2^1=3$, $f_1^2=2$, $f_2^2=4$.

With $\boldsymbol{a}+\boldsymbol{u}_i$: $v_1^1=1$, $v_2^1=2$, $v_1^2=1$, $v_2^2=3$; $f_1^1=2$, $f_2^1=4$, $f_1^2=3$, $f_2^2=5$.

With $\boldsymbol{a}+\boldsymbol{u}_j$: $v_1^1=0$, $v_2^1=1$, $v_1^2=2$, $v_2^2=3$; $f_1^1=1$, $f_2^1=3$, $f_1^2=3$, $f_2^2=4$.

With $\boldsymbol{a}+\boldsymbol{u}_i+\boldsymbol{u}_j$: $v_1^1=1$, $v_2^1=2$, $v_1^2=2$, $v_2^2=3$; $f_1^1=2$, $f_2^1=4$, $f_1^2=3$, $f_2^2=5$.

For $n>2$ or $k>2$, since $a_k^n\geq 6$, then $v_k^n=v_k^n(\boldsymbol{u}_i)=v_k^n(\boldsymbol{u}_j)=v_k^n(\boldsymbol{u}_i+\boldsymbol{u}_j)$ and $f_k^n=f_k^n(\boldsymbol{u}_i)=f_k^n(\boldsymbol{u}_j)=f_k^n(\boldsymbol{u}_i+\boldsymbol{u}_j)$.

Recall that $\mathit{\Gamma}_R^n=\sum_{k=2}^K(v_k^n-f_{k-1}^n)$, we have $\mathit{\Gamma}_R^2(a+\boldsymbol{u}_i)+\mathit{\Gamma}_R^2(\boldsymbol{a}+\boldsymbol{u}_j)=0$ which is strictly smaller than $\mathit{\Gamma}_R^2(\boldsymbol{a})+\mathit{\Gamma}_R^2(\boldsymbol{a}+\boldsymbol{u}_i+\boldsymbol{u}_j)=1$; and for $n\neq 2$, $\mathit{\Gamma}_R^n(\boldsymbol{a}+\boldsymbol{u}_i)+\mathit{\Gamma}_R^n(\boldsymbol{a}+\boldsymbol{u}_j)=\mathit{\Gamma}_R^n(\boldsymbol{a})+\mathit{\Gamma}_R^n(\boldsymbol{a}+\boldsymbol{u}_i+\boldsymbol{u}_j)$.

By the definitions of submodularity and L-convexity, we can conclude that, given any N, K, $\boldsymbol{T}$ and $C_R^n>0$, there exists a service time distribution such that the total recoverable idling cost $\sum_{n=1}^N C_R^n\mathit{\Gamma}_R^n(\boldsymbol{a})$ is not submodular or L-convex on $\boldsymbol{a}$.

5. Fix an arbitrary set of N, K and $\boldsymbol{T}$. We first show that $\sum_{n=1}^N C_O^n\mathit{\Gamma}_O^n(\boldsymbol{a}+\boldsymbol{1})=\sum_{n=1}^N C_O^n\mathit{\Gamma}_O^n(\boldsymbol{a})$. Note that $\boldsymbol{a}+\boldsymbol{1}$ indicates that the appointment times of all services are postponed by 1 slot, causing each element of $\boldsymbol{s}$, $\boldsymbol{r}$, $\boldsymbol{v}$ and $\boldsymbol{f}$ increasing by a constant number 1. By the definitions of $\mathit{\Gamma}_O^n(\boldsymbol{a})$, we have $\mathit{\Gamma}_O^n(\boldsymbol{a}+\boldsymbol{1})=\mathit{\Gamma}_O^n(\boldsymbol{a})$.

Next, we will show that $\mathit{\Gamma}_O^n(\boldsymbol{a})$ is submodular on $\boldsymbol{a}$ for any n.

Following Hajek (1985), to prove the submodularity of $\mathit{\Gamma}_O^n(\boldsymbol{a})$ is equivalent to prove

$$\mathit{\Gamma}_O^n(\boldsymbol{a}+\boldsymbol{u}_a)+\mathit{\Gamma}_O^n(\boldsymbol{a}+\boldsymbol{u}_b)\geq\mathit{\Gamma}_O^n(\boldsymbol{a})+\mathit{\Gamma}_O^n(\boldsymbol{a}+\boldsymbol{u}_a+\boldsymbol{u}_b)$$

for all $\boldsymbol{a}\in\mathbb{Z}^+$, $\boldsymbol{u}_a,\boldsymbol{u}_b\in\mathcal{U}$, $\boldsymbol{u}_a\neq\boldsymbol{u}_b$, where

$$\mathcal{U}=\left\{\begin{matrix}(1,0,0,\ldots,0),\\(0,1,0,\ldots,0),\\\ldots\\(0,0,\ldots,0,1)\end{matrix}\right\}. \tag{E.6}$$

Note that $\boldsymbol{a}+\boldsymbol{u}_a$ means that the appointment time of service a is postponed by 1 slot; $\boldsymbol{a}+\boldsymbol{u}_b$ means that the appointment time of service b is postponed by 1 slot; $\boldsymbol{a}+\boldsymbol{u}_a+\boldsymbol{u}_b$ means that the appointment times of service a and service b are postponed by 1 slot. A service here means the service of a patient at a particular stage. For instance, $a=i+(l-1)K$, so service a is the service of patient i at stage l.

Let ω denote a sample path. Then $\mathit{\Gamma}_O^n(\boldsymbol{a})=\mathbb{E}\left[\mathit{\Gamma}_O^n(\boldsymbol{a},\omega)\right]$. It is sufficient to show $\mathit{\Gamma}_O^n(\boldsymbol{a}+\boldsymbol{u}_a,\omega)+\mathit{\Gamma}_O^n(\boldsymbol{a}+\boldsymbol{u}_b,\omega)\geq\mathit{\Gamma}_O^n(\boldsymbol{a},\omega)+\mathit{\Gamma}_O^n(\boldsymbol{a}+\boldsymbol{u}_a+\boldsymbol{u}_b,\omega)$ holds for any sample path ω. In the following arguments, we will focus on one sample path, and ω is omitted for notational convenience. And we use $x(\boldsymbol{u}_y)$ to denote the value of x with perturbation y. By the definition of $\Gamma_O^n(\boldsymbol{a})$, to prove $\Gamma_O^n(\boldsymbol{a}+\boldsymbol{u}_a)+\Gamma_O^n(\boldsymbol{a}+\boldsymbol{u}_b) \geq \Gamma_O^n(\boldsymbol{a})+\Gamma_O^n(\boldsymbol{a}+\boldsymbol{u}_a+\boldsymbol{u}_b)$, it suffices to show

$$(s_K^n(\boldsymbol{u}_a)+\delta_K^n-T^n)^+ + (s_K^n(\boldsymbol{u}_b)+\delta_K^n-T^n)^+ \geq (s_K^n+\delta_K^n-T^n)^+ + (s_K^n(\boldsymbol{u}_a,\boldsymbol{u}_b)+\delta_K^n-T^n)^+, \tag{E.7}$$

for any n. We now will show that

$$s_k^n(\boldsymbol{u}_a)+s_k^n(\boldsymbol{u}_b) \geq s_k^n + s_k^n(\boldsymbol{u}_a,\boldsymbol{u}_b), \tag{E.8}$$

for any k and n.

Without loss of generality, we assume $a<b$, $a=i+(l-1)K$, and $b=j+(m-1)K$. So service a is the service of patient i at stage l, and service b is the service of patient j at stage m. The fact that $a<b$ indicates $l \leq m$; it also implies that if $j \leq i$ then $l<m$. Then we have $a_i^l(\boldsymbol{u}_a)=a_i^l+1$ under perturbation $\boldsymbol{u}_a$, $a_j^m(\boldsymbol{u}_b)=a_j^m+1$ under perturbation $\boldsymbol{u}_b$, and $a_i^l(\boldsymbol{u}_a,\boldsymbol{u}_b)=a_i^l+1$, $a_j^m(\boldsymbol{u}_a,\boldsymbol{u}_b)=a_j^m+1$ under perturbation $\boldsymbol{u}_a+\boldsymbol{u}_b$.

Recall the definition of $s_k^n=\max\{a_k^n, s_{k-1}^n+\delta_{k-1}^n, s_k^{n-1}+\delta_k^{n-1}\}$. Since the appointment time of any service is postponed by at most 1 slot, then for any k and n, $s_k^n(\boldsymbol{u}_a)-s_k^n$, $s_k^n(\boldsymbol{u}_b)-s_k^n$, and $s_k^n(\boldsymbol{u}_a,\boldsymbol{u}_b)-s_k^n$ are either 0 or 1. Then we have:

$s_k^n(\boldsymbol{u}_a,\boldsymbol{u}_b)=s_k^n(\boldsymbol{u}_a)=s_k^n(\boldsymbol{u}_b)=s_k^n$, for services which are not influenced by any perturbation. This happens for $n<l$, or $k<i$ and $l \leq n<m$, or $k<j$ and $k<i$.

$s_k^n(\boldsymbol{u}_a,\boldsymbol{u}_b)=s_k^n(\boldsymbol{u}_a)$ and $s_k^n(\boldsymbol{u}_b)=s_k^n$, for services which are only influenced by perturbation a. This happen for $k \geq i$ and $l \leq n<m$, or $i \leq k<j$ and $n \geq m$.

$s_k^n(\boldsymbol{u}_a,\boldsymbol{u}_b)=s_k^n(\boldsymbol{u}_b)$ and $s_k^n(\boldsymbol{u}_a)=s_k^n$, for services which are only influenced by perturbation b. This happens for $j \leq k<i$ and $n \geq m$.

For the cases of k and n above, we will have (E.8) holds. For other cases of k and n, i.e., services which are influenced by both perturbations, we have: 1) if $s_k^n(\boldsymbol{u}_a)=s_k^n+1$ or $s_k^n(\boldsymbol{u}_b)=s_k^n+1$, then (E.8) must hold; 2) if $s_k^n(\boldsymbol{u}_a)=s_k^n$ and $s_k^n(\boldsymbol{u}_b)=s_k^n$, we will show $s_k^n(\boldsymbol{u}_a,\boldsymbol{u}_b)=s_k^n$, then (E.8) holds. Next, we will prove that if $s_k^n(\boldsymbol{u}_a)=s_k^n$ and $s_k^n(\boldsymbol{u}_b)=s_k^n$ then $s_k^n(\boldsymbol{u}_a,\boldsymbol{u}_b)=s_k^n$ by contradiction. Let us suppose $s_k^n(\boldsymbol{u}_a)=s_k^n(\boldsymbol{u}_b)=s_k^n$ and $s_k^n(\boldsymbol{u}_a,\boldsymbol{u}_b)=s_k^n+1$.

Case (1) $i \leq j$: recall that these four systems (without perturbation, with perturbation a, with perturbation b, with perturbation $a+b$) have the same appointment times and service times after the service of patient j at stage m. Since $s_k^n(\boldsymbol{u}_a,\boldsymbol{u}_b)=s_k^n+1$, $s_k^n(\boldsymbol{u}_a)=s_k^n$ and $s_k^n(\boldsymbol{u}_b)=s_k^n$, then $s_j^m(\boldsymbol{u}_a,\boldsymbol{u}_b)=s_j^m+1$, $s_j^m(\boldsymbol{u}_a)=s_j^m$ and $s_j^m(\boldsymbol{u}_b)=s_j^m$. By the definition of s_j^m, we have $\max\{a_j^m, s_{j-1}^m(\boldsymbol{u}_a)+\delta_{j-1}^m, s_j^{m-1}(\boldsymbol{u}_a)+\delta_j^{m-1}\} = \max\{a_j^m+1, s_{j-1}^m(\boldsymbol{u}_b)+\delta_{j-1}^m, s_j^{m-1}(\boldsymbol{u}_b)+\delta_j^{m-1}\} = \max\{a_j^m, s_{j-1}^m+\delta_{j-1}^m, s_j^{m-1}+\delta_j^{m-1}\}$. Since $s_{j-1}^m(\boldsymbol{u}_a,\boldsymbol{u}_b) = s_{j-1}^m(\boldsymbol{u}_a)$, $s_j^{m-1}(\boldsymbol{u}_a,\boldsymbol{u}_b)=s_j^{m-1}(\boldsymbol{u}_a)$, $s_{j-1}^m(\boldsymbol{u}_b)=s_{j-1}^m$ and $s_j^{m-1}(\boldsymbol{u}_b)=s_j^{m-1}$, then $\max\{a_j^m, s_{j-1}^m(\boldsymbol{u}_a,\boldsymbol{u}_b)+\delta_{j-1}^m, s_j^{m-1}(\boldsymbol{u}_a,\boldsymbol{u}_b)+\delta_j^{m-1}\} = \max\{a_j^m+1, s_{j-1}^m+\delta_{j-1}^m, s_j^{m-1}+\delta_j^{m-1}\} = \max\{a_j^m, s_{j-1}^m+\delta_{j-1}^m, s_j^{m-1}+\delta_j^{m-1}\}$, which implies $\max\{s_{j-1}^m+\delta_{j-1}^m, s_j^{m-1}+\delta_j^{m-1}\} > a_j^m$. So $\max\{s_{j-1}^m(\boldsymbol{u}_a,\boldsymbol{u}_b)+\delta_{j-1}^m, s_j^{m-1}(\boldsymbol{u}_a,\boldsymbol{u}_b)+\delta_j^{m-1}\} > a_j^m$. Then we will have $s_j^m(\boldsymbol{u}_a,\boldsymbol{u}_b) = \max\{a_j^m+1, s_{j-1}^m(\boldsymbol{u}_a,\boldsymbol{u}_b)+\delta_{j-1}^m, s_j^{m-1}(\boldsymbol{u}_a,\boldsymbol{u}_b)+\delta_j^{m-1}\} = \max\{a_j^m, s_{j-1}^m+\delta_{j-1}^m, s_j^{m-1}+\delta_j^{m-1}\}=s_j^m$, which contradicts to $s_j^m(\boldsymbol{u}_a,\boldsymbol{u}_b)=s_j^m+1$. So we must have $s_k^n(\boldsymbol{u}_a,\boldsymbol{u}_b)=s_k^n$, if $s_k^n(\boldsymbol{u}_a)=s_k^n$ and $s_k^n(\boldsymbol{u}_b)=s_k^n$.

Case (2) $i>j$: recall that these four systems (without perturbation, with perturbation a, with perturbation b, with perturbation $a+b$) have the same appointment times and service times after the service of patient i at stage m. Since $s_k^n(\boldsymbol{u}_a,\boldsymbol{u}_b)=s_k^n+1$, $s_k^n(\boldsymbol{u}_a)=s_k^n$ and $s_k^n(\boldsymbol{u}_b)=s_k^n$, then $s_i^m(\boldsymbol{u}_a,\boldsymbol{u}_b)=s_i^m+1$, $s_i^m(\boldsymbol{u}_a)=s_i^m$ and $s_i^m(\boldsymbol{u}_b)=s_i^m$. By the definition of s_i^m, we have $\max\{a_i^m, s_{i-1}^m(\boldsymbol{u}_a)+\delta_{i-1}^m, s_i^{m-1}(\boldsymbol{u}_a)+\delta_i^{m-1}\} = \max\{a_i^m, s_{i-1}^m(\boldsymbol{u}_b)+\delta_{i-1}^m, s_i^{m-1}(\boldsymbol{u}_b)+\delta_i^{m-1}\} = \max\{a_i^m, s_{i-1}^m+\delta_{i-1}^m, s_i^{m-1}+\delta_i^{m-1}\}$, which implies: 1) $\max\{s_{i-1}^m(\boldsymbol{u}_a)+\delta_{i-1}^m, s_i^{m-1}(\boldsymbol{u}_a)+\delta_i^{m-1}\} \leq a_i^m$ and $\max\{s_{i-1}^m(\boldsymbol{u}_b)+\delta_{i-1}^m, s_i^{m-1}(\boldsymbol{u}_b)+\delta_i^{m-1}\} \leq a_i^m$; or 2)

$\max\{s_{i-1}^m(\boldsymbol{u}_a)+\delta_{i-1}^m, s_i^{m-1}(\boldsymbol{u}_a)+\delta_i^{m-1}\}=\max\{s_{i-1}^m(\boldsymbol{u}_b)+\delta_{i-1}^m, s_i^{m-1}(\boldsymbol{u}_b)+\delta_i^{m-1}\}>a_i^m$. Recall that $s_{i-1}^m(\boldsymbol{u}_a)=s_{i-1}^m$, $s_{i-1}^m(\boldsymbol{u}_a,\boldsymbol{u}_b)=s_{i-1}^m(\boldsymbol{u}_b)$, $s_i^{m-1}(\boldsymbol{u}_a,\boldsymbol{u}_b)=s_i^{m-1}(\boldsymbol{u}_a)$ and $s_i^{m-1}(\boldsymbol{u}_b)=s_i^{m-1}$. Then for the first situation, we must have $\max\{s_{i-1}^m+\delta_{i-1}^m, s_i^{m-1}(\boldsymbol{u}_a,\boldsymbol{u}_b)+\delta_i^{m-1}\}\leq a_i^m$ and $\max\{s_{i-1}^m(\boldsymbol{u}_a,\boldsymbol{u}_b)+\delta_{i-1}^m, s_i^{m-1}+\delta_i^{m-1}\}\leq a_i^m$, which imply that $\max\{a_i^m, s_{i-1}^m(\boldsymbol{u}_a,\boldsymbol{u}_b)+\delta_{i-1}^m, s_i^{m-1}(\boldsymbol{u}_a,\boldsymbol{u}_b)+\delta_i^{m-1}\}=a_i^m=\max\{a_i^m, s_{i-1}^m+\delta_{i-1}^m, s_i^{m-1}+\delta_i^{m-1}\}$, contradicting to $s_i^m(\boldsymbol{u}_a,\boldsymbol{u}_b)=s_i^m+1$. For the second situation, we have $\max\{s_{i-1}^m+\delta_{i-1}^m, s_i^{m-1}(\boldsymbol{u}_a,\boldsymbol{u}_b)+\delta_i^{m-1}\}=\max\{s_{i-1}^m(\boldsymbol{u}_a,\boldsymbol{u}_b)+\delta_{i-1}^m, s_i^{m-1}+\delta_i^{m-1}\}>a_i^m$. If $s_i^{m-1}(\boldsymbol{u}_a,\boldsymbol{u}_b)=s_i^{m-1}+1$, then we must have $s_{i-1}^m+\delta_{i-1}^m=s_{i-1}^m(\boldsymbol{u}_a,\boldsymbol{u}_b)+\delta_{i-1}^m\geq s_i^{m-1}(\boldsymbol{u}_a,\boldsymbol{u}_b)+\delta_i^{m-1}$, which implies $\max\{a_i^m, s_{i-1}^m(\boldsymbol{u}_a,\boldsymbol{u}_b)+\delta_{i-1}^m, s_i^{m-1}(\boldsymbol{u}_a,\boldsymbol{u}_b)+\delta_i^{m-1}\}=s_{i-1}^m+\delta_{i-1}^m$; if $s_{i-1}^m(\boldsymbol{u}_a,\boldsymbol{u}_b)=s_{i-1}^m+1$, then we must have $s_i^{m-1}+\delta_i^{m-1}=s_i^{m-1}(\boldsymbol{u}_a,\boldsymbol{u}_b)+\delta_i^{m-1}\geq s_{i-1}^m(\boldsymbol{u}_a,\boldsymbol{u}_b)+\delta_{i-1}^m$, which implies $\max\{a_i^m, s_{i-1}^m(\boldsymbol{u}_a,\boldsymbol{u}_b)+\delta_{i-1}^m, s_i^{m-1}(\boldsymbol{u}_a,\boldsymbol{u}_b)+\delta_i^{m-1}\}=s_i^{m-1}+\delta_i^{m-1}$; otherwise $s_{i-1}^m(\boldsymbol{u}_a,\boldsymbol{u}_b)=s_{i-1}^m$ and $s_i^{m-1}(\boldsymbol{u}_a,\boldsymbol{u}_b)=s_i^{m-1}$. All cases contradict to $s_i^m(\boldsymbol{u}_a,\boldsymbol{u}_b)=s_i^m+1$. So we must have $s_k^n(\boldsymbol{u}_a,\boldsymbol{u}_b)=s_k^n$, if $s_k^n(\boldsymbol{u}_a)=s_k^n$ and $s_k^n(\boldsymbol{u}_b)=s_k^n$.

Now we have $s_K^n(\boldsymbol{u}_a)+s_K^n(\boldsymbol{u}_b)\geq s_K^n+s_K^n(\boldsymbol{u}_a,\boldsymbol{u}_b)$ for any n. Since $(s_K^n(\boldsymbol{u}_a)-s_K^n)$, $(s_K^n(\boldsymbol{u}_b)-s_K^n)$ and $(s_K^n(\boldsymbol{u}_a,\boldsymbol{u}_b)-s_K^n)$ are either 1 or 0, then (E.7) must hold for any n.

By the definition of L-convexity, we can conclude that, given any N, K, $\boldsymbol{T}$, $\{C_O^n, n=1,2,\ldots,N\}$ and any service time distributions, the total overtime cost $\sum_{n=1}^N C_O^n \mathit{\Gamma}_O^n(\boldsymbol{a})$ is submodular and L-convex on $\boldsymbol{a}$.

6. Since the regular waiting cost, the leisure waiting cost, the regular idling cost and the recoverable idling cost are not submodular or L-convex on $\boldsymbol{a}$ for some service time distributions, neither is the total cost. □

Proof of Lemma 2: Proposition 1 has shown that the total overtime cost is L-convex on $\boldsymbol{a}$ for any given service time distribution. Hence, to prove Lemma 2, it suffices to show that the total waiting cost and the total idling cost are L-convex on $\boldsymbol{a}$ for any given service time distribution if $C_{L,k}=C_{W,k},\forall k$ and $C_R^n=C_I^n,\forall n$.

1. We first show that the total waiting cost $\sum_{k=1}^K\big(C_{W,k}\mathit{\Gamma}_{W,k}(\boldsymbol{a})+C_{L,k}\mathit{\Gamma}_{L,k}(\boldsymbol{a})\big)$ is L-convex on $\boldsymbol{a}$ for any service time distributions if $C_{L,k}=C_{W,k},\forall k$.

We start by showing that $\sum_{k=1}^K\big(C_{W,k}\mathit{\Gamma}_{W,k}(\boldsymbol{a}+\boldsymbol{1})+C_{L,k}\mathit{\Gamma}_{L,k}(\boldsymbol{a}+\boldsymbol{1})\big)=\sum_{k=1}^K\big(C_{W,k}\mathit{\Gamma}_{W,k}(\boldsymbol{a})+C_{L,k}\mathit{\Gamma}_{L,k}(\boldsymbol{a})\big)$. Note that $\boldsymbol{a}+\boldsymbol{1}$ indicates that the appointment times of all services are postponed by 1 slot, causing $\boldsymbol{s}$, $\boldsymbol{r}$, $\boldsymbol{v}$ and $\boldsymbol{f}$ increasing by a constant number 1. By the definitions of $\mathit{\Gamma}_{W,k}(\boldsymbol{a})$ and $\mathit{\Gamma}_{L,k}(\boldsymbol{a})$, we have

$$\sum_{k=1}^K\big(C_{W,k}\mathit{\Gamma}_{W,k}(\boldsymbol{a}+\boldsymbol{1})+C_{L,k}\mathit{\Gamma}_{L,k}(\boldsymbol{a}+\boldsymbol{1})\big)=\sum_{k=1}^K\big(C_{W,k}\mathit{\Gamma}_{W,k}(\boldsymbol{a})+C_{L,k}\mathit{\Gamma}_{L,k}(\boldsymbol{a})\big).$$

Next, we show that $\sum_{k=1}^K\big(C_{W,k}\mathit{\Gamma}_{W,k}(\boldsymbol{a})+C_{L,k}\mathit{\Gamma}_{L,k}(\boldsymbol{a})\big)$ is submodular on $\boldsymbol{a}$ when $C_{L,k}=C_{W,k}$. Let ω denote a sample path. Then $\mathit{\Gamma}_{W,k}(\boldsymbol{a})=\mathbb{E}\big[\mathit{\Gamma}_{W,k}(\boldsymbol{a},\omega)\big]$ and $\mathit{\Gamma}_{L,k}(\boldsymbol{a})=\mathbb{E}\big[\mathit{\Gamma}_{L,k}(\boldsymbol{a},\omega)\big]$. It suffices to show $\sum_{k=1}^K\big(C_{W,k}\mathit{\Gamma}_{W,k}(\boldsymbol{a},\omega)+C_{L,k}\mathit{\Gamma}_{L,k}(\boldsymbol{a},\omega)\big)$ is submodular for any sample path ω. In the following arguments, we will focus on one sample path, and ω is omitted for notational convenience.

When $C_{L,k}=C_{W,k}$, we have $\sum_{k=1}^K\big(C_{W,k}\mathit{\Gamma}_{W,k}(\boldsymbol{a})+C_{L,k}\mathit{\Gamma}_{L,k}(\boldsymbol{a})\big)=\sum_{k=1}^K C_{W,k}\big(\sum_{n=1}^N(s_k^n-r_k^n)+\sum_{n=2}^N(r_k^n-s_k^{n-1}-\delta_k^{n-1})\big)=\sum_{k=1}^K C_{W,k}\big(s_k^N-a_k^1-\sum_{n=1}^{N-1}\delta_k^n\big)$ by the definition of $\mathit{\Gamma}_{W,k}(\boldsymbol{a})$ and $\mathit{\Gamma}_{L,k}(\boldsymbol{a})$. Then it suffices to prove, for any n, that

$$s_k^N(\boldsymbol{u}_a)-a_k^1(\boldsymbol{u}_a)+s_k^N(\boldsymbol{u}_b)-a_k^1(\boldsymbol{u}_b)\geq s_k^N-a_k^1+s_k^N(\boldsymbol{u}_a,\boldsymbol{u}_b)-a_k^1(\boldsymbol{u}_a,\boldsymbol{u}_b), \tag{E.9}$$

for any $\boldsymbol{u}_a,\boldsymbol{u}_b\in\mathcal{U}$, $\boldsymbol{u}_a\neq\boldsymbol{u}_b$, where $\mathcal{U}$ is defined as (E.6). Here we use $x(\boldsymbol{u}_y)$ to denote the value of x with perturbation y.

Recall part 5 in the proof of Proposition 1. We have already shown that, for any k and n, s_k^n is submodular on $\boldsymbol{a}$, i.e.,

$$s_k^n(\boldsymbol{u}_a)+s_k^n(\boldsymbol{u}_b)\geq s_k^n+s_k^n(\boldsymbol{u}_a,\boldsymbol{u}_b).$$

Without loss of generality, we assume $a < b$.

If $a \neq k$ or $b \neq k$, then the appointment time of patient k at stage 1 is not influenced by perturbation a or b, i.e., $a_k^1 = a_k^1(\boldsymbol{u}_a) = a_k^1(\boldsymbol{u}_b) = a_k^1(\boldsymbol{u}_a, \boldsymbol{u}_b)$. Then (E.9) holds.

When $a = k$, $a_k^1(\boldsymbol{u}_a) = a_k^1(\boldsymbol{u}_a, \boldsymbol{u}_b) = a_k^1 + 1$ and $a_k^1(\boldsymbol{u}_b) = a_k^1$. Then (E.9) holds.

When $b = k$, $a_k^1(\boldsymbol{u}_b) = a_k^1(\boldsymbol{u}_a, \boldsymbol{u}_b) = a_k^1 + 1$ and $a_k^1(\boldsymbol{u}_a) = a_k^1$. Then (E.9) holds.

Thus, by the definition of L-convexity, we can conclude that, the total waiting cost $\sum_{k=1}^K \big(C_{W,k}\Gamma_{W,k}(\boldsymbol{a}) + C_{L,k}\Gamma_{L,k}(\boldsymbol{a})\big)$ is L-convex on $\boldsymbol{a}$ for any given service time distribution if $C_{L,k} = C_{W,k}, \forall k$.

2. Now, we will show that the total idling cost $\sum_{n=1}^N \big(C_I^n \Gamma_I^n(\boldsymbol{a}) + C_R^n \Gamma_R^n(\boldsymbol{a})\big)$ is L-convex on $\boldsymbol{a}$ for any service time distribution if $C_R^n = C_I^n, \forall n$.

First, following a similar argument above, we can show that

$$\sum_{n=1}^N \big(C_I^n \Gamma_I^n(\boldsymbol{a}+\boldsymbol{1}) + C_R^n \Gamma_R^n(\boldsymbol{a}+\boldsymbol{1})\big) = \sum_{n=1}^N \big(C_I^n \Gamma_I^n(\boldsymbol{a}) + C_R^n \Gamma_R^n(\boldsymbol{a})\big).$$

Next, we will show that $\sum_{n=1}^N \big(C_I^n \Gamma_I^n(\boldsymbol{a}) + C_R^n \Gamma_R^n(\boldsymbol{a})\big)$ is submodular on $\boldsymbol{a}$ when $C_R^n = C_I^n$. Let ω denote a sample path. Then $\Gamma_I^n(\boldsymbol{a}) = \mathbb{E}\big[\Gamma_I^n(\boldsymbol{a}, \omega)\big]$ and $\Gamma_R^n(\boldsymbol{a}) = \mathbb{E}\big[\Gamma_R^n(\boldsymbol{a}, \omega)\big]$. It suffices to show $\sum_{n=1}^N \big(C_I^n \Gamma_I^n(\boldsymbol{a}, \omega) + C_R^n \Gamma_R^n(\boldsymbol{a}, \omega)\big)$ is submodular for any sample path ω. In the following arguments, we will focus on on sample path, and omit ω in our notations for convenience.

When $C_R^n = C_I^n$, we will have $\sum_{n=1}^N \big(C_I^n \Gamma_I^n(\boldsymbol{a}) + C_R^n \Gamma_R^n(\boldsymbol{a})\big) = \sum_{n=1}^N C_I^n \big(\sum_{k=1}^K (s_k^n - v_k^n) + \sum_{k=2}^K (v_k^n - s_{k-1}^n - \delta_{k-1}^n)\big) = \sum_{n=1}^N C_I^n \big(s_K^n - a_1^n - \sum_{k=1}^{K-1} \delta_k^n\big)$ by the definition of $\Gamma_I^n(\boldsymbol{a})$ and $\Gamma_R^n(\boldsymbol{a})$. Then it suffices to prove that

$$s_K^n(\boldsymbol{u}_a) - a_1^n(\boldsymbol{u}_a) + s_K^n(\boldsymbol{u}_b) - a_1^n(\boldsymbol{u}_b) \geq s_K^n - a_1^n + s_K^n(\boldsymbol{u}_a, \boldsymbol{u}_b) - a_1^n(\boldsymbol{u}_a, \boldsymbol{u}_b), \tag{E.10}$$

for any n, where $\boldsymbol{u}_a, \boldsymbol{u}_b \in \mathcal{U}$, $\boldsymbol{u}_a \neq \boldsymbol{u}_b$, where $\mathcal{U}$ is defined as (E.6). Here we use $x(\boldsymbol{u}_y)$ to denote the value of x with perturbation y.

Recall part 5 in the proof of Proposition 1. We have already shown that, for any k and n, s_k^n is submodular on $\boldsymbol{a}$, i.e.,

$$s_k^n(\boldsymbol{u}_a) + s_k^n(\boldsymbol{u}_b) \geq s_k^n + s_k^n(\boldsymbol{u}_a, \boldsymbol{u}_b).$$

Without loss of generality, we assume $a < b$.

If $a \neq 1 + (n-1)K$ or $b \neq 1 + (n-1)K$, then the appointment time of patient 1 at stage n is not influenced by perturbation a or b, i.e., $a_1^n = a_1^n(u_a) = a_1^n(\boldsymbol{u}_b) = a_1^n(\boldsymbol{u}_a, \boldsymbol{u}_b)$. Then (E.10) holds.

If $a = 1 + (n-1)K$, $a_1^n(\boldsymbol{u}_a) = a_1^n(\boldsymbol{u}_a, \boldsymbol{u}_b) = a_1^n + 1$ and $a_1^n(\boldsymbol{u}_b) = a_1^n$. Then (E.10) holds.

If $b = 1 + (n-1)K$, $a_1^n(\boldsymbol{u}_b) = a_1^n(\boldsymbol{u}_a, \boldsymbol{u}_b) = a_1^n + 1$ and $a_1^n(\boldsymbol{u}_a) = a_1^n$. Then (E.10) holds.

Thus, by the definition of L-convexity, we can conclude that the total idling cost $\sum_{n=1}^N \big(C_I^n \Gamma_I^n(\boldsymbol{a}) + C_R^n \Gamma_R^n(\boldsymbol{a})\big)$ is L-convex on $\boldsymbol{a}$ for any service time distribution if $C_R^n = C_I^n, \forall n$. This completes the proof. □

Proof of Proposition 2: In the constraint set, we apply the Big M method to reformulate constraint (1), i.e.,

$$r_k^n(\omega) = \begin{cases} a_k^1 & \text{if } n = 1, \\ \max\{a_k^n, s_k^{n-1}(\omega) + \delta_k^{n-1}(\omega)\} & \text{if } n \geq 2, \end{cases}$$

then for $n \geq 2$, we have $r_k^n(\omega) \geq a_k^n$, $r_k^n(\omega) \geq s_k^{n-1}(\omega) + \delta_k^{n-1}(\omega)$, $r_k^n(\omega) \leq a_k^n + M(1 - z_k^n(\omega))$ and $r_k^n(\omega) \leq s_k^{n-1}(\omega) + \delta_k^{n-1}(\omega) + M z_k^n(\omega)$, where M is a sufficiently large number and $z_k^n(\omega)$ is a binary variable.

Similarly, we reformulate constraint (2), i.e.,

$$v_k^n(\omega) = \begin{cases} a_1^n & \text{if } k = 1, \\ \max\{a_k^n, s_{k-1}^n(\omega) + \delta_{k-1}^n(\omega)\} & \text{if } k \geq 2, \end{cases}$$

then for $k \geq 2$, we have $v_k^n(\omega) \geq a_k^n$, $v_k^n(\omega) \geq s_{k-1}^n(\omega) + \delta_{k-1}^n(\omega)$, $v_k^n(\omega) \leq a_k^n + M(1 - y_k^n(\omega))$ and $v_k^n(\omega) \leq s_{k-1}^n(\omega) + \delta_{k-1}^n(\omega) + My_k^n(\omega)$, where M is a sufficiently large number and $y_k^n(\omega)$ is a binary variable.

Finally, we rewrite $s_k^n(\omega) = \max\{r_k^n(\omega), v_k^n(\omega)\}$ as $s_k^n(\omega) \geq r_k^n(\omega)$ and $s_k^n(\omega) \geq v_k^n(\omega)$, and $f^n(\omega) = \max\{T^n, s_K^n(\omega) + \delta_K^n(\omega)\}$ as $f^n(\omega) \geq T^n$ and $f^n(\omega) \geq s_K^n(\omega) + \delta_K^n(\omega)$. These relaxations are exact because we are dealing with a minimization problem. □

Proof of Proposition 3: Substituting $r_k^n(\omega)$ by $\max\{a_k^n, s_k^{n-1}(\omega) + \delta_k^{n-1}(\omega)\}$ and $v_k^n(\omega)$ by $\max\{a_k^n, s_{k-1}^n(\omega) + \delta_{k-1}^n(\omega)\}$ in the original objective function in (**MILP**) leads to the objective function of (**CP**). These substitutions reformulate the waiting cost and idling cost as follows.

$$
\begin{aligned}
&C_{W,k}\Gamma_{W,k}(\omega) + C_{L,k}\Gamma_{L,k}(\omega) = C_{W,k}\sum_{n=1}^{N}\left(s_k^n(\omega) - r_k^n(\omega)\right) + C_{L,k}\sum_{n=2}^{N}\left(r_k^n(\omega) - s_k^{n-1}(\omega) - \delta_k^{n-1}(\omega)\right)\\
=&C_{W,k}\left[\sum_{n=1}^{N}\left(s_k^n(\omega) - r_k^n(\omega)\right) + \sum_{n=2}^{N}\left(r_k^n(\omega) - s_k^{n-1}(\omega) - \delta_k^{n-1}(\omega)\right)\right] - (C_{W,k} - C_{L,k})\sum_{n=2}^{N}\left(r_k^n(\omega) - s_k^{n-1}(\omega) - \delta_k^{n-1}(\omega)\right)\\
=&C_{W,k}\big(s_k^N(\omega) - r_k^1(\omega) - \sum_{n=1}^{N-1}\delta_k^n(\omega)\big) - (C_{W,k} - C_{L,k})\sum_{n=2}^{N}\left(r_k^n(\omega) - s_k^{n-1}(\omega) - \delta_k^{n-1}(\omega)\right)\\
=&C_{W,k}\big(s_k^N(\omega) - a_k^1 - \sum_{n=1}^{N-1}\delta_k^n(\omega)\big) - (C_{W,k} - C_{L,k})\sum_{n=2}^{N}\left(\max\{a_k^n, s_k^{n-1}(\omega) + \delta_k^{n-1}(\omega)\} - s_k^{n-1}(\omega) - \delta_k^{n-1}(\omega)\right)\\
=&C_{W,k}\big(s_k^N(\omega) - a_k^1 - \sum_{n=1}^{N-1}\delta_k^n(\omega)\big) - (C_{W,k} - C_{L,k})\sum_{n=2}^{N}\max\{a_k^n - s_k^{n-1}(\omega) - \delta_k^{n-1}(\omega), 0\},
\end{aligned}
$$

and

$$
\begin{aligned}
&C_I^n\Gamma_I^n(\omega) + C_R^n\Gamma_R^n(\omega) = C_I^n\sum_{k=1}^{K}\left(s_k^n(\omega) - v_k^n(\omega)\right) + C_R^n\sum_{k=2}^{K}\left(v_k^n(\omega) - s_{k-1}^n(\omega) - \delta_{k-1}^n(\omega)\right)\\
=&C_I^n\left[\sum_{k=1}^{K}\left(s_k^n(\omega) - v_k^n(\omega)\right) + \sum_{k=2}^{K}\left(v_k^n(\omega) - s_{k-1}^n(\omega) - \delta_{k-1}^n(\omega)\right)\right] - (C_I^n - C_R^n)\sum_{k=2}^{K}\left(v_k^n(\omega) - s_{k-1}^n(\omega) - \delta_{k-1}^n(\omega)\right)\\
=&C_I^n\big(s_K^n(\omega) - r_1^n(\omega) - \sum_{k=1}^{K-1}\delta_k^n(\omega)\big) - (C_I^n - C_R^n)\sum_{k=2}^{K}\left(v_k^n(\omega) - s_{k-1}^n(\omega) - \delta_{k-1}^n(\omega)\right)\\
=&C_I^n\big(s_K^n(\omega) - a_1^n - \sum_{k=1}^{K-1}\delta_k^n(\omega)\big) - (C_I^n - C_R^n)\sum_{k=2}^{K}\left(\max\{a_k^n, s_{k-1}^n(\omega) + \delta_{k-1}^n(\omega)\} - s_{k-1}^n(\omega) - \delta_{k-1}^n(\omega)\right)\\
=&C_I^n\big(s_K^n(\omega) - a_1^n - \sum_{k=1}^{K-1}\delta_k^n(\omega)\big) - (C_I^n - C_R^n)\sum_{k=2}^{K}\max\{a_k^n - s_{k-1}^n(\omega) - \delta_{k-1}^n(\omega), 0\}.
\end{aligned}
$$

Then, we are left to deal with the constraint set. We first show that relaxing $s_k^n(\omega) = \max\{a_k^n, r_k^n(\omega), v_k^n(\omega)\}$ to $s_k^n(\omega) \geq a_k^n$, $s_k^n(\omega) \geq s_k^{n-1}(\omega) + \delta_k^{n-1}(\omega)$ and $s_k^n(\omega) \geq s_{k-1}^n(\omega) + \delta_{k-1}^n(\omega)$ is exact. Suppose that there exists an optimal solution such that $s_k^n(\omega) > a_k^n$, $s_k^n(\omega) > s_k^{n-1}(\omega) + \delta_k^{n-1}(\omega)$ and $s_k^n(\omega) > s_{k-1}^n(\omega) + \delta_{k-1}^n(\omega)$ for some k and n. We can construct another solution with $s_k^n(\omega)' = \max\{a_k^n, s_k^{n-1}(\omega) + \delta_k^{n-1}(\omega), s_{k-1}^n(\omega) + \delta_{k-1}^n(\omega)\}$ and other decision variables unchanged. It is clear that all constrains are still satisfied. The terms associated with $s_k^n(\omega)$ in the objective function change to

$$
C_{W,k}s_k^n(\omega)' + C_{L,k}\big(\max\{a_k^{n+1}, s_k^n(\omega)' + \delta_k^n(\omega)\} - s_k^n(\omega)' - \delta_k^n(\omega)\big)
$$

and

$$
C_I^n s_k^n(\omega)' + C_R^n\big(\max\{a_{k+1}^n, s_k^n(\omega) + \delta_k^n(\omega)\} - s_k^n(\omega)' - \delta_k^n(\omega)\big).
$$

After reorganizing these terms, we have

$$(C_{W,k}-C_{L,k})s_k^n(\omega)'+C_{L,k}\big(\max\{a_k^{n+1},s_k^n(\omega)'+\delta_k^n(\omega)\}-\delta_k^n(\omega)\big)$$

and

$$(C_I^n-C_R^n)s_k^n(\omega)'+C_R^n\big(\max\{a_{k+1}^n,s_k^n(\omega)'+\delta_k^n(\omega)\}-\delta_k^n(\omega)\big).$$

Since $C_{W,k}-C_{L,k}\geq 0$ and $C_I^n-C_R^n\geq 0$, decreasing $s_k^n(\omega)$ to $s_k^n(\omega)'$ does not increase the objective value. We can continue doing this until arriving at a solution which satisfies $s_k^n(\omega)=\max\{r_k^n(\omega),v_k^n(\omega)\}$ and generates no greater cost. Therefore, relaxing $s_k^n(\omega)=\max\{r_k^n(\omega),v_k^n(\omega)\}$ to $s_k^n(\omega)\geq r_k^n(\omega)$ and $s_k^n(\omega)\geq v_k^n(\omega)\}$ is exact.

Similarly, we can relax $f^n(\omega)=\max\{T^n,s_K^n(\omega)+\delta_K^n(\omega)\}$ to $f^n(\omega)\geq T^n$ and $f^n(\omega)\geq s_K^n(\omega)+\delta_K^n(\omega)$. Suppose that there exists an optimal solution such that $f^n(\omega)>T^n$ and $f^n(\omega)>s_K^n(\omega)+\delta_K^n(\omega)$ for some n. We can construct another solution with $f^n(\omega)'=\max\{T^n,s_K^n(\omega)+\delta_K^n(\omega)\}$ and other variables unchanged. It is evident that all constrains are still satisfied. The terms associated with $f^n(\omega)$ in the objective function change to $C_O^n(f^n(\omega)'-T^n)$.

So decreasing $f^n(\omega)$ to $f^n(\omega)'$ does not increase the objective value. We can continue doing this until arriving at a solution which satisfies $f^n(\omega)=\max\{T^n,s_K^n(\omega)+\delta_K^n(\omega)\}$ and generates no greater cost. Hence, relaxing $f^n(\omega)=\max\{T^n,s_K^n(\omega)+\delta_K^n(\omega)\}$ to $f^n(\omega)>T^n$ and $f^n(\omega)>s_K^n(\omega)+\delta_K^n(\omega)$ is exact. This completes the proof. □

Proof of Lemma 4: By the definition of $\boldsymbol{U}$, the matrix is full-rank and has $|\boldsymbol{x}|$ independent columns. Thus $\mathcal{M}$ has $|\boldsymbol{x}|$ edges. Recall that $[\boldsymbol{I},\boldsymbol{W}]$ is the associated simplex tableau of $(\boldsymbol{x}^0,\boldsymbol{w}^0)$ (note that the columns in $[\boldsymbol{I},\boldsymbol{W}]$ are reordered by the basis and non-basis). According to the definition of $\boldsymbol{U}$, each column $\boldsymbol{u}^i$ represents the direction from the neighbor vertex to $\boldsymbol{x}^0$. Thus there must be one vertex of $\mathcal{D}$ on each edge of $\mathcal{M}$. Then we must have $\mathcal{M}$ covers $\mathcal{D}$ because the space dimension is $|\boldsymbol{x}|$ and $\mathcal{M}$ has $|\boldsymbol{x}|$ edges. □

Proof of Lemma 5: Recall that $\boldsymbol{u}^i$ is the i^{th} edge of the cone $\mathcal{M}$. According to the definition of θ_i, $\boldsymbol{x}^0+\theta_i\boldsymbol{u}^i$ is the one where the i^{th} edge meets the boundary of $\mathcal{C}(U-\epsilon)$. Let us denote $\boldsymbol{x}^0+\theta_i\boldsymbol{u}^i$ as $\boldsymbol{x}_i$. Thus we have $\sum_{i=1}^{|\boldsymbol{x}|}\frac{t_i}{\theta_i}=1$, where $\boldsymbol{t}=\boldsymbol{U}^{-1}(\boldsymbol{x}-\boldsymbol{x}^0)$, defines the hyperplane passing through $\boldsymbol{x}_1$, ..., $\boldsymbol{x}_{|\boldsymbol{x}|}$.

Note that $\boldsymbol{\theta}^{-1}\boldsymbol{U}^{-1}(\boldsymbol{x}^0-\boldsymbol{x}^0)=0$, where $\boldsymbol{\theta}^{-1}=(\frac{1}{\theta_1},\frac{1}{\theta_2},\ldots,\frac{1}{\theta_{|\boldsymbol{x}|}})$. Then $\boldsymbol{x}^0$ is not on the hyperplane $\boldsymbol{\theta}^{-1}\boldsymbol{U}^{-1}(\boldsymbol{x}-\boldsymbol{x}^0)=1$. It follows that $\boldsymbol{x}^0,\boldsymbol{x}^1,...,\boldsymbol{x}^{|\boldsymbol{x}|}$ compose a polyhedron in $|\boldsymbol{x}|$-dimension space. Denote this polyhedron as $\mathcal{H}$. So $\boldsymbol{x}^0,\boldsymbol{x}^1,...,\boldsymbol{x}^{|\boldsymbol{x}|}$ are the vertexes of $\mathcal{H}$. Recall that $\Gamma(\boldsymbol{x}_i)\geq U-\epsilon$ for $i=0,1,...,|\boldsymbol{x}|$. Then all vertexes of $\mathcal{H}$ are in $\mathcal{C}(U-\epsilon)$. Because $\Gamma(\boldsymbol{x})$ is concave, $\mathcal{C}(U-\epsilon)$ is convex by its definition. Thus we have $\mathcal{H}\subseteq\mathcal{C}(U-\epsilon)$. So for any $\boldsymbol{x}\notin\mathcal{C}(U-\epsilon)$, we have $\Gamma(\boldsymbol{x})<U-\epsilon$, and hence $\boldsymbol{x}\notin\mathcal{H}$. It follows that $\boldsymbol{\theta}^{-1}\boldsymbol{U}^{-1}(\boldsymbol{x}-\boldsymbol{x}^0)\geq 1$. □

Proof of Lemma 6: Note that $\boldsymbol{t}^{\mathcal{M}}$ is a vertex of the polyhedron $\{\boldsymbol{t}|\boldsymbol{A}\boldsymbol{U}\boldsymbol{t}\geq\boldsymbol{b}_1-\boldsymbol{A}\boldsymbol{x}^0,\boldsymbol{U}\boldsymbol{t}\leq\boldsymbol{b}_2-\boldsymbol{x}^0,-\boldsymbol{U}\boldsymbol{t}\leq\boldsymbol{x}^0,\boldsymbol{t}\geq 0\}$. Replacing $\boldsymbol{x}^0+\boldsymbol{U}\boldsymbol{t}$ by $\boldsymbol{x}$, we have $\boldsymbol{x}^{\mathcal{M}}=\boldsymbol{x}^0+\boldsymbol{U}\boldsymbol{t}^{\mathcal{M}}$ is a vertex of the polyhedron $\{\boldsymbol{x}|\boldsymbol{A}\boldsymbol{x}\geq\boldsymbol{b}_1,\boldsymbol{x}\leq\boldsymbol{b}_2,\boldsymbol{x}\geq 0,\boldsymbol{x}=\boldsymbol{x}^0+\boldsymbol{U}\boldsymbol{t},\boldsymbol{t}\geq 0\}$, i.e., $\boldsymbol{x}^{\mathcal{M}}$ is a vertex of $\mathcal{D}\cap\mathcal{M}$.

If $\psi^{\mathcal{M}}\leq 1$, then $\mathcal{D}\cap\mathcal{M}$ must be covered by the half-space which cannot contain a point yielding smaller value.

If $\psi^{\mathcal{M}}>1$ and $\Gamma(\boldsymbol{x}^{\mathcal{M}})<U$ then we find a better solution $\boldsymbol{x}^{\mathcal{M}}$. Since (**LP-**$\mathcal{DM}$) is an LP, we have $\boldsymbol{x}^{\mathcal{M}}$ is a vertex in $\mathcal{D}$.

Finally, we consider the case when $\psi^{\mathcal{M}}>1$ and $\Gamma(\boldsymbol{x}^{\mathcal{M}})\geq U$. Suppose that $\boldsymbol{x}^{\mathcal{M}}$ lies on some edge of $\mathcal{M}$. Since $\psi^{\mathcal{M}}>1$ and $\Gamma(\boldsymbol{x}_i)=U-\epsilon$ for $\psi(\boldsymbol{x}_i)=1$, we have $\Gamma(\boldsymbol{x}^{\mathcal{M}})<U-\epsilon$, leading to a contradiction. Hence we must have that $\boldsymbol{x}^{\mathcal{M}}$ does not lie on any edge of $\mathcal{M}$. □

Proof of Lemma 7: Because $\boldsymbol{x}^{\mathcal{M}}$ does not lie on any edge of $\mathcal{M}$ and $\boldsymbol{x}^{\mathcal{M}} \in \mathcal{M}$, $\boldsymbol{x}^{\mathcal{M}}$ must be an internal point of $\mathcal{M}$. So the ray $\boldsymbol{x}^{\mathcal{M}} - \boldsymbol{x}^0 = \boldsymbol{U}\boldsymbol{t}^{\mathcal{M}}$ which is from $\boldsymbol{x}^0$ through $\boldsymbol{x}^{\mathcal{M}}$ is not parallel to any edge of $\mathcal{M}$, i.e., $\boldsymbol{u}^1$, $\boldsymbol{u}^2$, ...,$\boldsymbol{u}^{|\boldsymbol{x}|}$. Then combining any $|\boldsymbol{x}|-1$ of these edges and $\boldsymbol{U}\boldsymbol{t}^{\mathcal{M}}$, we can get a sub-cone with $|\boldsymbol{x}|$ edges. By this way, the cone $\mathcal{M}$ is partitioned to $|\boldsymbol{x}|$ sub-cones with $|\boldsymbol{x}|$ edges. □

Proof of Proposition 4: First, note that (**Sub-LP**) and (**CP**) have similar constraints for $\boldsymbol{s}', \boldsymbol{f}'$. Next, we show that the solution $(\boldsymbol{a}, \boldsymbol{s}', \boldsymbol{f}')$ to (**Sub-LP**) must satisfy $s_k^n(\omega)' = \max\{a_k^n, s_k^{n-1}(\omega)' + \delta_k^{n-1}(\omega), s_{k-1}^n(\omega)' + \delta_{k-1}^n(\omega)\}$ and $f^n(\omega)' = \max\{T^n, s_k^n(\omega)' + \delta_K^n(\omega)\}$ by contradiction. Suppose there exists some $s_k^n(\omega)'$ such that $s_k^n(\omega)' > \max\{a_k^n, s_k^{n-1}(\omega)' + \delta_k^{n-1}(\omega), s_{k-1}^n(\omega)' + \delta_{k-1}^n(\omega)\}$, we can replace $s_k^n(\omega)'$ by $\max\{a_k^n, s_k^{n-1}(\omega)' + \delta_k^{n-1}(\omega), s_{k-1}^n(\omega)' + \delta_{k-1}^n(\omega)\}$ without violating the constraints or increasing the objective value. Similarly, suppose there exists some $f^n(\omega)' > \max\{T^n, s_k^n(\omega)' + \delta_K^n(\omega)\}$, then we can replace $f^n(\omega)'$ by $\max\{T^n, s_k^n(\omega)' + \delta_K^n(\omega)\}$ without violating the constraints or increasing the objective value. So in the optimal solution to (**Sub-LP**), we must have $s_k^n(\omega)' = \max\{a_k^n, s_k^{n-1}(\omega)' + \delta_k^{n-1}(\omega), s_{k-1}^n(\omega)' + \delta_{k-1}^n(\omega)\}$ and $f^n(\omega)' = \max\{T^n, s_k^n(\omega)' + \delta_K^n(\omega)\}$. It follows that the solution $(\boldsymbol{a}, \boldsymbol{s}', \boldsymbol{f}')$ is feasible to (**CP**) and achieves the smallest $\boldsymbol{s}$ and $\boldsymbol{f}$ for the given $\boldsymbol{a}$. In (**CP**), the objective function is a decreasing function in $\boldsymbol{s}$ and $\boldsymbol{f}$, so $(\boldsymbol{a}, \boldsymbol{s}', \boldsymbol{f}')$ achieves the smallest objective value for the given $\boldsymbol{a}$. Thus $\Gamma(\boldsymbol{a}, \boldsymbol{s}', \boldsymbol{f}') \leq \Gamma(\boldsymbol{x})$. □

Proof of Proposition 5: We first show that Algorithm 1 terminates after a finite number of steps. To do this, we first give a definition for a minimum sub-cone.

Definition 3 (Minimum Sub-cone)**.** *For a sub-cone $\mathcal{M}_{min}$, if the number of vertexes of $\mathcal{D} \cap \mathcal{M}_{min}$ is exactly $|\boldsymbol{x}|+1$, then $\mathcal{M}_{min}$ is a minimum sub-cone.*

Proposition E.1. *For a minimum sub-cone $\mathcal{M}_{min}$, in Step 8, we either have $\psi^{\mathcal{M}} \leq 1$ or $\Gamma^{\mathcal{M}} < U$.*

Proof: According to Lemma 5, for a sub-cone $\mathcal{M}_{min} = \{\boldsymbol{x} | \boldsymbol{x} = \boldsymbol{x}^0 + \boldsymbol{U}\boldsymbol{t}, \boldsymbol{t} \geq \boldsymbol{0}\}$, we define θ_i and $\boldsymbol{x}_i$ such that $\boldsymbol{x}_i = \boldsymbol{x}^0 + \theta_i \boldsymbol{u}^i \in \partial\mathcal{C}(U-\epsilon)$ where ϵ is a small number.

Note that $\mathcal{M}_{min}$ is a sub-cone of the initial cone $\mathcal{M}^0$. According to Lemma 4, on each edge of $\mathcal{M}^0$, there must be one vertex of $\mathcal{D}$. By Lemma 7, we have, on each edge of $\mathcal{M}_{min}$, there must be one vertex of $\mathcal{D}$. Because $\mathcal{D} \cap \mathcal{M}_{min}$ has $|\boldsymbol{x}|+1$ vertexes, these vertexes (excluding $\boldsymbol{x}_0$) must be the vertexes on the edges of $\mathcal{M}_{min}$. We denote these $|\boldsymbol{x}|$ vertexes (excluding $\boldsymbol{x}_0$) as $\boldsymbol{x}_1^D, \boldsymbol{x}_2^D, ..., \boldsymbol{x}_{|\boldsymbol{x}|}^D$ such that $\boldsymbol{x}_i^D$ and $\boldsymbol{x}_i$ are on the same edge.

In Step 8, after solving (**LP-**$\mathcal{DM}$) and obtaining $\boldsymbol{x}^{\mathcal{M}}$, $\boldsymbol{x}^{\mathcal{M}}$ must be one of $\boldsymbol{x}_1^D, \boldsymbol{x}_2^D, ..., \boldsymbol{x}_{|\boldsymbol{x}|}^D$, because there is no other vertex in $\mathcal{D} \cap \mathcal{M}_{min}$. If $\psi^{\mathcal{M}} > 1$, since $\boldsymbol{x}_i^D$ and $\boldsymbol{x}_i$ are on the same edge and $\boldsymbol{x}_i \in \partial\mathcal{C}(U-\epsilon)$, we must have $\boldsymbol{x}_i^D \notin \mathcal{C}(U-\epsilon)$. Thus $\Gamma(\boldsymbol{x}_i^D) < \Gamma(\boldsymbol{x}_i) = U - \epsilon$. By Proposition 4, we have $\Gamma^{\mathcal{M}} \leq \Gamma(\boldsymbol{x}_i^D) < U - \epsilon < U$. □

In each iteration, after obtaining a new vertex in Step 8, we will have three cases. By Lemma 6, the first case confirms that the current cone is sub-optimal, thus we remove this cone; the second case obtains a strictly better solution, and we restart the procedure; the third case splits the current cone.

Since the numbers of constraints and variables are finite, the polyhedron $\mathcal{D}$ has finite vertexes. Thus the number of different minimum sub-cones is also finite. By Proposition E.1, for a minimum sub-cone, we either remove it or obtain a strictly better solution (a new vertex). Hence Algorithm 1 terminates after finitely many steps.

Finally, we prove that, when the algorithm terminates, the returned solution is ϵ-optimal. In Step 23, we split the current cone into $|\boldsymbol{x}|$ sub-cones. By Lemma 7, such splitting is exhaustive. According to the algorithm procedure, if no better solution is found, all sub-cones will be checked; if a better solution is found, the procedure will be restarted. In Step 7, we compute θ_i by $\boldsymbol{x}^0 + \theta_i \boldsymbol{u}^i \in \partial\mathcal{C}(U-\epsilon)$. And the algorithm can only terminate in Step 21 when $\mathcal{P} = \emptyset$, i.e., all sub-cones are checked. Hence, if the algorithm terminates, i.e., all sub-cones are removed, we cannot find a better solution whose objective value is smaller than $U - \epsilon$. In other words, the maximum gap between U and the optimal cost is at most ϵ. □

Proof of Proposition 6: Suppose there exists some $a^{n}_{R,k}$ such that $a^{n}_{R,k} > \max\{a^{n}_{R,k-1}, a^{n-1}_{R,k}\}$ in the optimal solution $\boldsymbol{a}_R$. We can find another solution $\boldsymbol{a}_R{}'$ such that $a^{n}_{R,k}{}' = \max\{a^{n}_{R,k-1}, a^{n-1}_{R,k}\}$ for k,n and $\boldsymbol{a}_R{}' = \boldsymbol{a}_R$ for other entries. All the constraints are still satisfied. Since $k > 1$ and $n > 1$, the objective value does not change. So there must be an optimal solution $\boldsymbol{a}^*_R$ such that $a^{n*}_{R,k} = \max\{a^{n*}_{R,k-1}, a^{n-1*}_{R,k}\}$. □

Proof of Proposition 7:

1. Because $\boldsymbol{a}^*_R$ must be a feasible schedule to (**CP**), we have $\Gamma(\boldsymbol{a}^*) \le \Gamma(\boldsymbol{a}^*_R)$.

To show $\Gamma(\boldsymbol{a}^*_R) \le \Gamma_R(\boldsymbol{a}^*_R)$, we note that the objective of (**CP**) contains two more negative terms, namely,

$$-\frac{1}{|\Omega|}\sum_{\omega\in\Omega}\sum_{k=1}^{K}\left[(C_{W,k}-C_{L,k})\sum_{n=2}^{N}\max\{a^n_k - s^{n-1}_k(\omega) - \delta^{n-1}_k(\omega), 0\}\right]$$

and

$$-\frac{1}{|\Omega|}\sum_{\omega\in\Omega}\sum_{n=1}^{N}\left[(C^n_I - C^n_R)\sum_{k=2}^{K}\max\{a^n_k - s^n_{k-1}(\omega) - \delta^n_{k-1}(\omega), 0\}\right],$$

than that in (**APR**). Hence, for any given solution, the objective value of (**CP**) is no larger than that of (**APR**).

Let $(\boldsymbol{a}^*_R, \boldsymbol{s}^*_R(\omega), \boldsymbol{f}^*_R(\omega), \omega\in\Omega)$ be the optimal solution to (**APR**). For a scenario $\omega\in\Omega$, recall that $\boldsymbol{s}(\omega|\boldsymbol{a})$ and $\boldsymbol{f}(\omega|\boldsymbol{a})$ denote the service start time and finish time under a given schedule $\boldsymbol{a}$ in the original model, i.e., $\boldsymbol{s}(\omega|\boldsymbol{a})$ and $\boldsymbol{f}(\omega|\boldsymbol{a})$ are calculated by $s^n_k(\omega|\boldsymbol{a}) = \max\{a^n_k, s^n_{k-1}(\omega) + \delta^n_{k-1}(\omega), s^{n-1}_k(\omega) + \delta^{n-1}_k(\omega)\}$ and $f^n(\omega|\boldsymbol{a}) = \max\{T^n, s^n_K(\omega) + \delta^n_K(\omega)\}$. We can show that $\boldsymbol{s}^*_R(\omega) = \boldsymbol{s}(\omega|\boldsymbol{a}^*_R)$ and $\boldsymbol{f}^*_R(\omega) = \boldsymbol{f}(\omega|\boldsymbol{a}^*_R)$. To see this, suppose otherwise. If there were an optimal solution to (**APR**) such that $s^{n,*}_{R,k}(\omega) > \max\{a^n_k, s^{n,*}_{R,k-1}(\omega) + \delta^n_{k-1}(\omega), s^{n-1,*}_{R,k}(\omega) + \delta^{n-1}_k(\omega)\}$, then we can set $s^{n,*}_{R,k}(\omega) = \max\{a^n_k, s^{n,*}_{R,k-1}(\omega) + \delta^n_{k-1}(\omega), s^{n-1,*}_{R,k}(\omega) + \delta^{n-1}_k(\omega)\}$ to decrease the objective value without violating the constraints. Similarly, suppose there were an optimal solution to (**APR**) such that $f^{n,*}_R(\omega) > \max\{T^n, s^{n,*}_{R,K}(\omega) + \delta^n_K(\omega)\}$, then we can set $f^{n,*}_R(\omega) = \max\{T^n, s^{n,*}_{R,K}(\omega) + \delta^n_K(\omega)\}$ to decrease the objective value without violating the constraints. We can do this for all $s^{n,*}_{R,k}(\omega)$ and $f^{n,*}_R(\omega)$, and finally we will have $s^{n,*}_{R,k}(\omega) = \max\{a^n_k, s^{n,*}_{R,k-1}(\omega) + \delta^n_{k-1}(\omega), s^{n-1,*}_{R,k}(\omega) + \delta^{n-1}_k(\omega)\}$ and $f^{n,*}_R(\omega) = \max\{T^n, s^{n,*}_{R,K}(\omega) + \delta^n_K(\omega)\}$ for all k,n,ω, indicating that $\boldsymbol{s}^*_R(\omega) = \boldsymbol{s}(\omega|\boldsymbol{a}^*_R)$ and $\boldsymbol{f}^*_R(\omega) = \boldsymbol{f}(\omega|\boldsymbol{a}^*_R)$. Because $(\boldsymbol{a}^*, \boldsymbol{s}(\omega|\boldsymbol{a}^*), \boldsymbol{f}(\omega|\boldsymbol{a}^*), \omega\in\Omega)$ must be a feasible solution to (**APR**) and $(\boldsymbol{a}^*_R, \boldsymbol{s}(\omega|\boldsymbol{a}^*_R), \boldsymbol{f}(\omega|\boldsymbol{a}^*_R), \omega\in\Omega) = (\boldsymbol{a}^*_R, \boldsymbol{s}^*_R(\omega), \boldsymbol{f}^*_R(\omega), \omega\in\Omega)$ is the optimal solution to (**APR**), we have $\Gamma_R(\boldsymbol{a}^*_R) \le \Gamma_R(\boldsymbol{a}^*)$.

So we have $\Gamma(\boldsymbol{a}^*) \le \Gamma(\boldsymbol{a}^*_R) \le \Gamma_R(\boldsymbol{a}^*_R) \le \Gamma_R(\boldsymbol{a}^*)$.

2. The inequality chain above suggests that $\Gamma(\boldsymbol{a}^*_R) - \Gamma(\boldsymbol{a}^*) \le \Gamma_R(\boldsymbol{a}^*) - \Gamma(\boldsymbol{a}^*)$.

In addition, we have $\Gamma_R(\boldsymbol{a}^*) - \Gamma(\boldsymbol{a}^*) =$

$$\begin{aligned}
&\frac{1}{|\Omega|}\sum_{\omega\in\Omega}\left[\sum_{k=1}^{K}(C_{W,k}-C_{L,k})\sum_{n=2}^{N}(a^{n*}_k - s^{n-1}_k(\omega) - \delta^{n-1}_k(\omega))^+ + \sum_{n=1}^{N}(C^n_I - C^n_R)\sum_{k=2}^{K}(a^{n*}_k - s^n_{k-1}(\omega) - \delta^n_{k-1}(\omega))^+\right]\\
\le&\frac{1}{|\Omega|}\sum_{\omega\in\Omega}\left[\sum_{k=1}^{K}(C_{W,k}-C_{L,k})\sum_{n=2}^{N}(a^{n*}_k - a^{n-1*}_k - \delta^{n-1}_k(\omega))^+ + \sum_{n=1}^{N}(C^n_I - C^n_R)\sum_{k=2}^{K}(a^{n*}_k - a^{n*}_{k-1} - \delta^n_{k-1}(\omega))^+\right]\\
\le&\sum_{k=1}^{K}(C_{W,k}-C_{L,k})\sum_{n=2}^{N}(a^{n*}_k - a^{n-1*}_k) + \sum_{n=1}^{N}(C^n_I - C^n_R)\sum_{k=2}^{K}(a^{n*}_k - a^{n*}_{k-1})\\
\le&\max_{k,n}\{C_{W,k}-C_{L,k}, C^n_I - C^n_R\}\left[\sum_{k=1}^{K}\sum_{n=2}^{N}(a^{n*}_k - a^{n-1*}_k) + \sum_{k=2}^{K}\sum_{n=1}^{N}(a^{n*}_k - a^{n*}_{k-1})\right]\\
=&\max_{k,n}\{C_{W,k}-C_{L,k}, C^n_I - C^n_R\}\left[\sum_{k=1}^{K}(a^{N*}_k - a^{1*}_k) + \sum_{n=1}^{N}(a^{n*}_K - a^{n*}_1)\right]\\
=&\max_{k,n}\{C_{W,k}-C_{L,k}, C^n_I - C^n_R\}\left[\sum_{k=2}^{K}a^{N*}_k - \sum_{k=1}^{K-1}a^{1*}_k + \sum_{n=2}^{N}a^{n*}_K - \sum_{n=1}^{N-1}a^{n*}_1\right]
\end{aligned}$$

$$\leq \max_{k,n}\{C_{W,k}-C_{L,k},C_I^n-C_R^n\}\left[\sum_{k=2}^{K}a_k^{N*}+\sum_{n=2}^{N}a_K^{n*}\right]$$
$$\leq \max_{k,n}\{C_{W,k}-C_{L,k},C_I^n-C_R^n\}(N+K-2)\max_{n}\{T^n\},$$

where the first inequality follows from that $a_{k-1}^{n*}\leq s_{k-1}^{n}(\omega)$ and $a_k^{n-1*}\leq s_k^{n-1}(\omega)$, $\forall k,n,\omega$; the second one follows from that $\delta_k^n(\omega)\geq 0$, $\forall k,n,\omega$; the third one follows from that $C_{W,k}-C_{L,k}\leq \max_{k,n}\{C_{W,k}-C_{L,k},C_I^n-C_R^n\}$ and $C_I^n-C_R^n\leq \max_{k,n}\{C_{W,k}-C_{L,k},C_I^n-C_R^n\}$, $\forall k,n$; the fourth one follows from that $a_k^n\geq 0$, $\forall k,n$; and the last one follows from that $a_k^n\leq \max\{T^n\}$, $\forall k,n$. This completes the proof. □

Proof of Corollary D.1:

The proof is similar to that of Proposition 7. We will be brief. First, because $\boldsymbol{a}_R^*$ must be a feasible schedule to (**CP.ext**), we have $\mathit{\Gamma}(\boldsymbol{a}^*)\leq \mathit{\Gamma}(\boldsymbol{a}_R^*)$. Also, note that the objective function of (**CP.ext**) has two more negative terms than that in (**APR.ext**). It follows that $\mathit{\Gamma}(\boldsymbol{a}_R^*)\leq \mathit{\Gamma}_R(\boldsymbol{a}_R^*)$.

Let $(\boldsymbol{a}_R^*,\boldsymbol{s}_R^*(\omega),\boldsymbol{f}_R^*(\omega),\omega\in\Omega)$ be the optimal solution to (**APR.ext**). For a scenario $\omega\in\Omega$, recall that $\boldsymbol{s}(\omega|\boldsymbol{a})$ and $\boldsymbol{f}(\omega|\boldsymbol{a})$ denote the service start time and finish time under a given schedule $\boldsymbol{a}$ in the original model, i.e., $\boldsymbol{s}(\omega|\boldsymbol{a})$ and $\boldsymbol{f}(\omega|\boldsymbol{a})$ are calculated by $s_k^n(\omega)=\max\{a_k^n,s_{k-1}^n(\omega)+\delta_{k-1}^n(\omega),s_k^{n-1}(\omega)+\delta_k^{n-1}(\omega)\}$ and $f^n(\omega|\boldsymbol{a})=\max\{T^n,s_K^n(\omega)+\delta_K^n(\omega)\}$. We can show that $\boldsymbol{s}_R^*(\omega)=\boldsymbol{s}(\omega|\boldsymbol{a}_R^*)$ and $\boldsymbol{f}_R^*(\omega)=\boldsymbol{f}(\omega|\boldsymbol{a}_R^*)$. To see this, suppose there is an optimal solution to (**APR.ext**) such that $s_{R,k}^{n,*}(\omega)>\max\{a_k^n,s_{R,k-1}^{n,*}(\omega)+\delta_{k-1}^n(\omega),s_{R,k}^{n-1,*}(\omega)+\delta_k^{n-1}(\omega)\}$, then we can set $s_{R,k}^{n,*}(\omega)=\max\{a_k^n,s_{R,k-1}^{n,*}(\omega)+\delta_{k-1}^n(\omega),s_{R,k}^{n-1,*}(\omega)+\delta_k^{n-1}(\omega)\}$ to decrease the objective value without violating the constraints. Similarly, suppose there is an optimal solution to (**APR.ext**) such that $f_R^{n,*}(\omega)>\max\{T^n,s_{R,K}^{n,*}(\omega)+\delta_K^n(\omega)\}$, then we can set $f_R^{n,*}(\omega)=\max\{T^n,s_{R,K}^{n,*}(\omega)+\delta_K^n(\omega)\}$ to decrease the objective value without violating the constraints. We can do this for all $s_{R,k}^{n,*}(\omega)$ and $f_R^{n,*}(\omega)$, and finally we will have $s_{R,k}^{n,*}(\omega)=\max\{a_k^n,s_{R,k-1}^{n,*}(\omega)+\delta_{k-1}^n(\omega),s_{R,k}^{n-1,*}(\omega)+\delta_k^{n-1}(\omega)\}$ and $f_R^{n,*}(\omega)=\max\{T^n,s_{R,K}^{n,*}(\omega)+\delta_K^n(\omega)\}$ for all k,n,ω, indicating $\boldsymbol{s}_R^*(\omega)=\boldsymbol{s}(\omega|\boldsymbol{a}_R^*)$ and $\boldsymbol{f}_R^*(\omega)=\boldsymbol{f}(\omega|\boldsymbol{a}_R^*)$. As $(\boldsymbol{a}^*,\boldsymbol{s}(\omega|\boldsymbol{a}^*),\boldsymbol{f}(\omega|\boldsymbol{a}^*),\omega\in\Omega)$ must be a feasible solution to (**APR.ext**), and $(\boldsymbol{a}_R^*,\boldsymbol{s}(\omega|\boldsymbol{a}_R^*),\boldsymbol{f}(\omega|\boldsymbol{a}_R^*),\omega\in\Omega)=(\boldsymbol{a}_R^*,\boldsymbol{s}_R^*(\omega),\boldsymbol{f}_R^*(\omega),\omega\in\Omega)$ is the optimal solution to (**APR**). Thus $\mathit{\Gamma}_R(\boldsymbol{a}_R^*)\leq \mathit{\Gamma}_R(\boldsymbol{a}^*)$.

So we have

$$\mathit{\Gamma}(\boldsymbol{a}^*)\leq \mathit{\Gamma}(\boldsymbol{a}_R^*)\leq \mathit{\Gamma}_R(\boldsymbol{a}_R^*)\leq \mathit{\Gamma}_R(\boldsymbol{a}^*).$$

It follows that

$$\mathit{\Gamma}(\boldsymbol{a}_R^*)-\mathit{\Gamma}(\boldsymbol{a}^*)\leq \mathit{\Gamma}_R(\boldsymbol{a}^*)-\mathit{\Gamma}(\boldsymbol{a}^*).$$

In addition, we have $\mathit{\Gamma}_R(\boldsymbol{a}^*)-\mathit{\Gamma}(\boldsymbol{a}^*)\leq$

$$\frac{1}{|\Omega|}\sum_{\omega\in\Omega}\left[\sum_{k=1}^{K}(C_{W,k}-C_{L,k})\sum_{n=2}^{N}(a_k^{n*}-a_k^{n-1*}-\sigma_k^{n-1}(\omega)\delta_k^{n-1}(\omega))^+ +\sum_{n=1}^{N}(C_I^n-C_R^n)\sum_{k=2}^{K}(a_k^{n*}-a_{k-1}^{n*}-\sigma_{k-1}^{n}(\omega)\delta_{k-1}^{n}(\omega))^+\right]$$
$$\leq \sum_{k=1}^{K}(C_{W,k}-C_{L,k})\sum_{n=2}^{N}(a_k^{n*}-a_k^{n-1*})+\sum_{n=1}^{N}(C_I^n-C_R^n)\sum_{k=2}^{K}(a_k^{n*}-a_{k-1}^{n*})$$
$$\leq \max_{k,n}\{C_{W,k}-C_{L,k},C_I^n-C_R^n\}(N+K-2)\max_{n}\{T^n\},$$

where the first inequality follows from that $\sigma_k^n(\omega)\leq 1$, $a_{k-1}^{n*}\leq s_{k-1}^n(\omega)$ and $a_k^{n-1*}\leq s_k^{n-1}(\omega)$, $\forall k,n,\omega$; the second one follows from that $\delta_k^n(\omega)\geq 0$, $\forall k,n,\omega$; and the last one follows from a similar argument in the proof of Proposition 7. This completes the proof. □

Appendix References

Hajek, Bruce. 1985. Extremal splittings of point processes. *Mathematics of Operations Research* **10**(4) 543–556.