Managing Appointment-based Services in the Presence of Walk-in Customers

Shan Wang
Antai College of Economics and Management, Shanghai Jiao Tong University, Shanghai 200030, China, wangshan_731@sjtu.edu.cn

Nan Liu
Department of Health Policy and Management, Mailman School of Public Health, Columbia University, New York, NY 10032, nl2320@columbia.edu

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

Despite the prevalence and significance of walk-ins in healthcare, we know relatively little about how to plan and manage the daily operations of a healthcare facility that accepts both scheduled and walk-in patients. In this paper, we develop the first optimization model to determine the optimal appointment schedule in the presence of potential walk-ins. We show that the natural formulation of the problem, which is difficult to deal with directly, can be reformulated as a two-stage stochastic integer program with a simple and tractable structure. This modeling framework is very flexible, and can accommodate random service times, patient-dependent and time-dependent no-show behaviors as well as patient preferences. We demonstrate that, with walk-ins, the structures of the optimal schedule are fundamentally different from those identified in the earlier literature which does not consider walk-ins. Using data from practice, we predict a significant cost reduction (20% reduction on average and max 59%) if providers were to switch from current practice to our proposed schedules. Though our work is motivated by healthcare, our models and insights can be applied to general appointment-based services in the presence of random walk-in customers.

Key words: service operations management, healthcare, appointment scheduling, walk-ins, optimization

History:

1. Introduction

Making an appointment is a common way for customers to get service in many industries. Walk-in customers without appointments (or “walk-ins” for short), however, are often welcome and accepted as well. Providing service to walk-ins benefits a firm in a range of ways, such as increasing revenues, enlarging the customer pool and building a good business image. To name a few examples,
banks accept walk-ins for more business; hotels seldom reject requests from walk-ins if rooms are still available; restaurants rely on walk-ins to build the word of mouth; beauty salons always try to make walk-ins become their regular clients; tech support accepts walk-ins to attract more customers. As walk-ins arrive spontaneously without advance notice, they may interrupt the firm’s daily operations, in particular the service of scheduled customers who have set specific arrival times for services.

One industry that often sees the conflict between serving walk-ins and scheduled customers is healthcare. In the outpatient care setting, walk-in patients without appointments are usually accepted and constitute a major stream of the customers. In the US, walk-ins can range from 10% to 60% of the total daily visits to primary care practices; see, e.g., Shonick and Klein (1977), Moore et al. (2001), and Cayirli et al. (2008). In the UK, 63% of genitourinary medicine (GUM) clinics operate both appointment-based and walk-in services (Djuretic et al. 2001). Su and Shih (2003) find that on average 72% of the total daily visits to healthcare organizations in Taiwan are walk-ins. This percentage is even higher in mainland China, in which walking in is the traditional way of getting healthcare service (Barber et al. 2014).

Despite the prevalence and significance of walk-ins in healthcare, we know relatively little about how to plan and manage the daily operations of a healthcare facility in the presence of walk-ins. Current practice of outpatient care deals with walk-ins by setting up daily schedule templates (also called “appointment books”), which specify when to schedule an appointment and when, if ever, to intentionally leave open in anticipation for walk-ins. However, there is a lack of scientific understanding on how to set up such a daily template to accommodate both scheduled patients and potential walk-ins. Most extant literature develops models and insights that can only be applied to an environment free of walk-ins; managing a practice that accepts walk-ins requires fundamentally different tools and guidelines (see a detailed discussion in Section 2). Without careful planning and adjusting for walk-ins, daily service operations may be interrupted, resulting in long patient waits, provider overtime work and ultimately poor service quality.

The negative and potentially serious impact on the organization due to not carefully considering walk-ins becomes evident when we interact and collaborate with two large outpatient care systems in New York City (NYC). The first one is a community health center that provides comprehensive medical and dental care to the Central Harlem and Washington Heights areas for over 25 years. Being a Federally Qualified Health Center (FQHC), this facility has to serve all patients regardless of their ability to pay (Rural Assistance Center 2013); as a result, more than 15% of the total patient visits to this center are walk-ins (see detailed data presentation in Section 3). However, the administrative team of this center has informed us that walk-ins are “believed to be the main reason for long patient waits”. The other organization we interact with is a large community healthcare
network, made up of 11 FQHCs located across different boroughs of NYC. One of their primary care physicians told us that “I know there are always many walk-ins at 10am, but I can’t take them. I have appointments [at that time]” (Berman 2016). Undoubtedly, walk-ins have presented a significant challenge in running both organizations, and how to delivery high quality care services in the presence of these uncertain walk-ins becomes a critical operational issue faced by their administrative teams.

In this paper, we take a data-driven approach to develop analytic models to inform the design of daily schedule templates in outpatient care practices where both scheduled and walk-in patients are accepted. Using a large dataset obtained from our first collaborating organization, we find that patient walk-in processes and patterns vary across providers even in one practice. More importantly, walk-ins may not arrive according to the classic (time-inhomogeneous) Poisson process, as often found in the previous literature, e.g., Kim and Whitt (2014). In particular, the “zero-event” probability, i.e., the chance that no walk-ins arrive in a short time period, may be too large for the Poisson distribution. Motivated by these empirical findings, we develop optimization models that can accommodate general arrival patterns of walk-ins. Specifically, we consider a generic clinic session for a single provider. The clinic session consists of $T > 0$ appointment slots. The provider needs to schedule some preset number of patients, say $N_s$ patients, into these $T$ slots. Throughout the day, a random number of walk-ins may arrive, according to some general arrival process, for services. We are concerned with assigning the $N_s$ patients to the $T$ slots, in anticipation for potential walk-ins that may arrive over time. The objective is to minimize the expected total cost due to patient waiting, provider idling and overtime.

Another important factor to consider when designing schedule templates in healthcare is patient no-show behavior. Patient no-shows occur when patients miss their booked appointments without early notice or cancellation. Patient no-show rate can range from 1% to 60% depending on practice and patient profiles, and not accounting for patient no-shows may lead to significant operational inefficiency; see Cayirli and Veral (2003), Gupta and Denton (2008) and Liu (2016) for detailed discussions on the phenomenon and impact of patient no-shows.

Depending on the magnitude of no-show rates, we develop two sets of analyses. If patient no-show rate is negligible, we characterize the structure of the optimal schedule, which is completely different from that when walk-ins are not present. This structural result drastically reduces the search space of the optimal schedule, and enables an efficient enumeration search for practically-sized problems.

When patient no-shows become more significant, the resulting optimization becomes much more difficult, as the optimal schedule does not show a (clear) structure (see Section 5). We use the “natural” definition of decision variables to formulate a two-stage optimization model in which
the second stage evaluates the objective given a schedule and the first stage searches for the optimal schedule. This natural formulation is nonlinear and not easy to solve. However, via a simple and yet innovative variable transformation by modeling individual patient’s no-show behavior, we are able to reformulate the problem into a stochastic integer programming problem where the first stage is a 0-1 integer program and the second stage is a pure linear program. This new reformulation is amenable to off-the-shelf optimization software. In addition, this reformulation has a special structure that makes it quite easy to (1) check the optimality of a solution without fully solving the whole problem, and (2) identify redundant constraint sets. Leveraging these nice properties, we develop efficient solution methods which are shown to be much faster than the standard optimization software in solving large-scale instances (see Section 6.1).

Our contributions in this work can be summarized as follows. We develop the first optimization model to determine the optimal appointment schedule in the presence of potential walk-ins. Our modeling framework is very flexible, and can accommodate random service times (of certain distributions), patient-dependent and time-dependent no-show rates as well as patient preferences (see Section 7). We demonstrate that, with walk-ins, the structures of the optimal schedule are fundamentally different from those identified in the previous literature which does not consider walk-ins. When patient no-shows are of less concern, we develop effective solution procedures to achieve optimality. When patient no-shows become non-negligible, we propose an innovative variable transformation to reformulate the original problem into a stochastic integer programming model with simple structures which are much more tractable. To the best of our knowledge, we are the first to propose such a reformulation, which may have broader applications in other modeling contexts. Besides these analytical contributions, our empirical investigation of patient walk-ins also contributes to the relatively scant empirical literature on customer arrivals by revealing new temporal patterns and models for customer demand.

The remainder of this paper is organized as follows. Section 2 reviews the relevant literature. In section 3, we present the empirical analysis of walk-in patterns. Section 4 develops our basic scheduling model with random walk-ins and investigates the properties of the optimal schedule. In section 5, we incorporate patient no-show behavior into the basic model and develop the solution approaches. We report our numerical results and case study in Section 6. Section 7 discusses several model extensions, and Section 8 provides concluding remarks. The proofs of all the analytical results can be found in the Online Appendix.

2. Literature Review
Our work draws upon several streams of literature. First, our work is closely related to the (outpatient) appointment scheduling literature that investigates how to schedule patients over time in a
day. A common theme of this literature is to develop mathematical programming models to optimize the tradeoff between patient in-clinic waiting and provider utilization. Two types of decision variables have been considered. The first type of decision scenarios is concerned with the exact appointment time for each patient (decision variables are continuous); see, e.g., Denton and Gupta (2003), Hassin and Mendel (2008), Begen and Queyranne (2011), Kong et al. (2013), Ge et al. (2013), Chen and Robinson (2014) and Jiang et al. (2015). The second type of decision scenario, like ours, divides a day into a certain number of appointment slots, and determines the number of patients assigned to each slot (decision variables are integers); see, e.g., Kaandorp and Koole (2007), Robinson and Chen (2010), LaGanga and Lawrence (2012) and Zacharias and Pinedo (2014). These previous studies have considered a variety of uncertainties in practice that may affect the design of appointment templates (such as patient no-shows and random service times). However, none of the above work explicitly considers walk-ins, an important phenomenon in healthcare as discussed earlier. Our work complements and advances this stream of literature by proposing new models and solution approaches to optimize the appointment schedule in anticipation for random walk-ins.

An early group of literature develops simulation models to analyze and propose scheduling patterns, and some considers the impact of walk-ins; see, e.g., Fetter and Thompson (1966), Rising et al. (1973), Cayirli et al. (2006), Cayirli et al. (2008) and Cayirli et al. (2012). In contrast, we adopt a mathematical programming approach to obtain the optimal schedule. Two recent modeling papers also consider walk-ins, but have a completely different focus from ours; see Qu et al. (2015) and Wiesche et al. (2016). They both study how best to use existing service capacity when walk-ins arise (e.g., whether to accept a walk-in or not), rather than to determine the schedule template.

In the appointment scheduling literature mentioned above, our work is most related to Kaandorp and Koole (2007), Robinson and Chen (2010) and Zacharias and Pinedo (2014). All these studies, including ours, treat the number of patients assigned to each appointment slot as the decision variable. In Kaandorp and Koole (2007), patients’ no-show behaviors are homogeneous and provider service times are exponentially distributed. They develop a local search procedure for the optimal schedule. Different from Kaandorp and Koole (2007), Robinson and Chen (2010) and Zacharias and Pinedo (2014) both assume deterministic service times for providers. Under this assumption, Robinson and Chen (2010) identify an important property – the “No Hole” property – for the optimal schedule (more on this below). Zacharias and Pinedo (2014) consider both offline and online scheduling, and develop structural properties and effective heuristics for the optimal schedule.

Our work departs from the three studies above in several important ways. First, we explicitly take into account potential walk-ins during the day and we allow the walk-in process to be general. We demonstrate that the resulting optimization problem is much more complicated; and those elegant properties that hold without walk-ins (e.g., the “No Hole” property) do not hold any
more when walk-ins are accepted. Second, the previous literature solves for the optimal schedule either using local search or via enumeration (after characterizing the structural results). We are, however, able to provide the first two-stage stochastic integer programming formulation for the appointment scheduling problem with both walk-ins and no-shows present. This formulation not only is amenable to many standard mixed integer programming solvers, but also has a special structure which allows us to develop a unified solution approach proven to be highly effective in numerical experiments. Third, our modeling framework and solution approaches are very flexible, and can accommodate random service times, heterogeneous patient no-shows and preferences.

Our work is also connected to, but differs significantly from, the literature in service operations management that deals with walk-in customers. For instance, Bertsimas and Shioda (2003) develop methods to dynamically decide when, if at all, to seat an incoming party during the day of operations of a restaurant. This is an online decision problem in the restaurant industry, while our work focuses an offline decision to determine the best schedule in a doctor’s office. Alexandrov and Lariviere (2012) develop a game-theoretic model to study whether reservations are recommended for restaurants where walk-in customers are often allowed. Bitran and Gilbert (1996) and Gans and Savin (2007) study reservation management problem with uncertain walk-in customers for hotels and rental firms, respectively. The last three studies focus on capacity level decisions (e.g., how much capacity to reserve for walk-ins), rather than within-day operations investigated by us.

3. Empirical Analysis of Walk-in Arrival Patterns

While the previous literature has a rich documentation on the volume of walk-ins (see, e.g., Shonick and Klein 1977, Moore et al. 2001, Cayirli et al. 2008), relatively little is known about the temporal pattern of walk-in arrivals. In this section, we use a dataset obtained from a large community health center located in NYC to study the temporal pattern of walk-ins.

3.1. Data

The data was extracted from the EMR (Electronic Medical Record) system of our collaborating health center. This center provides comprehensive medical and dental care to the local community, and serves more than 25,000 patient visits every year. The dataset spans over 3 years ranging from Jan. 2011 to Jan. 2014, and contains 67847 valid records of patient visits. In these valid records, more than 15% (10402) are walk-ins. There are 38 providers (including physicians and nurse practitioners) in the dataset; some providers have more than 50% of the patients they see as walk-ins. In this center, walk-ins are accepted throughout the office hour. When analyzing this dataset, we focus on three specialties, Nurse Practitioner, Internal Medicine and Pediatrics, which serve more than 80% of the walk-in visits (the numbers of records of them are 5076, 2669 and 1128 respectively). Then we choose 6 providers who have the most walk-in records (4 Nurse Practitioners, 1 Internist and 1 Pediatrician).
3.2. Statistical Analysis Framework

To study the arrival patterns of walk-ins, we adopt a Poisson Regression framework to model the number of walk-ins in each hour. Specifically, for each of the six providers, we estimate and compare three regression models below (from simple to more comprehensive). Model 1 is a classic Poisson regression model where \( Y_t \), the number of walk-ins in hour \( t \), has a Poisson distribution with mean \( \lambda_t \), which depends on the hour \( t \). That is,

\[
Pr(Y_t = k) = \frac{\lambda_t^k e^{-\lambda_t}}{k!}, \quad k = 0, 1, 2, \ldots
\]  

(1)

Using the logarithm as the canonical link function, Model 1 is specified as follows.

\[
\log(\lambda_t) = \gamma_1 + \gamma_2 x_2 + \cdots + \gamma_T x_T,
\]  

(Model 1)

where \( x_i \) is a dummy variable which takes value 1 if \( t = i \) and value 0 otherwise, \( i = 2, 3, \ldots, T \) (note that we do not have \( x_1 \) in Model 1, because hour 1 is the base category whose effect is captured by \( \gamma_1 \)). In Model 1, \( \gamma \)'s are the statistical parameters we will estimate, and the hourly arrival rates can then be estimated as \( \lambda_1 = e^{\gamma_1} \) and \( \lambda_t = e^{\gamma_1 + \gamma_t} \) for \( t > 1 \).

A close look at our data reveals that for some of the providers, there are an excessive number of zeros in hourly arrivals, which may make Model 1 not a good fit. To address this problem of excess zeros, we consider zero-inflated Poisson regression models (Lambert 1992), which first determine whether there are zero events or any events, and then use a Poisson distribution to determine the number of events if there are any. That is, the number of walk-ins in hour \( t \) is modeled as below.

\[
Pr(Y_t = k) = \begin{cases} 
  a + (1-a)e^{-\lambda_t} & \text{if } k = 0, \\
  (1-a)\frac{\lambda_t^k e^{-\lambda_t}}{k!} & \text{if } k > 0,
\end{cases}
\]  

(2)

where \( a \) is the zero-event probability and \( \lambda_t \) is the hourly arrival rate. Using the canonical link functions, the statistical specification of the above model can be written as follows.

\[
\log\left(\frac{a}{1-a}\right) = b \quad \text{and} \quad \log(\lambda_t) = \gamma_1 + \gamma_2 x_2 + \cdots + \gamma_T x_T,
\]  

(Model 2)

where \( x_i \) is defined as in Model 1. Under Model 2, \( a = 1 - \frac{1}{e^{\gamma_1 + \gamma_t}} \), \( \lambda_1 = e^{\gamma_1} \) and \( \lambda_t = e^{\gamma_1 + \gamma_t} \) for \( t > 1 \).

Model 2 assumes a constant zero probability \( a \). A more comprehensive model, however, is to specify that the zero probability also depends on time \( t \). That is, the number of walk-ins in hour \( t \) is modeled as below.

\[
Pr(Y_t = k) = \begin{cases} 
  a_t + (1-a_t)e^{-\lambda_t} & \text{if } k = 0, \\
  (1-a_t)\frac{\lambda_t^k e^{-\lambda_t}}{k!} & \text{if } k > 0,
\end{cases}
\]  

(3)

where \( a_t \) is the zero-event probability in hour \( t \). Its corresponding statistical specification is:

\[
\log\left(\frac{a_t}{1-a_t}\right) = b_1 + b_2 x_2 + \cdots + b_T x_T \quad \text{and} \quad \log(\lambda_t) = \gamma_1 + \gamma_2 x_2 + \cdots + \gamma_T x_T,
\]  

(Model 3)
where \( x_i \) is defined as in Model 1. Under Model 3, \( a_1 = 1 - \frac{1}{e^{\gamma_1+1}} \) and \( a_t = 1 - \frac{1}{e^{\gamma_1 + \gamma_t + 1}} \) for \( t > 1 \); 
\[ \lambda_1 = e^{\gamma_1} \text{ and } \lambda_t = e^{\gamma_1 + \gamma_t} \text{ for } t > 1. \]

These three models increase in their generality. To assemble the data for analysis, for each provider we count the number of patients who arrive between 30 minutes before an hour and 30 minutes after as those arriving for that hour. To arrive at the most parsimonious model that adequately describe the data, we conduct a series of statistical tests. Note that Model 2 is a reduced model of Model 3 (by specifying that \( a_t = a \)), we can use the likelihood ratio test to test if Model 3 makes a significant improvement over Model 2. Model 1 and Model 2, however, are not nested. So we use Vuong’s closeness test to test if Model 2 improves upon Model 1 significantly (Vuong 1989). For each provider, we adopt the simplest model, to which more complicated models cannot make a significant improvement, as our final model. We test the goodness-of-fit of the final model using the Chi-square test.

### 3.3. Empirical Results

Table 1 summaries the testing results of three fitted models for each provider. Providers’ initials are used to protect their confidentiality. It is important and interesting to note that all three models have appeared as the final model for some provider. Specifically, for providers GED, KNI and WAT, we find that the number of walk-ins in each slot follows the zero-inflated Poisson distribution rather than the classic Poisson distribution. For providers GED and KNI, the estimated zero event probabilities \( a \) are constant over time and they are both 0.14 in the final model, respectively. For provider WAT, the zero event probability depends on hour of day, and its estimated value is 0.99, 0.24, 0.42, 0.50, 0.44, 0.59, 0.39, 0.63, 0.33, 0.80 and 0.33 from 8am to 6pm, respectively. For providers ALD, GAR and LOK, we find that the Poisson distribution is appropriate to model the number of walk-ins arriving in each hour, though its mean varies over time.

<table>
<thead>
<tr>
<th>Provider</th>
<th>Specialty</th>
<th>Sample Size</th>
<th>Model 2 v.s. 3</th>
<th>Model 1 v.s. 2</th>
<th>Final Model</th>
<th>Goodness of Fit</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALD</td>
<td>Nurse Practitioner</td>
<td>1524</td>
<td>0.98</td>
<td>0.49(0.63)</td>
<td>Model 1</td>
<td>0.74</td>
</tr>
<tr>
<td>GAR</td>
<td>Nurse Practitioner</td>
<td>3403</td>
<td>0.99</td>
<td>-0.00(0.99)</td>
<td>Model 1</td>
<td>0.86</td>
</tr>
<tr>
<td>GED</td>
<td>Internal Medicine</td>
<td>1115</td>
<td>0.97</td>
<td>1.00(0.32)</td>
<td>Model 2</td>
<td>0.18</td>
</tr>
<tr>
<td>KNI</td>
<td>Nurse Practitioner</td>
<td>1729</td>
<td>0.42</td>
<td>1.48(0.14)</td>
<td>Model 2</td>
<td>0.70</td>
</tr>
<tr>
<td>LOK</td>
<td>Nurse Practitioner</td>
<td>1308</td>
<td>0.69</td>
<td>0.37(0.71)</td>
<td>Model 1</td>
<td>0.27</td>
</tr>
<tr>
<td>WAT</td>
<td>Pediatrics</td>
<td>3045</td>
<td>0.08</td>
<td>3.09(0.00)</td>
<td>Model 3</td>
<td>0.45</td>
</tr>
</tbody>
</table>

Note: (1) Providers ALD, LOK and WAT work from 8am to 6pm; GAR works from 8am to 4pm; GED works from 9am to 6pm; and KNI works from 8am to 5pm. (2) Column “Model 2 v.s. 3” shows the p-value of the likelihood ratio test. If \( p < 0.05 \), Model 3 makes a significant improvement over Model 2. (3) Column “Model 1 v.s. 2” shows the Vuong-test statistic value (p-value in brackets). A positive Vuong-test statistic suggests that Model 2 is closer to the true model. Unless \( p <= 0.5 \), we still choose Model 1 as the final model. (4) Column “Goodness of Fit” shows the p-value of the goodness-of-fit test (\( p > 0.05 \) indicates a good fit).

Figure 1 shows the expected number of hourly walk-ins for each provider. All providers except for GAR take a lunch break around 1pm (but still may have a few walk-ins at that time) and thus
we see a bimodal distribution of the arrival rates. In contrast, GAR does not take lunch break and goes home earlier; the walk-in rate to this provider shows a unimodal pattern over the day.

Our empirical results make an important implication that there is no one-size-fit-all model for a walk-in process, and it has to be individualized for each provider. There is relatively scant literature that examines the arrival pattern of walk-ins using empirical data. The extant limited literature unanimously suggests that the unscheduled walk-in process follows a (nonhomogeneous) Poisson process; see, e.g., Swartzman (1970) and Kim and Whitt (2014). Our work contributes to this literature by revealing new arrival patterns of walk-ins (i.e., zero-inflated Poisson process). While the previous appointment scheduling work that considers walk-ins (all are simulation-based to the best of our knowledge) has predominantly used Poisson process to model walk-in arrivals (see, e.g., Swisher et al. 2001, Cayirli et al. 2008), we suggest that appointment scheduling models should be able to accommodate general arrival patterns of walk-ins. We develop one such optimization model below.

4. Basic Model
In this section, we develop a basic appointment scheduling model with random walk-in patients. For now, we assume that all scheduled patients will show up at their appointment times. We will extend our modeling framework to incorporate patient no-show behavior in Section 5. Throughout, we will use lower-case (upper-case) Greek letters to denote random variables (calculated values), lower-case (upper-case) letters to denote variables (constants), and bold-faced lower-case (upper-case) letters to denote vectors (matrices). The dimensions of vectors or matrices should be evident from the context. We provide a summary of the notations in Table 7 of the Appendix.
Consider a generic clinic session for a single provider. In practice, the length of a clinic session is often measured by the number of appointment slots, and patients are scheduled to arrive at the beginning of these slots (patients are rarely scheduled to arrive in the middle of a slot). Following this convention, we consider a clinic session with $T$ appointment slots, and the provider needs to schedule $N_S$ patients in these slots. The number $N_S$ is set in advance based on the provider's expected clinic workload and other professional obligations (e.g., teaching, research and service).

Besides these scheduled patients, a random number of patients may walk in for services. For tractability, we assume that walk-in patients always arrive at the beginning of each appointment slot. (In reality, patients may arrive anytime within a slot. Our assumption above at most misjudges the wait time of a walk-in by half a slot, i.e., 10 minutes or so, and therefore will not misinterpret individual patient’s experience too much.) Let $\beta = (\beta_1, \beta_2, \ldots, \beta_T)$ be a random vector with support on non-negative integers, where $\beta_t$ represents the number of walk-ins arriving at the beginning of slot $t$. We assume that $\beta_t$’s are independent of each other, but they may draw from different probability distributions. Thus, the arrival pattern of walk-ins may depend on time $t$.

For now, we assume that the service time of each patient is exactly one appointment slot (normalized as one unit of time in our model). In clinical practice, especially in primary care, the provider usually can control her consultation time with patients to be within the allotted time by adjusting the conversation content and speed (Gupta and Denton 2008). Indeed, deterministic service time is a reasonable assumption commonly made in the appointment scheduling literature; see, e.g., Robinson and Chen (2010), LaGanga and Lawrence (2012) and Zacharias and Pinedo (2014). Nevertheless, as we will discuss in Section 7, our models and solution approaches can be easily extended to incorporate random service times with certain probability distributions.

In the setting above, we need to schedule $N_S$ patients into $T$ slots. Specifically, we need to decide the number of patients to be scheduled in each slot. Let $x = (x_1, x_2, \ldots, x_T)$ be our decision vector, in which $x_t$ is the number of patients scheduled at slot $t$. Following the previous literature, e.g., Robinson and Chen (2010) and Zacharias and Pinedo (2014), we assume that all scheduled patients are punctual for tractability.

A common optimization framework in the literature is to assign different cost rates to patient wait time, provider idle time and overtime; and then to minimize the expected total weighted cost with these cost rates serving as the weights. We follow this framework, but note that this cost structure can be slightly simplified in our model without loss of generality. To see that, let $C_S$ and $C_W$ be the waiting cost for a scheduled patient and a walk-in patient per appointment slot of time, respectively. Let $C_I$ and $C_O$ be the provider’s idle cost and overtime cost per appointment slot of time, respectively. For a given schedule, let $\Gamma_S$ and $\Gamma_W$ be the expected total wait time of
scheduled patients and that of walk-in patients, respectively. Let $\Gamma_I$ and $\Gamma_O$ be the expected idle time and overtime of the provider. Thus, the expected total weighted cost is

$$C_S \Gamma_S + C_W \Gamma_W + C_I \Gamma_I + C_O \Gamma_O.$$  \hfill (4)

Let $\Gamma_D$ be the expected duration from the beginning of the session to the departure time of the last patient. It is clear that $\Gamma_O = \Gamma_D - T$. Let $N_W$ be the expected number of walk-in patients, i.e., $N_W = \mathbb{E}(\sum_{t=1}^{T} \beta_t)$. Then, $N_S + N_W$ is the expected total consultation time that the provider spends with patients, and thus the difference between $\Gamma_D$ and $N_S + N_W$ is the expected idle time of the provider, i.e., $\Gamma_I = \Gamma_D - N_S - N_W$. We can rewrite the expected total weighted cost (4) as

$$C_S \Gamma_S + C_W \Gamma_W + C_I (\Gamma_D - N_S - N_W) + C_O (\Gamma_D - T).$$  \hfill (5)

As $N_S$, $N_W$ and $T$ are constants, they can be omitted from the optimization process. Let $C_D = C_I + C_O$, and normalize $C_S$ to be 1. The expected total weighted cost in our optimization objective can be simplified as follows,

$$\Gamma_S + C_W \Gamma_W + C_D \Gamma_D.$$  \hfill (6)

When deriving the optimal solution, we use (6) for simplicity; we will use (5) when the actual objective value is needed, such as calculating the expected total cost associated with a schedule.

To calculate $\Gamma_S$, $\Gamma_W$ and $\Gamma_D$, we first evaluate $\Pi_t(k)$, the probability of $k$ patients waiting for services at the beginning of $t$. Let $p_t(b)$ be the probability of $b$ walk-ins arriving at slot $t$, i.e., $p_t(b) = \Pr(\beta_t = b)$, and let $\bar{N}$ be a sufficiently large number such that the probability of more than $\bar{N} - N_S$ walk-ins arriving is close to 0, i.e., $\Pr(\bar{N} < \sum_{t=1}^{T} \beta_t + N_S) < \varepsilon$ where $\varepsilon$ is a small number, say $10^{-9}$. Thus, it only suffices to consider at most $\bar{N}$ patients in the system. Given a schedule $\mathbf{x}$, we can write $\Pi_t(k)$ recursively as

$$\Pi_t(k) = \Pi_{t-1}(0)p_t(k - x_t) + \sum_{j=1}^{k-x_t+1} \Pi_{t-1}(j)p_t(k - x_t - j + 1)$$  \hfill (7)

for $k = 0, \ldots, \bar{N}$ and $t = 1, \ldots, T$ with $\Pi_0(0) = 1$. The first term in equation (7) calculates the joint probability that the system is empty at the beginning of $t - 1$ and that $k - x_t$ patients walk in at the beginning of $t$. The second term evaluates the joint probability that the system is nonempty at the beginning of $t - 1$ and that there are $k$ patients waiting at the beginning of $t$. For $\Gamma_D$, we have

$$\Gamma_D = T + \sum_{k=1}^{\bar{N}} (k - 1) \Pi_T(k),$$  \hfill (8)

in which the second term is the expected number of patients waiting at the end of $T$. 
Before analyzing patient wait time, we need to specify the priority order between scheduled patients and walk-ins. Though some walk-ins arrive due to acute care needs, their health conditions are stable. If indeed walk-ins have emergency issues that require immediate attention, they are often diverted to emergency rooms following the standard clinical protocol. Therefore, common practice usually gives walk-ins lower priority compared to scheduled patients (Berman 2016). The underlying cost structure adopted by practitioners implies that the waiting cost of walk-ins is no larger than that of scheduled patients, i.e., \( C_W \leq C_S = 1 \). The proposition below suggests that, as expected, if \( C_W \leq 1 \) then the service order adopted by practitioners is optimal.

**Proposition 1.** If \( C_W \leq 1 \), it is optimal to serve scheduled patients, if any, before walk-ins.

Next we evaluate patient wait time, starting with the scheduled patients who have priority. Let \( s_t \) be the number of scheduled patients waiting at the end of slot \( t \). We can write \( s_t \) recursively as \( s_t = (s_{t-1} + x_{t-1})^+ \) for \( t = 2, ..., T \) with \( s_1 = (x_1 - 1)^+ \). It follows that the expected wait time of scheduled patients \( \Gamma S(x) \) can be calculated as,

\[
\Gamma S(x) = \sum_{t=1}^{T} s_t + \frac{(s_T - 1)s_T}{2}.
\]

Let \( \Gamma T(x) \) be the expected total wait time of all patients given a schedule \( x \). Using (7), we have

\[
\Gamma T(x) = \sum_{t=1}^{T-1} \sum_{k=1}^{N} (k-1)\Pi_t(k) + \sum_{k=1}^{N} \left[ \sum_{j=1}^{k} (j-1) \right] \Pi_T(k).
\]

Noting that \( \Gamma W \) is the difference between \( \Gamma T \) and \( \Gamma S \), we obtain

\[
\Gamma W(x) = \Gamma T(x) - \Gamma S(x).
\]

Finally, our optimization problem in this section can be represented as,

\[
\min_{x} \Gamma S(x) + C_W \Gamma W(x) + C_D \Gamma D(x) \quad (P1)
\]

\[
\sum_{t=1}^{T} x_t = N_S,
\]

\[
x \in Z^T_+,
\]

\( \Gamma S(x), \Gamma W(x), \Gamma D(x) \) are defined in (8), (9), (10), respectively.

The problem (P1) is a combinatorial optimization problem which is difficult to solve. One common approach in the literature is to first identify the structure of the optimal schedule, and then use this structural result to reduce the feasible region. For instance, Robinson and Chen (2010) show that the optimal appointment schedule, if no walk-ins present, has a “No Holes” property.
That is, if no patients are scheduled for an appointment slot, then none will be scheduled for any subsequent slots. However, we find that this property does not hold when walk-ins are present. Table 2 gives such an example assuming the walk-in pattern follows a zero-inflated Poisson process. We see that the optimal schedule does contain holes at \( t = 4 \) and \( t = 6 \). In the next section, we analyze the structure of the optimal schedule when walk-ins are present.

<table>
<thead>
<tr>
<th>t=1</th>
<th>t=2</th>
<th>t=3</th>
<th>t=4</th>
<th>t=5</th>
<th>t=6</th>
<th>t=7</th>
<th>t=8</th>
<th>t=9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Note: \( T = 9, N_S = 10, C_W = 0.8, C_D = 25 \), the walk-in arrival rate vector is \([0.48, 0.96, 1.16, 1.13, 1.07, 0.90, 0.59, 0.10, 0.01]\), and the zero-event probability is 0.13 for all \( t \).

### 4.1. Properties of the Optimal Schedule

An exhaustive search for the optimal schedule needs to evaluate the total cost for \( \frac{(T+N_S-1)!}{T!} \) times. This becomes computationally infeasible for large-scale problem instances. In this section, we derive several key properties of the optimal schedule when walk-ins are present. These properties allow us to search for the optimal schedule in a much more efficient manner. We start with a few definitions.

**Definition 1 (Non-Empty Period).** If a schedule \( x = (x_1, x_2, ..., x_T) \) satisfies
\[
\sum_{i=\substack{k1 \leq t \leq k2}} x_i - (t - t_{k1}) \geq 1, \quad \forall t \in [t_{k1}, t_{k2}], \quad 1 \leq k1 < k2 \leq T,
\]
we say \([t_{k1}, t_{k2}]\) is a Non-Empty Period (NEP) in this schedule.

The definition above suggests that at any time during an NEP, there is at least one scheduled patient in the system regardless of walk-ins. For example, in schedule
\[
\mathbf{x}_A: \begin{array}{ccc|ccc|ccc}
  t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 & t_9 \\
  0   & 2   & 0   & 0   & 2   & 1   & 0   & 2   & 0
\end{array}
\]
\([t_1, t_2]\) is not an NEP, while \([t_5, t_7]\) is an NEP.

**Definition 2 (Maximum Non-Empty Period).** We say \([t_{k1}, t_{k2}]\) is a Maximum Non-Empty Period (MNEP) in a schedule if (1) \([t_{k1}, t_{k2}]\) is an NEP, and (2) any \([t_a, t_b] \neq [t_{k1}, t_{k2}]\) where \( t_a \leq t_{k1} \) and \( t_b \geq t_{k2} \) is not an NEP.

We note that a schedule may have multiple MNEPs, which, however, cannot be overlapped. A useful property of MNEP is that a slot \( t \) must be in an MNEP if \( x_i > 0 \), which implies that \( x_i = 0 \) if \( t \) is not in any MNEP. Consider the example of \( \mathbf{x}_A \) above: \([t_5, t_7]\) is not an MNEP, while \([t_5, t_9]\) is an MNEP; all MNEPs in \( \mathbf{x}_A \) are \([t_2, t_3]\) and \([t_5, t_9]\).

Based on the MNEPs that a schedule contains, we can organize all feasible schedules into mutually exclusive groups. Within each group of schedules that share the same MNEPs, we are able to fully characterize the optimal one. As discussed below, this knowledge characterizes the structures of the global optimal schedule. We start our discussion with the following result.
Proposition 2. Consider two schedules \( x \) and \( x' \), and let \( \Theta_t(k) \) and \( \Theta'_t(k) \) be their respective probabilities of \( k \) walk-in patients waiting for services at the beginning of \( t \). If \( x \) and \( x' \) have the same MNEPs, then

\[
\Theta_t(k) = \Theta'_t(k), \quad \forall t, k.
\]

We use an example to illustrate Proposition 2. Consider a schedule

\[
x_B: \begin{array}{cccccccc}
t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 & t_9 \\
0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1
\end{array}
\]

Note that \( x_B \) has exactly the same MNEPs (i.e., \([t_2, t_3]\) and \([t_5, t_9]\)) as \( x_A \). As a result, \( x_A \) and \( x_B \) have stochastically the same number of walk-ins waiting for services at any time.

By Definition 2 and Proposition 2, at the end of the last appointment slot \( T \), two schedules with the same MNEPs would have stochastically the same number of scheduled patients and the same number of walk-ins in the system. It suggests that they will have the same expected total duration \( \gamma_D \). Proposition 2 also implies that the expected wait times of walk-ins in these two schedules are equal. Thus, to determine which schedule is better, one only needs to consider the wait time of scheduled patients. This analysis allows us to identify the best schedule among those which share the same MNEPs. We formalize the main result of this section in the following theorem.

Theorem 1. Among all schedules that have the same MNEPs, the optimal schedule starts from \( t = 1 \) and assigns one patient to each appointment slot that belongs to an MNEP. If \( T \) belongs to an MNEP and there are more than one patients left, then the optimal schedule assigns all the remaining patients at \( T \). Otherwise, if \( T \) does not belong to an MNEP, then all the scheduled patients must have been assigned.

To illustrate Theorem 1, consider the schedules \( x_A \) and \( x_B \) mentioned above. We deduce that among all schedules whose MNEPs are \([t_2, t_3]\) and \([t_5, t_9]\), \( x_B \) is the optimal one. And clearly, \( x_B \) results in a smaller overall cost compared to \( x_A \).

Theorem 1 suggests that for all schedules that share the same MNEPs, overbooking, if any, occurs at the last appointment slot \( T \). This overbooking pattern, therefore, will be inherited in the global optimal schedule. More formally, we have the following corollary.

Corollary 1. There exists an optimal schedule for the formulation (P1) such that overbooking, if any, occurs in the last appointment slot \( T \).

These structural properties drastically reduce the search space for the optimal schedule. Specifically, we only need to evaluate and compare those schedules that obey the overbooking pattern described in Corollary 1: (1) if there are \( N_S \geq T \) patients to schedule, slots 1 through \( T-1 \) can either contain one or zero patient, and the rest of the patients will be scheduled in slot \( T \); (2) otherwise, if there are \( N_S \leq T-1 \) patients to schedule, at most \( N_S \) slots among the first \( T-1 \) slots
can be filled with patients (at most one patient in a slot), and the rest, if any, will fill the last slot $T$. The following corollary calculates the exact number of schedules that need to be enumerated in order to find the optimal schedule.

**Corollary 2.** For the formulation (P1), an optimal schedule $x^*$ can be found among $2^{(T-1)}$ possible schedules when $N_S \geq T$, or $\sum_{i=0}^{N_S} \binom{T-1}{i}$ possible ones when $N_S \leq T - 1$.

To illustrate how Corollary 2 helps solve (P1), consider an instance with $T = 12$ and $N_S = 12$. A brute-force enumeration needs to consider $\left( \binom{12}{12} + 12 - 1 \right) = 1352078$ possible schedules, while ours only needs 2048. This reduction of 99.98% in the search space yields a sufficiently small set of possible schedules that are amenable to enumeration. It is important to note that, for even larger scale problems, our approach is still viable.

### 5. Model Incorporating No-show Behavior

In this section, we discuss the model to optimize the appointment schedule when both random walk-ins and customer no-show behaviors are present. To be consistent with our earlier developments, let $x$ be the schedule, our decision vector, and $\beta$ represent the vector of random walk-ins. Let $\alpha(x) = (\alpha_1(x_1), \alpha_2(x_2), ... , \alpha_T(x_T))$ denote the number of show-up patients among those scheduled. Specifically, $\alpha_t(x_t)$ is the number of show-ups at $t$ given that $x_t$ patients are scheduled at $t$. Let $q_t(k, x_t) = Pr(\alpha_t(x_t) = k)$. If patients are homogeneous and independent, and each patient shows up with a specific probability $p_s^t$ at $t$, then $\alpha_t(x_t)$ follows the binomial distribution, i.e.,

$$q_t(k, x_t) = \binom{x_t}{k} p_s^{t^k}(x_t - k)^{1 - p_s^t}, k = 0, 1, ..., x_t.$$

While patient show-up/no-show behaviors are often modeled using a binomial distribution in the appointment scheduling literature (e.g., Hassin and Mendel (2008) and LaGanga and Lawrence (2012)), our optimization model applies to a more general setting. We note that our scheduling model does not need to specify the distribution of $\alpha(x)$, and $q_t(k, x_t)$ can be any type of probability measures which depend on $k$, $x_t$ and $t$.

Same as before, the objective of our optimization model is to minimize the expected total weighted cost, i.e., $\Gamma_S + C_W \Gamma_W + C_D \Gamma_D$. For convenience, we still use $\Pi_t(k)$ to denote the probability of $k$ patients waiting at the beginning of $t$ (before anyone receives service). Given $x$, we have,

$$\Pi_t(k) = \sum_{i=0}^{x_t} \Pi_{t-1}(0) q_t(i, x_t) p_t(k - i) + \sum_{i=0}^{x_t} \sum_{j=1}^{k-i+1} \Pi_{t-1}(j) q_t(i, x_t) p_t(k - i - j + 1)$$

(11)

for $k = 0, ..., N$ and $t = 1, ..., T$ with $\Pi_0(0) = 1$. The first term on the RHS of (11) is the probability of no patient waiting at $t - 1$, $i$ out of $x_t$ scheduled patients showing up at $t$ and $k - i$ patients
walking in at \( t \). The second term is the probability that \( j \) patients were waiting at \( t - 1 \), one patient was served during \( t - 1 \), \( i \) out of \( x_t \) scheduled patients show up at \( t \) and \( k - i - j + 1 \) patients walk in at \( t \).

Similar to our earlier derivation of (8), the expected duration \( \Gamma_D \) here is \( T \) plus the expected number of patients at \( T + 1 \). Thus, we have

\[
\Gamma_D(x) = T + \sum_{k=1}^{N} (k - 1)\Pi_T(k). \tag{12}
\]

Recall that scheduled patients are given priority over walk-ins\(^1\). Let \( \Psi_t(k) \) be the probability of \( k \) scheduled patients waiting at the beginning of \( t \) (before service), then we can write \( \Psi_t(k) \) recursively as,

\[
\Psi_t(k) = \Psi_{t-1}(0)q_t(k, x_t) + \sum_{j=1}^{k-i+1} \Psi_{t-1}(j)q_t(k - j + 1, x_t)
\]

for \( k = 0, \ldots, N_S \) and \( t = 1, \ldots, T \) with \( \Psi_0(0) = 1 \). The first term of the RHS is the probability of no scheduled patient waiting at \( t - 1 \) and \( k \) scheduled patients showing up. The second term is the probability of \( j \) scheduled patients waiting at \( t - 1 \), one of them served, and \( k - j + 1 \) scheduled patients showing up at \( t \). It follows that the expected total wait time for scheduled patients, \( \Gamma_S \), can be calculated by summing up the expected number of scheduled patients waiting at each appointment slot. More precisely, we have

\[
\Gamma_S(x) = \sum_{t=1}^{T-1} \sum_{k=1}^{N_S} (k - 1)\Psi_t(k) + \sum_{k=1}^{N} \sum_{j=1}^{k}(j - 1)\Psi_T(k). \tag{13}
\]

Similarly, the expected total wait time of all patients \( \Gamma_T \) can be calculated by summing up the expected number of all patients waiting at each slot, i.e.,

\[
\Gamma_T(x) = \sum_{t=1}^{T-1} \sum_{k=1}^{N} (k - 1)\Pi_t(k) + \sum_{k=1}^{N} \sum_{j=1}^{k}(j - 1)\Pi_T(k).
\]

And, the expected wait time of walk-ins \( \Gamma_W \) is the difference between \( \Gamma_T \) and \( \Gamma_S \), i.e.,

\[
\Gamma_W(x) = \Gamma_T(x) - \Gamma_S(x). \tag{14}
\]

Finally, the optimization model when both random walk-ins and patient no-show behaviors are present can be formulated as follows,

\[
\min_x \Gamma_S(x) + C_W\Gamma_W(x) + C_D\Gamma_D(x) \tag{P2}
\]

\(^1\)Proposition 1 suggests that if \( C_W \leq 1 \), this priority order is optimal without no-shows and we note that the same result still holds when no-shows are present.
\[
\sum_{t=1}^{T} x_t = N_S, \\
x \in \mathbb{Z}_T^+,
\]

\(\Gamma_S(x), \Gamma_W(x), \Gamma_D(x)\) are defined in (12), (13), (14), respectively.

We note that the formulation \((P2)\) is much harder than \((P1)\). First, the objective of \((P2)\) is more complicated. More importantly, there seems no clear structures for the optimal schedule when both walk-ins and no-shows are present in the model, as discussed in the following remarks.

**Remark 1.** The nice structure of the optimal schedule described in Corollary 1 breaks down, as illustrated by the following example. Table 3 shows the (unique) optimal schedule for the example in Table 2 when the patient no-show probability is 0.2. We see that overbooking does not necessarily occurs at the end.

<table>
<thead>
<tr>
<th>t=1</th>
<th>t=2</th>
<th>t=3</th>
<th>t=4</th>
<th>t=5</th>
<th>t=6</th>
<th>t=7</th>
<th>t=8</th>
<th>t=9</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Note: \(T = 9, N_S = 10, C_W = 0.8, C_D = 25\).

**Remark 2.** As discussed earlier, the “No Holes” property identified by Robinson and Chen (2010) does not hold when only walk-ins are present. We now observe that with no-shows included, this property still needs not to hold, e.g., we see a hole at \(t = 4\) in Table 3.

**Remark 3.** Without walk-ins, let \(x^{w/o}\) be the optimal schedule without no-shows and \(x^w\) be the optimal schedule with no-shows, then we can derive a necessary (and thus weaker) condition from the “No Holes” property in Robinson and Chen (2010):

\[
\sum_{i=1}^{t} x_i^w \geq \sum_{i=1}^{t} x_i^{w/o}, \forall t = 1, 2, \ldots, T. \quad (15)
\]

We defer the derivation of (15) to the Appendix. Condition (15) suggests that when no-show rates increase, it is better to “front-load” the schedule. Intuitively, one may contend that Condition (15) also holds when walk-ins are present. If Condition (15) indeed holds with walk-ins, then one may use the structural results obtained in Corollary 1 (for the system with walk-ins but without no-shows) to reduce the search space for the optimal schedule with both walk-ins and no-shows. However, Condition (15) does not hold when walk-ins are present. Table 4 shows an example in which the optimal schedule is more front-loaded when the no-show rate is smaller.

To give an explanation, consider moving the first scheduled patient from \(t = 1\) to \(t = 2\) in Table 4 while maintaining the positions of other scheduled patients. This movement does not affects the wait time of scheduled patients; it reduces the wait time (cost) of walk-ins, but increases the
total duration (cost) of service. Whether to move the first scheduled patient to a later slot or not depends on the tradeoff between the latter two costs. As the no-show rate increases, there is less reduction in the wait time of walk-ins, but the increment of overall service duration is also smaller. Thus there may still be a net benefit of moving the first scheduled patient to a later slot.

5.1. Two-stage Programming Model
To solve the formulation (P2), we consider a two-stage programming approach. We use \( \Omega^o(\mathbf{x}) \) to denote the set of all possible scenarios given a schedule \( \mathbf{x} \). Let \( \omega^o \in \Omega^o(\mathbf{x}) \) be an arbitrary random scenario, and \( \alpha(\omega^o, \mathbf{x}) \) and \( \beta(\omega^o) \) be the vector of show-up patients and the vector of walk-ins associated with scenario \( \omega^o \), respectively. Let \( T \) be the maximum number of time slots under consideration; one natural choice of \( T \) would be \( T + N - 1 \). Let \( y_t \) be the total number of patients waiting at the end of slot \( t \) and \( y = \{y_1, y_2, ..., y_T\} \). It follows that

\[
y_t = \begin{cases} 
(y_{t-1} + \alpha_t(x_t, \omega^o) + \beta_t(\omega^o) - 1)^+ & \text{for } 1 \leq t \leq T \text{ with } y_0 = 0, \\
(y_{t-1} - 1)^+ & \text{for } T < t \leq T.
\end{cases} 
\]  

(16)

Let \( y_t^s \) be the number of scheduled patients waiting at the end of slot \( t \) and \( y^s = \{y_1^s, y_2^s, ..., y_T^s\} \). We have

\[
y_t^s = \begin{cases} 
(y_{t-1}^s + \alpha_t(x_t, \omega^o) - 1)^+ & \text{for } 1 \leq t \leq T \text{ with } y_0^s = 0, \\
(y_{t-1}^s - 1)^+ & \text{for } T < t \leq T.
\end{cases} 
\]  

(17)

Note that the number of walk-ins waiting at the end of slot \( t \) is \( y_t - y_t^s \). Also, \( \Gamma_D = T + y_T \), and thus \( C_D T \) is a constant which can omitted from the objective function of (P2). We can rewrite (P2) as follows.

\[
\min_{\mathbf{x}} \mathbb{E}_{\omega^o}[\Upsilon(\mathbf{x}, \omega^o)] \\
\sum_{t=1}^T x_t = N_S, \\
x_t \in \mathbb{Z}_+ \text{ for } t = 1, \ldots, T,
\]

where

\[
\Upsilon(\mathbf{x}, \omega^o) := \left\{ \sum_{t=1}^T y_t^s + C_W \sum_{t=1}^T (y_t - y_t^s) + C_D y_T \right\}. 
\]  

(18)

In the following proposition, we demonstrate that evaluating \( \Upsilon(\mathbf{x}, \omega^o) \) in (18) for a given schedule \( \mathbf{x} \) and a realized scenario \( \omega^o \) can be done by solving a minimization problem.
Proposition 3.

\[
\forall (x, \omega) := \begin{cases} 
\min \ y_t + C_W \sum_{t=1}^{T} (y_t - y_t^*) + C_D y_T, \\
y_t \geq y_{t-1} + \alpha_t(x_t, \omega^\omega) + \beta_t(\omega^\omega) - 1 & \text{for } 1 \leq t \leq T \text{ with } y_0 = 0, \\
y_t \geq y_{t-1} - 1 & \text{for } T < t \leq T, \\
y_t^* \geq y_{t-1}^* + \alpha_t(x_t, \omega^\omega) - 1 & \text{for } 1 \leq t \leq T \text{ with } y_0^* = 0, \\
y_t^* \geq y_{t-1}^* - 1 & \text{for } T < t \leq T, \\
y_t, y_t^* \in \mathbb{Z}_+ & \text{for } t = 1, \ldots, T. 
\end{cases}
\]

In (19), the constraints on \( y_t^* \) are very similar to those on \( y_t \). To simplify the presentation, we set \( C_W = 1 \) for now to eliminate \( y_t^* \) and its corresponding constraints. Complete analysis of the model when \( C_W < 1 \) follows a similar procedure, but requires more tedious algebra. We defer it to the Appendix. Setting \( C_W = 1 \), we obtain the following equivalent two-stage programming formulation of (P2):

\[
\min \ E_{\omega^\omega}[\Upsilon(x, \omega^\omega)] \\
\sum_{t=1}^{T} x_t = N_S, \\
x_t \in \mathbb{Z}_+ & \text{for } t = 1, \ldots, T, 
\]

where

\[
\Upsilon(x, \omega^\omega) := \begin{cases} 
\min \ y_t + C_D y_T, \\
y_t \geq y_{t-1} + \alpha_t(x_t, \omega^\omega) + \beta_t(\omega^\omega) - 1 & \text{for } 1 \leq t \leq T \text{ with } y_0 = 0, \\
y_t \geq y_{t-1} - 1 & \text{for } T < t \leq T, \\
y_t \in \mathbb{Z}_+ & \text{for } t = 1, \ldots, T. 
\end{cases}
\]

The main difficulty in solving (T1) is that it is neither a two-stage linear nor integer programming model. The complicating term is \( \alpha_t(x_t, \omega^\omega) \), which for a given scenario \( \omega^\omega \) may not be represented as a linear function of \( x_t \). That is, \( \alpha_t(x_t, \omega^\omega) \) cannot be represented as \( f_1(x_t) \times f_2(\omega^\omega) \) for some \( f_1(\cdot) \) and \( f_2(\cdot) \). In the next section, we introduce a simple and yet innovative reformation which transforms (T1) into a stochastic integer programming model.

5.2. Problem Reformulation

We define a new set of decision variables \( z_{t,i}, t = 1,2,\ldots,T \) and \( i = 1,2,\ldots,N_S \), such that if patient \( i \) is scheduled at \( t \) then \( z_{t,i} = 1 \), otherwise \( z_{t,i} = 0 \). Let \( z = \)}
(z_{1,1}, \ldots, z_{1,N_S}, \ldots, z_{2,1}, \ldots, z_{2,N_S}, \ldots, z_{T,1}, \ldots, z_{T,N_S})^T \in \{0,1\}^{T \cdot N_S}. Noting that x_t = \sum_{i=1}^{N_S} z_{t,i}, \forall t = 1, 2, \ldots, T, we obtain an equivalent two-stage stochastic integer programming model of (T1), described in the following proposition. Let \( \Omega(z) \) be the set of all possible scenarios given \( z \). For an uncertain scenario \( \omega \in \Omega(z) \), \( \gamma_{t,i}(\omega) \) is the indicator for patient \( i \)'s show-up status at \( t \) (1 means show-up and 0 otherwise), and \( \beta_t(\omega) \) is the number of walk-ins in \( t \).

**Proposition 4.** If for all \( t = 1, 2, \ldots, T \), each scheduled patient in slot \( t \) shows up independently with the same probability \( p_s^t \geq 0 \), then Problem (T1) is equivalent to the following formulation:

\[
\min_z \mathbb{E}_\omega[\Upsilon(z, \omega)]
\]

\[
\sum_{t=1}^{T} z_{t,i} = 1 \text{ for } 1 \leq i \leq N_S,
\]

\[
z_{t,i} \in \{0,1\} \text{ for } 1 \leq t \leq T, 1 \leq i \leq N_S,
\]

where

\[
\Upsilon(z, \omega) := \begin{cases}
\min_y \sum_{t=1}^{T} y_t + C_D y_T \\
y_t \geq y_{t-1} + \sum_{i=1}^{N_S} \gamma_{t,i}(\omega) z_{t,i} + \beta_t(\omega) - 1 & \text{for } 1 \leq t \leq T \text{ with } y_0 = 0, \\
y_t \geq y_{t-1} - 1 & \text{for } T < t \leq \overline{T}, \\
y_t \in \mathbb{Z}_+ & \text{for } 1 \leq t \leq \overline{T}.
\end{cases}
\]

**Remark 4.** Formulation (T1) is equivalent to Formulation (20)-(22) when each scheduled patient in slot \( t \) shows up independently with the same probability \( p_s^t \geq 0 \). We note that Formulation (20) is more general, because it can deal with the situation where patients have independent but different show-up probabilities. To do that, one only needs to specify \( \gamma_{t,i}(\omega) \) as patient \( i \)'s show-up status at time \( t \) (see more discussion in Section 7).

To make it compact and easy to present, we will use the matrix form for (20) in the analysis below. The matrix form of Problem (20) is

\[
\min_z \mathbb{E}_\omega[\Upsilon(z, \omega)]
\]

\[
Wz = e,
\]

\[
z \in \{0,1\}^{T \cdot N_S},
\]

where

\[
\Upsilon(z, \omega) := \begin{cases}
\min_y c' y \\
U y \geq M(\omega) z + d(\omega),
\end{cases}
\]

\[
y \in \mathbb{Z}_+.
\]
In the formulation above, $e$ is the $N_S$ dimensional unit vector; $W$ is made up of $T$ identity matrices, i.e., $W = [I \ I \ \cdots \ I]$ where $I$ is the $N_S$ dimensional identity matrix; $c = (1, ..., 1, 1 + C_D, 1, ..., 1)^T$ is a $T$ dimensional vector where all elements are 1 except for the $T$th element; $U$ is a $T$ dimensional square matrix,

$$U = \begin{bmatrix}
1 & 0 & 0 & 0 & \cdots & 0 & 0 \\
-1 & 1 & 0 & 0 & \cdots & 0 & 0 \\
0 & -1 & 1 & 0 & \cdots & 0 & 0 \\
0 & 0 & -1 & 1 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \cdots & -1 & 1
\end{bmatrix};$$

$M(\omega)$ is a $T$ by $N_S \times T$ matrix, where element $M_{t,(t-1) \times N_S + i}(\omega) = \gamma_{i,\omega}$ for $t \leq T, i \leq N_S$ and all other elements are 0; $d(\omega)$ is a $T$ dimensional vector, where $d_t(\omega) = \beta_t(\omega) - 1$ when $t \leq T$ and $d_t(\omega) = -1$ when $t > T$.

With the matrix form above, we are now ready to present the main result of this section. Noting that the matrix $U$ in (23) is totally unimodular\(^2\), we deduce that (T1) can be reformulated into a stochastic integer programming problem where the first stage is a 0-1 integer program and the second stage is a pure linear program. This reformulation is stated in the theorem below.

**Theorem 2.** Problem (T1) can be reformulated as follows,

$$\min_z \mathbb{E}_\omega[\Upsilon(z,\omega)]$$

$$Wz = e,$$

$$z \in \{0, 1\}^{T \cdot N_S},$$

where

$$\Upsilon(z,\omega) := \{\min_{y \geq 0} c' y | U y \geq M(\omega)z + d(\omega)\}. \quad \text{(Prim)}$$

5.3. Solution Approaches

5.3.1. Sample Average Approximation One common approach to solve a two-stage stochastic programming problem is via Sample Average Approximation (SAA), i.e., randomly generating a sufficient number of sample scenarios and then minimizing the average cost of these samples. With a slight abuse of the notations, we let $\Omega$ be the set of all samples randomly generated, and $\omega \in \Omega$ represent one sample in the set. Let $|\Omega|$ denote the number of samples. Then, we can (approximately) solve (T2) by solving the following integer programming problem.

$$\min_{y(\omega), z} \frac{1}{|\Omega|} \sum_{\omega \in \Omega} [c' y(\omega)]$$

\(^2\) $U$ is totally unimodular because any element in $U$ is 0, 1 or -1, and every row in $U$ has at most 2 non-zero elements.
\[ Wz = e, \]
\[ Uy(\omega) - M(\omega)z \geq d(\omega), \quad \forall \omega \in \Omega, \]
\[ y(\omega) \geq 0, \quad \forall \omega \in \Omega, \]
\[ z \in \{0, 1\}^{T \cdot N_S}. \]

**Remark 5.** If we relax the 0-1 integer constraints for \( z \) in (T2-SAA), we obtain a simple linear program (LP). However, the coefficient matrix in (T2-SAA) is not necessarily unimodular (see more details in the Appendix), and thus the LP relaxation of (T2-SAA) may not be exact.

By reformulating the original problem (T1) into a mixed integer program (T2-SAA), we make an untractable problem amenable by many off-the-shelf optimization software packages such as CPLEX. However, we note that the second stage of (T2) is a pure LP, which enables us to develop more efficient solution methods.

### 5.3.2. Constraint Generation Algorithm

In this section we propose a constraint generation algorithm for (T2). Our algorithm can be applied to any two-stage stochastic program of which the second stage is a pure LP. Our algorithmic approach is motivated by Wollmer (1980). Given \( z \) and scenario \( \omega \), we can write the dual of the second stage problem (Prim) as follows.

\[
\max_v \quad v'(M(\omega)z + d(\omega)) \quad \text{(Dual)}
\]
\[ U'v \leq c, \]
\[ v \geq 0. \]

Recall that the primal problem (Prim) is to calculate the cost under scenario \( \omega \) and decision \( z \), so it is always feasible and bounded. Thus, the dual problem (Dual) is also feasible and bounded. Let \( v(z, \omega) \) be the optimal solution of (Dual) given \( z \) and \( \omega \). Denote the set \( \{ z | Wz = e, z \in \{0, 1\}^{T \cdot N_S} \} \) as \( Z \). By strong duality, we have

\[
v(z, \omega)^{tr}(M(\omega)z + d(\omega)) = \Upsilon(z, \omega), \quad \forall z \in Z. \quad (24)
\]

As (Dual) is a maximization problem, we know that

\[
v(z', \omega)^{tr}(M(\omega)z + d(\omega)) \leq \Upsilon(z, \omega), \quad \forall z' \in Z. \quad (25)
\]

Now, let

\[
a(z') = \mathbb{E}_\omega[v(z', \omega)^{tr}M(\omega)],
\]
\[
b(z') = \mathbb{E}_\omega[v(z', \omega)^{tr}d(\omega)],
\]
\[
\Upsilon(z') = \mathbb{E}_\omega[\Upsilon(z', \omega)].
\]
Taking expectation on both sides of (25), we can see that for any given \( z \in \mathcal{Z} \),

\[
a(z')z + b(z') \leq \Upsilon(z), \quad \forall z' \in \mathcal{Z}.
\]  

(26)

In particular, when \( z' = z \), we have

\[
a(z)z + b(z) = \Upsilon(z).
\]  

(27)

Observing (26) and (27), we can then reformulate (T2) as follows.

**Theorem 3.** Problem (T2) is equivalent to the following Problem (T2-D):

\[
\min_{z \in \mathcal{Z}, u} u \quad \text{subject to} \quad a(z')z + b(z') \leq u \quad \forall z' \in \mathcal{Z}.
\]  

(28)

Following Theorem 3, we have two corollaries which provide theoretical support to our algorithmic approach. We use \( z^* \) to denote the optimal solution to (T2).

**Corollary 3.** If \( \pi \geq \Upsilon(z^*) \) and a solution \( z^0 \) does not satisfy \( a(z')z^0 + b(z') \leq \pi \) for some \( z' \in \mathcal{Z} \), then \( z^0 \) is not an optimal solution.

**Corollary 4.** If \( z^0 \) is not an optimal solution, then \( a(z^0)z + b(z^0) \leq a(z)z + b(z) \), \( \forall z \in \mathcal{Z} \).

Corollary 3 suggests that we may eliminate a non-optimal solution \( z \) without fully solving for \( \Upsilon(z) \). Corollary 4 implies that if \( z^0 \) is not optimal, then its corresponding constraint set \( a(z^0)z + b(z^0) \leq u \), \( \forall z \in \mathcal{Z} \) is redundant. Following these ideas, we propose the following algorithm.

**Constraint Generation Algorithm (CGA)**

1. **Step 0:** Solve the LP relaxed version of (T2-SAA) (see (T2-LP) in the Appendix) and get the solution \( z_{LP}^i \) and its optimal objective value \( Val(LP) \). For each \( i \in [1, N_s] \), find \( t_i \) such that \( z_{LP}^i \) is the largest one among \( \{z_{LP}^1, ..., z_{LP}^N\} \) (ties broken by choosing the smallest \( t_i \)). Set \( z^* \) such that \( z^*_t,i := 1 \) for \( t = t_i \), \( z^*_t,i := 0 \) for \( t \neq t_i \). Set \( k := 1, A := [a(z^*)], b := [b(z^*)], e = [1], u := Val(LP) \) and \( \pi := \Upsilon(z^*) \).
2. **Step 1:** If \( \pi - u < \epsilon \) for some preset small number \( \epsilon \), then stop and return \( z^* \); otherwise go to Step 2.
3. **Step 2:** Solve the problem \( \min_{z,u} u | Az + b \leq u e, u \leq \pi, z \in \mathcal{Z} | \). Get the solution \( (z^0, u^0) \), and go to Step 3.
4. **Step 3:** If \( u^0 > \pi \) then set \( u := u^0 \). If \( u^0 = \pi \), stop and return \( z^0 \); otherwise go to Step 4.
5. **Step 4:** Calculate \( \Upsilon(z^0) \). If \( \Upsilon(z^0) < \pi \) then set \( \pi := \Upsilon(z^0) \) and \( z^* := z^0 \).
6. **Step 5:** Set \( A := [A; a(z^0)], b := [b; b(z^0)], e := [e; 1] \), and go to Step 1.
In the algorithm above, Step 0 initializes a solution from the linear relaxation problem. (This initial solution often turns out to be optimal in our numerical experiments.) \( \pi \) and \( \underline{u} \) represent the upper bound and lower bound of the optimal objective value, respectively. If they are equal (or very close) in Step 1, then we achieve optimality. In Step 2, we choose a solution \( z^0 \) which satisfies \( Az^0 + b \leq \pi \) (for the sake of efficiency, we choose the solution which gives the minimum \( u \)). In Step 3, we update the lower bound and check the optimality of the solution obtained in Step 2. If it is not optimal, then in Step 4, we check if it can update the existing solution and the upper bound. In Step 5, we generate the corresponding constraint of \( z^0 \) to narrow the search space when we return to Step 2. We know that the optimal solution satisfies this new constraint, because any solution that does not is not optimal by Corollary 3. This new constraint, by Corollary 4, is not (obviously) redundant and hence added to the constraint set. Our next theorem formalizes that the algorithm above indeed solves (T2) to optimality.

**Theorem 4.** The Constraint Generation Algorithm stops in a finite number of iterations, and its output is an optimal solution of problem (T2).

### 6. Numerical Study

Our numerical study has three purposes. First, we compare the brute-force enumeration, off-the-shelf optimization solver and our proposed constraint generation algorithm in terms of accuracy and speed. Second, we study the pattern of the optimal schedule. Specifically, we investigate how changes in the practice environment (e.g., waiting cost and no-show rate changes) impact the optimal appointment schedule. Finally, we carry out a case study using real data collected from our collaborating organization to demonstrate the efficiency gains that may result from adopting our proposed scheduling approaches compared to the current practice.

We use a variety of model parameters in our numerical study to capture various settings, and we introduce these parameters below. Motivated by our empirical findings in Section 3, we create three synthetic providers who face different walk-in patterns. Figure 2 shows the expected number of walk-ins faced by these providers in one clinic session. Provider 1 has \( T = 9 \) appointment slots in one session and observes a unimodal walk-in arrival pattern. Providers 2 and 3 have 10 and 11 slots, respectively. Walk-in arrivals to these two providers are bimodal. For each provider, we consider three levels of workload by choosing \( N_S \), the number of scheduled patients. For a light workload, we set \( N_S = 0.9T - N_W \), where \( N_W \) is the expected total number of walk-ins. For medium and heavy workloads, we set \( N_S \) to be \( T - N_W \) and \( 1.2T - N_W \), respectively. Previous literature suggests that the provider unit overtime cost is around 15 times of the patient unit waiting cost, and the provider unit idle cost is around 10 times (Robinson and Chen 2010, LaGanga and Lawrence 2012, Zacharias and Pinedo 2014). Recall that we normalize the waiting cost for scheduled patients to
be 1. We thus vary the unit duration cost $C_D$ (sum of overtime cost and idle cost) between 15 and 35. For walk-ins, we set their unit waiting cost $C_W$ to be in the range of 0.5 to 1.

![Figure 2: Expected Number of Walk-ins Faced by Three Synthetic Providers Over Time.](image)

**Note.** (1) Walk-ins to Provider 1 follow a zero-inflated Poisson process with time-varying arrival rates [0.48 0.96 1.16 1.13 0.77 0.83 0.02 0.83 0.13 0.04] and a constant zero-event probability 0.13. (2) Walk-ins to Provider 2 follow a zero-inflated Poisson process with time-varying arrival rates [0.61 1.00 0.77 0.83 0.02 0.83 0.53 0.19 0.04] and time-varying zero-event probabilities [0.39 0.58 0.40 0.33 0.00 0.14 0.22 0.16 0.00]. (3) Walk-ins to Provider 3 follow a Poisson process with time-varying arrival rates [0.04 0.39 0.54 0.52 0.28 0.02 0.80 0.29 0.10 0.03 0.01].

### 6.1. Performance Comparison of SAA and Enumeration

In this section, we compare the computational performances of using enumeration and using the Sample Average Approximation (SAA) formulation (T2-SAA) to solve for the optimal schedule. We only consider Provider 2 in our comparison. We use two methods to solve the SAA formulation: an off-the-shelf solver “lpsolve 5.5.2.0” and our Constraint Generation Algorithm (CGA) described in Section 5.3.2. We vary the cost parameters, no-show rate and the workload level in our tests. All computations were conducted in a computer equipped with Intel Core i7-5500U 2.40 GHz CPU, 8.00 GB RAM, 64-bit Windows 10 OS and Matlab R2013a. We randomly sample 150 scenarios in our SAA formulation, and report that increasing the sample size seems to linearly increase the computation time but does not increase the accuracy of the results. Table 5 shows the comparison details.

The solutions achieved by SAA are very close to, and in most cases, exactly the optimal solutions obtained by enumeration. The gaps, if any, in the objective values are no more than 0.1% in all scenarios.

---

3 The problem size of Provider 3 is too big making it impossible to use enumeration. The problem size of Provider 1, however, is fairly small and thus is less interesting.

4 See: http://lpsolve.sourceforge.net/.
cases tested; and thus we choose not to list these gaps. Our SAA, however, solves the problem much faster than enumeration. When $N_S = 9$, SAA is about 100 times faster than enumeration. When $N_S = 18$, enumeration fails to find the optimal solution within the time allowance we give (and thus we do not show the corresponding computation times in Table 5).

Compared to the off-the-shelf solver, our CGA performs equally well in solving small problems ($N_S = 6, 7, 8$). As the problem size increases, the existing solver slows down quite a bit, while our CGA remains fast. When $N_S = 18$, our CGA can be 5-10 times faster than the existing solver in most cases. We note that in only a few cases our CGA is slow. This may be due to the fact that the constraints generated in those cases happen to be ineffective. Given the strong potentials of the CGA, it would be interesting to further improve its efficiency in the future.

### 6.2. Analysis of the Optimal Schedule Patterns

Using the synthetic data above, we conduct an extensive sensitivity analysis to investigate the impact of walk-in patterns, no-show rate and cost structure on the optimal schedule. Figures 3
and 4 depict, respectively, the optimal schedules for Providers 1 and 3 under various combinations of model parameters. We note that Provider 1 faces a unimodal walk-in pattern while Provider 3 sees bimodal. To save space, we present the numerical results of Provider 2 in the Appendix since Provider 2, like Provider 3, also sees a bimodal walk-in pattern.

In Figure 3, the total number of slots $T = 9$, and the number of scheduled patients $N_S = 5$ so that $N_W + N_S = 1.2T$ (i.e., Provider 1 is under a high workload). The figure contains 4 panels, and each consists of 4 sub-figures. Within each panel, $C_D$ and $C_W$ are the same; the no-show rate increases from 0 to 0.2 at an increment size of 0.05 from the leftmost sub-figure to the rightmost one. In each sub-figure, the height of each blue bar represents the optimal number of patients scheduled in each time slot (see the left vertical axis); the grey curve on the top shows the expected number of walk-ins arriving in each time slot (see the right axis). Figure 4 shows the similar information for Provider 3, who has $T = 12$ and $N_S = 10$ (so that she also faces a high workload).

Intuition suggests us to reserve “holes” in the appointment schedule in anticipation for walk-ins and that these holes should follow, but not coincide with, the peak of walk-in arrivals (due to the queueing effects). While we observe such patterns in some scenarios (e.g., for Provider 1 when $C_D = 15$ and $C_W = 0.5$), the exact optimal schedule exhibits much more complicated forms. If we think of a consecutive period without scheduled patients as a single “hole”, we see that the number of holes is not necessarily the same as the number of modes in the distribution of walk-ins. Provider 1 who sees a unimodal walk-in pattern may have two holes in her schedule; while Provider 2 who faces a bimodal walk-in process can only have a single hole in her schedule.
The exact optimal schedule depends on a variety of factors such as walk-in patterns and cost structures, and cannot be easily obtained without an optimization approach. We can, however, make a few important observations here. First, when the duration cost rate $C_D$ (i.e., the sum of the idle time cost rate $C_I$ and the overtime cost rate $C_O$) is larger, the optimal schedule appears to be more front-loaded. The reason is that front-loading the schedule reduces the overall duration of the service. Second, when the waiting cost of walk-ins becomes higher, we want to reduce waits for walk-ins. Two approaches can be used here and which one to use depends on the magnitude of the no-show rate. When the no-show rate is relatively low, one may place more scheduled patients towards the end of the session; if the no-show rate becomes high, then one can overbook earlier slots in the session. Third, we observe that for these providers, when the no-show rate increases, the optimal schedule tends to move scheduled patients to earlier slots and to overbook the first slot. This scheduling pattern, however, is not universally optimal (recall Remark 3 following the formulation $\textbf{(P2)}$ that when the no-show rate increases, it may be more beneficial to move scheduled patients to later slots). The schedule pattern we observe here is likely due to the fact that there are relatively few walk-ins in the early session.

6.3. Case Study

In this section, we examine the potential performance improvement by adopting the optimal appointment schedule suggested by our model to the current practice. To populate our case study, we use the same dataset as used in Section 3. We select Providers KNI and GAR as cases due to their representativeness: these two providers have quite different walk-in patterns, no-show rates
and workloads as discussed below. For each provider, we sample a number of days during which he/she works through the whole clinic session (sometimes the providers may leave early).

In May 2011, KNI worked from 9am to 4pm every Friday, and from 9am to 1pm every Saturday. We choose to analyze the morning session on Fridays, because KNI takes a lunch break at 1pm. For each of these days, we reconstruct from the historical data the total number of scheduled patients $N_S$ and the actual schedule used from 9am to 1pm (i.e., the number of scheduled patients in each appointment slot). As this health center uses half-hour slots, we have 8 slots for each morning session. The average patient no-show rate for KNI is estimated to be 0.36 based on the data. As for walk-ins, we use the empirical result in Section 3.3. Recall that the walk-in pattern of KNI in the morning is a time-varying zero-inflated Poisson process with a peak at 10am.

For GAR, we use data of all Fridays from July 1, 2011 to August 5, 2011. During that period, GAR worked from 9:00am to 3pm, and did not take a lunch break. Thus, we reconstruct 6 original schedules, each with 12 slots. The average no-show rate faced by GAR is estimated to be 0.16, and the walk-ins follow a Poisson process with increasing arrival rates over time.

For each clinic session reconstructed above, we evaluate the expected cost under the original schedule and the expected cost under the schedule suggested by our optimization model, based on the provider-specific data. We then calculate the percentage reduction in expected total cost if our schedule were adopted. Tables 6 shows these percentage improvements in a range of parameter settings. We note that the potential daily cost savings for Provider KNI range from 1% to 21%, and from 4% to 59% for Provider GAR. On average, KNI sees a 10% cost reduction and GAR 30%. These represent substantial improvements that may result from adopting our scheduling approach to the current practice. These improvements can translate into increased operational efficiency, reduced patient wait time, improved experience of care and staff morale.

7. Extension

Our modeling framework and solution approach can be easily extended to incorporate several other factors that may affect the design of appointment schedule templates if necessary.

7.1. Random Service Times

As discussed before, it is reasonable to assume deterministic service times in many practical contexts as providers can often adjust their time with patients depending on the progress of the day. It is, however, sometimes important to consider the variability in service times when scheduling patients. There are empirical evidences suggesting that provider service times may follow the exponential distribution (Kopach et al. 2007). Some previous literature, such as Kaandorp and Koole (2007), Hassin and Mendel (2008), has also considered exponentially distributed service times in their scheduling models. Our models can be extended to incorporate such random service times.
Table 6  Case Study: Performance Improvement for Providers KNI and GAR.

<table>
<thead>
<tr>
<th>Provider</th>
<th>Date</th>
<th>NS</th>
<th>$C_D = 15$</th>
<th>$C_D = 35$</th>
<th>$C_D = 15$</th>
<th>$C_D = 35$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>$C_W = 0.5$</td>
<td>$C_W = 0.5$</td>
<td>$C_W = 1$</td>
<td>$C_W = 1$</td>
</tr>
<tr>
<td>KNI</td>
<td>5/6/11</td>
<td>10</td>
<td>3.19%</td>
<td>6.46%</td>
<td>5.23%</td>
<td>9.72%</td>
</tr>
<tr>
<td></td>
<td>5/7/11</td>
<td>22</td>
<td>17.03%</td>
<td>9.15%</td>
<td>13.71%</td>
<td>5.89%</td>
</tr>
<tr>
<td></td>
<td>5/13/11</td>
<td>14</td>
<td>6.71%</td>
<td>3.36%</td>
<td>5.31%</td>
<td>4.43%</td>
</tr>
<tr>
<td></td>
<td>5/14/11</td>
<td>22</td>
<td>18.34%</td>
<td>10.32%</td>
<td>15.26%</td>
<td>7.19%</td>
</tr>
<tr>
<td></td>
<td>5/20/11</td>
<td>13</td>
<td>8.97%</td>
<td>2.73%</td>
<td>5.12%</td>
<td>1.44%</td>
</tr>
<tr>
<td></td>
<td>5/21/11</td>
<td>16</td>
<td>20.77%</td>
<td>9.61%</td>
<td>15.95%</td>
<td>6.09%</td>
</tr>
<tr>
<td></td>
<td>5/27/11</td>
<td>14</td>
<td>18.92%</td>
<td>9.00%</td>
<td>14.60%</td>
<td>7.26%</td>
</tr>
<tr>
<td></td>
<td>5/28/11</td>
<td>14</td>
<td>17.77%</td>
<td>7.32%</td>
<td>12.75%</td>
<td>5.05%</td>
</tr>
<tr>
<td>GAR</td>
<td>7/1/11</td>
<td>7</td>
<td>26.44%</td>
<td>31.96%</td>
<td>19.86%</td>
<td>27.65%</td>
</tr>
<tr>
<td></td>
<td>7/8/11</td>
<td>6</td>
<td>47.68%</td>
<td>49.92%</td>
<td>38.93%</td>
<td>45.28%</td>
</tr>
<tr>
<td></td>
<td>7/15/11</td>
<td>3</td>
<td>14.71%</td>
<td>13.51%</td>
<td>12.65%</td>
<td>12.57%</td>
</tr>
<tr>
<td></td>
<td>7/22/11</td>
<td>8</td>
<td>16.26%</td>
<td>17.88%</td>
<td>4.04%</td>
<td>10.08%</td>
</tr>
<tr>
<td></td>
<td>7/29/11</td>
<td>9</td>
<td>33.27%</td>
<td>36.51%</td>
<td>25.53%</td>
<td>31.78%</td>
</tr>
<tr>
<td></td>
<td>8/5/11</td>
<td>6</td>
<td>55.61%</td>
<td>58.57%</td>
<td>47.95%</td>
<td>54.60%</td>
</tr>
</tbody>
</table>

Note: (1) For Provider KNI, $T = 8$ and the no-show rate is 0.36. (2) For Provider GAR, $T = 12$ and the no-show rate is 0.16. (3) Percentage improvement is evaluated as the percentage reduction in expected total cost due to adopting the optimal schedule.

Specifically, suppose that the service times of patients are i.i.d. exponential random variables with mean $\phi$. Then, the number of potential departures within a single slot of time (given that there are enough patients waiting) has a Poisson distribution with mean $1/\phi$. Let $\delta_t$ be this potential number of departures in slot $t$, then $Pr(\delta_t = i) = \frac{\phi^i}{i!} e^{-1/\phi}$, $i = 0, 1, 2, \ldots$. Note that this distributional result is independent of time given the memoryless property of the exponential distribution.

Recall that (16) defines the relationship between $y_t$ and $y_{t-1}$, where $y_t$ is the total number of patients waiting at the end of $t$. By expanding the definition of the random scenario $\omega^o$ to include the uncertainty of random service times, we can redefine the relationship of $y_t$ and $y_{t-1}$ as follows.

$$y_t = \begin{cases} (y_{t-1} + \alpha_t(x_t, \omega^o) + \beta_t(\omega^o) - \delta_t(\omega^o)^+) & \text{for } 1 \leq t \leq T \text{ with } y_0 = 0, \\ (y_{t-1} - \delta_t(\omega^o))^+ & \text{for } T < t \leq T, \end{cases}$$  \hspace{1cm} (29)

where $\delta_t(\omega^o)$ is the aforementioned number of potential maximum departures in slot $t$ associated with scenario $\omega^o$. The correctness of the recursive equation (29) follows from the memoryless property of the exponential distribution.

Similarly, we can redefine the relationships between $y^s_t$ and $y^s_{t-1}$, where $y^s_t$ is the total number of scheduled patients waiting at the end of $t$. Then, we can follow the same reformulation process above and use the same solution approach to solve this new problem. We note that, however, if the service time distribution is not exponential, the problem becomes much more complicated because we would need to record the service starting time of each patient in the system state. We leave this more general extension to future research.
7.2. Heterogeneous No-show Behaviors

Patients may exhibit different no-show behaviors. For instance, Goldman et al. (1982) show that patient no-show behavior is correlated with age, race, presence of psychosocial problems, and the past no-show behavior. Other factors such as marital status and distance to the provider may also be important predictors for patient no-shows (Daggy et al. 2010).

If individual patient information is not known \textit{a priori} when designing the schedule template, it is reasonable to use a constant “average” no-show rate in the formulation. This is a key assumption of most previous appointment scheduling literature, e.g., Kaandorp and Koole (2007), Robinson and Chen (2010) and LaGanga and Lawrence (2012). These models require that the number of show-ups (for a given number of scheduled patients) follows a binomial distribution, and thus may not work in the presence of heterogenous patient no-shows.

If individual no-show behavior is known in advance, then incorporating such knowledge in the optimization may deliver better schedules (Zacharias and Pinedo 2014). Our formulation (20) can easily deal with \textit{individual-dependent} and \textit{time-dependent} patient no-show behavior by specifying $\gamma_{t,i}(\omega)$ to be patient $i$’s show-up probability at time $t$.

7.3. Patient Preferences

As patient-centeredness is an important goal in the current health care environment (Feldman et al. 2014), patient preferences may play a role in scheduling decision making. Our model can take patient preferences into account in the following way. Let $f$ be a non-negative vector which represents the cost of scheduling patients to their undesired time slots. Specifically, let $f_{t,i} > 0$ represent the “preference” cost of scheduling patient $i$ to slot $t$. A larger $f_{t,i}$ indicates that patient $i$ is more aversive to be scheduled in slot $t$. The extended model considering patient preferences can be formulated as,

$$\min_{z} \mathbb{E}_{\omega}[\Upsilon(z,\omega)] + f'z \quad \text{(T2-P)}$$

$$Wz = e,$$

$$z \in \{0,1\}^{T \times N_{S}},$$

where

$$\Upsilon(z,\omega) := \{\min_{y \geq 0} c'y | Uy \geq M(\omega)z + d(\omega)\}.$$ 

Note that (T2-P) is still a two-stage stochastic programming problem as (T2), in which the second stage is a pure LP and the first stage only involves binary decision variables. Our reformulation and solution approaches developed above still apply.
8. Conclusion

In this paper we study how to schedule a fixed number of patients within a clinic session, during which a random number of walk-ins may arrive for services. The objective is to minimize the expected total cost of patient waiting, provider idling and overtime. We formulate the problem as a two-stage stochastic optimization model, and develop effective solution approaches for various settings. Specifically, when patient no-shows are of less concern, we characterize the structure of the optimal schedule. Using this structural result, we develop effective solution approaches to achieve optimality. When patient no-shows become non-negligible, we show that the original model, which is difficult to deal with directly, can be reformulated as a two-stage stochastic integer program which is amenable to standard optimization software. Leveraging the structure of the reformulation, we also propose effective solution algorithms. Our numerical experiments demonstrate that, with walk-ins, the optimal schedule has a completely different structure from those identified in the previous literature which often does not explicitly consider walk-ins. Using our data from practice, we reconstruct the actual schedules currently in use, and predict a significant cost reduction if providers were to switch to the schedules suggested by our models.

Our work is motivated by the operations management challenge faced by many outpatient care facilities that accept both scheduled and walk-in patients. This challenge becomes more evident in our own analysis of a large dataset obtained from a major community health center located in New York City. Our data analysis shows that walk-ins constitute a large stream of customer arrivals, and also reveals new and complex temporal-patterns for customer arrivals. Though our work is motivated by healthcare applications, our optimization models and results can be applied to general appointment-based services in the presence of random walk-ins.

There are several ways to extend our research. First, in this work we consider how to assign patients into a predetermined set of discrete slots, and this is a commonly adopted approach in the literature; see, e.g., Robinson and Chen (2010), LaGanga and Lawrence (2012) and Zacharias and Pinedo (2014). Another modeling approach is to decide the scheduled arrival time for each patient. Second, while our model can deal with exponentially distributed service times, it would be useful to incorporate random service times with general probability distributions in the model. This, however, is a very difficult task as we lose the memoryless property. Third, some practice may set a specific time window (e.g., 10-2) to accept walk-ins (see, e.g., McKinley Health Center (2016)), and it would be interesting to study how to choose such a time window. Modeling this decision requires a deep understanding on how changes of this time window affect the arrival pattern of walk-ins. As all these directions are likely lead to completely new models and solution approaches, we leave them for future research.
References


Djuretic, Tamara, Mike Catchpole, James S Bingham, Angela Robinson, Gwenda Hughes, George Kinghorn. 2001. Genitourinary medicine services in the united kingdom are failing to meet current demand. Int. J. STD & AIDS 12(9) 571–572.


Appendix

A. Summary of Notations

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>The total number of regular time slots (i.e., the regular length of a clinic session)</td>
</tr>
<tr>
<td>$N_S$</td>
<td>The total number of patients to be scheduled</td>
</tr>
<tr>
<td>$N_W$</td>
<td>The expected total number of walk-in patients</td>
</tr>
<tr>
<td>$C_S$</td>
<td>Unit time waiting cost per scheduled patient, normalized to be 1</td>
</tr>
<tr>
<td>$C_W$</td>
<td>Unit time waiting cost per walk-in</td>
</tr>
<tr>
<td>$C_I$</td>
<td>Unit time idle cost for the provider</td>
</tr>
<tr>
<td>$C_O$</td>
<td>Unit time overtime cost for the provider</td>
</tr>
<tr>
<td>$C_D$</td>
<td>Unit time duration cost, i.e., $C_I + C_O$</td>
</tr>
<tr>
<td>$N$</td>
<td>A sufficiently large number</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decision variables</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_t$</td>
<td>The number of patients scheduled in slot $t$</td>
</tr>
<tr>
<td>$x$</td>
<td>The vector of $x_t$</td>
</tr>
<tr>
<td>$y_t$</td>
<td>The total number of patients waiting at the end of slot $t$</td>
</tr>
<tr>
<td>$y$</td>
<td>The vector of $y_t$</td>
</tr>
<tr>
<td>$z_{t,i}$</td>
<td>If patient $i$ is scheduled at $t$ then $z_{t,i} = 1$, otherwise $z_{t,i} = 0$</td>
</tr>
<tr>
<td>$z$</td>
<td>The vector of $z_{t,i}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Random variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta_t$</td>
</tr>
<tr>
<td>$\beta$</td>
</tr>
<tr>
<td>$\alpha_t(x_t)$</td>
</tr>
<tr>
<td>$\alpha(x)$</td>
</tr>
<tr>
<td>$\Omega_0(x)$</td>
</tr>
<tr>
<td>$\omega_0$</td>
</tr>
<tr>
<td>$\Omega(z)$</td>
</tr>
<tr>
<td>$\omega$</td>
</tr>
<tr>
<td>$\gamma_{t,i}(\omega)$</td>
</tr>
<tr>
<td>$\gamma(\omega)$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Values to be calculated</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Gamma_S$</td>
<td>The expected total wait time of scheduled patients</td>
</tr>
<tr>
<td>$\Gamma_W$</td>
<td>The expected total wait time of walk-ins</td>
</tr>
<tr>
<td>$\Gamma_I$</td>
<td>The expected idle time of the provider</td>
</tr>
<tr>
<td>$\Gamma_O$</td>
<td>The expected overtime of the provider</td>
</tr>
<tr>
<td>$\Gamma_D$</td>
<td>The expected duration from the beginning of the session to the departure time of the last patient</td>
</tr>
<tr>
<td>$\Pi_t(k)$</td>
<td>The probability of $k$ patients waiting for services at the beginning of $t$</td>
</tr>
<tr>
<td>$\Psi_t(k)$</td>
<td>The probability of $k$ scheduled patients waiting for services at the beginning of $t$</td>
</tr>
<tr>
<td>$\Upsilon(x, \omega_0)$</td>
<td>The total cost under a schedule $x$ when scenario $\omega_0$ occurs</td>
</tr>
<tr>
<td>$\Upsilon(z, \omega)$</td>
<td>The total cost under a schedule $z$ when scenario $\omega$ occurs</td>
</tr>
</tbody>
</table>

B. Proofs of the Results

B.1. Proof of Proposition 1

Suppose that we follow an optimal policy $\pi$. Along some sample path at time $t$, $\pi$ serves a walk-in while there are still scheduled patients waiting in the system. Let $t + \Delta$ (for some $\Delta > 0$) be the first time after $t$ that $\pi$ serves a scheduled patient. We show that if we switch the service order of these two patients but
keep following \( \pi \) for other patients, we will result in a total cost that is no larger than that under \( \pi \). By keep doing this, we can obtain a policy that obeys the service order specified by the proposition and results in a total cost that is no larger than that of \( \pi \). In other words, this new policy must be optimal, completing the proof.

Now it suffices to show that the single switch above leads to a lower cost. Let \( \pi' \) be the policy that differs from \( \pi \) in this single switch but otherwise is the same. Let \( \Gamma_D^\pi + C_W \Gamma_W^\pi + C_D \Gamma_D^\pi \) denote the total cost under \( \pi \). The corresponding cost for \( \pi' \) has a subscript \( \pi' \). Suppose that at \( t \), there are \( k_1 \geq 1 \) scheduled patients and \( k_2 \geq 1 \) walk-ins in the system. It is easy to see \( \Gamma_D^\pi = \Gamma_D^{\pi'} \), because there are \( k_1 + k_2 - 1 \) patients after \( t \) no matter who is severed. Also notice that \( \Gamma_S^\pi + \Gamma_W^\pi = \Gamma_S^{\pi'} + \Gamma_W^{\pi'} \) since the total number of waiting patients does not change. Along the sample path, we have \( \Gamma_W^\pi = \Gamma_W^{\pi'} + \Delta \), then \( \Gamma_S^\pi = \Gamma_S^{\pi'} - \Delta \). It follows that \( \Gamma_S^{\pi'} + C_W \Gamma_W^{\pi'} + C_D \Gamma_D^{\pi'} = \Gamma_S^\pi + C_W \Gamma_W^\pi + C_D \Gamma_D^\pi + (C_W - 1)\Delta \), implying that \( \pi' \) which serves the scheduled patients before walk-ins incurs no additional cost than \( \pi \) (because \( C_W \leq 1 \)). Q.E.D.

### B.2. Proof of Proposition 2

By Proposition 1 and Definition 2, all time slots within an MNEP will be used to serve scheduled patients. Thus, if two schedules have the same sets of MNEPs, the time spent on scheduled patients of these two systems are exactly the same, resulting in the same distribution of walk-ins at all times. For instance, consider an MNEP \([t_{k_1}, t_{k_2}]\) for \( t_{k_2} < T \) and there are \( k \) walk-ins waiting at the beginning of this MNEP, there must be \( k + \sum_{i=1}^{t} \beta_i \) walk-ins at the end of time \( t \in [t_{k_1}, t_{k_2}] \). Similar arguments also work for the case if \( T \) belongs to some MNEP \([t_{k_1}, T]\). Q.E.D.

### B.3. Proof of Theorem 1

By Definition 2 and Proposition 2, two schedules with the same MNEPs will have same number of scheduled patients and probabilistically the same number of walk-ins at all times. It suggests that they will have same expected duration \( \Gamma_D \). Proposition 2 also shows that two schedules with the same MNEPs will have same \( \Gamma_W \), i.e., the same expected wait time of walk-ins. So we only need to consider the wait time of scheduled patients \( \Gamma_S \). Their wait time in an MNEP can be evaluated as \( [t_{k_1}, t_{k_2}] \) is \( \sum_{i=1}^{t} (t_{k_2} - t) x_i - (t_{k_2} - t_{k_1} + 1) \). Thus, \( x_i \) must follow the pattern described by the theorem. Q.E.D.

### B.4. Proof of Condition (15)

We use the following definition in our proof. The vector \( a \) is said to majorize the vector \( b \) (denoted \( a \succ b \)) if

\[
\sum_{i=1}^{k} a_i \geq \sum_{i=1}^{k} b_i, \quad \forall k = 1, 2, \ldots, n - 1,
\]

and

\[
\sum_{i=1}^{n} a_i = \sum_{i=1}^{n} b_i.
\]

Condition (15) can be equivalently written as \( x^{w/o} \succ x^{w/o} \). We note that \( x^{w/o} \) has the following pattern:

\[
x^{w/o} = (1, 1, \ldots).
\]

That is, we start from the first slot, and assign one patient to each subsequent slot and all remaining patients (if any) to slot \( T \). The “No Hole” property in Robinson and Chen (2010) implies that empty slots, if any,
will be placed towards the end in $x^{w/o}$. In other words, compared to $x^{w/o}$, $x^{w}$ “pushes” patients scheduled towards the end upfront in the schedule, leading to a “larger” vector in terms of the majorization order. Therefore, we have $x^{w} \succ x^{w/o}$. Q.E.D.

B.5. Proof of Theorem 3

By definition, $y_t \; (y_t^*)$ is number of waiting (scheduled) patients after time slot $t$. Then $y_{t-1} + \alpha_t(x_t, \omega^*) + \beta_t(\omega^*) - y_t \; (y_{t-1} - y_t \; \text{when } t > T)$ is the number of served patient at $t$. This number cannot be larger than 1 by inequality constraints in (19). And it must be 0 or 1 with integer constraints of $y_t$ and $y_t^*$. Notice that the objective function is increasing in $y_t$ and $y_t^* \; \text{so problem (19) is choosing to serve a patient or not at each time to minimize the total weighted waiting patients. It is easy to see that the optimal solution must be serving a patient if there is any, which means that problem (19) will give the same result as (18). Q.E.D.}

B.6. Proof of Proposition 4

It suffices to show that $\forall a_t, b_t, t = 1, 2, \ldots, T,$

$$Pr(\alpha_t(x_t, \omega^{*}) = a_t, \beta_t(\omega^{*}) = b_t, \forall t = 1, 2, \ldots, T) = \prod_{t=1}^{T} Pr(\alpha_t(x_t, \omega^{*}) = a_t) \cdot Pr(\beta_t(\omega^{*}) = b_t).$$

As events occurring in different slots are independent and walk-ins and scheduled patients are independent, we have

$$Pr(\alpha_t(x_t, \omega^{*}) = a_t, \beta_t(\omega^{*}) = b_t, \forall t = 1, 2, \ldots, T) = \prod_{t=1}^{T} Pr(\alpha_t(x_t, \omega^{*}) = a_t) \cdot Pr(\beta_t(\omega^{*}) = b_t).$$

Due to (21) and (22), we know that $\sum_{i=1}^{N_S} \gamma_{t,i}(\omega)z_{t,i}$ and $\sum_{i=1}^{N_S} \gamma_{s,i}(\omega)z_{s,i}$ have no overlapping terms, and thus are independent. It follows that

$$Pr(\sum_{i=1}^{N_S} \gamma_{t,i}(\omega)z_{t,i} = a_t, \beta_t(\omega) = b_t, \forall t = 1, 2, \ldots, T) = \prod_{t=1}^{T} Pr(\sum_{i=1}^{N_S} \gamma_{t,i}(\omega)z_{t,i} = a_t) \cdot Pr(\beta_t(\omega) = b_t).$$

For any $t$, we observe that

$$Pr(\sum_{i=1}^{N_S} \gamma_{t,i}(\omega)z_{t,i} = a_t) = \left(\frac{\sum_{i=1}^{N_S} z_{t,i}}{a_t}\right) \left(p_t^\omega \sum_{i=1}^{N_S} z_{t,i} - \sum_{i=1}^{N_S} z_{t,i} \right)$$

$$= Pr(\alpha_t(x_t, \omega^{*}) = a_t),$$

where the second equality is resulted from our definition of the new decision variables $z_{t,i}$ that $x_t = \sum_{t=1}^{N_S} z_{t,i}$. Q.E.D.

B.7. Proof of Theorem 2

The result follows directly from the unimodularity of matrix $U$ and Proposition 4. Q.E.D.

B.8. Proof of Theorem 3

Proof: Let $z^*$ and be the optimal solution to (T2), and thus $\Upsilon(z^*)$ is the optimal objective value. It is clear that $z = z^*$ and $u = \Upsilon(z^*)$ is a feasible solution to (T2-D). Now, let $z^{**}$ and $u^{**}$ be the optimal solution to (T2-D). It must be that $u^{**} \leq \Upsilon(z^*)$. Suppose that $u^{**} < \Upsilon(z^*)$. Then we have

$$a(z^{**})z^{**} + b(z^{**}) = \Upsilon(z^{**}) \leq u^{**} < \Upsilon(z^*),$$

where the first equality is from (27) and the second inequality follows from (28). This contradicts with that $z^*$ is the optimal solution to (T2), and therefore, it must be that $u^{**} = \Upsilon(z^*)$. Q.E.D.
B.9. Proof of Corollary 3

\( Y(z^*) \leq \pi < a(z')z^0 + b(z') \leq Y(z^0) \). Q.E.D.

B.10. Proof of Corollary 4

By (26) and (27), we have that \( a(z^0)z + b(z^0) \leq Y(z) = a(z)z + b(z) \). Q.E.D.

B.11. Proof of Theorem 4

By Corollary 3, a solution which does not satisfy a subset of (28) is not optimal. So, in Step 2, we get a solution which satisfies a subset of (28). By Corollary 4, the constraints that correspond to non-optimal solutions are redundant. Thus in Step 5, we only generate the constraints associated with the solution obtained in Step 2. We will not have an endless loop because there are only a finite number of constraints that can be added. Once the necessary constraints have been added to the program, we will have an optimal solution. The optimality of the algorithm is evident by the stopping criteria. Q.E.D.

C. Reformulation when \( C_W < 1 \)

When \( C_W < 1 \), Problem (20) can be written as,

\[
\begin{align*}
\min_z & \quad \mathbb{E}_\omega[Y(z, \omega)] \\
& \quad \sum_{t=1}^T z_{t,i} = 1 \text{ for } 1 \leq i \leq N_S, \\
& \quad z_{t,i} \in \{0, 1\} \text{ for } 1 \leq t \leq T, 1 \leq i \leq N_S,
\end{align*}
\]

where

\[
Y(z, \omega) := \left\{ \begin{array}{l}
\min_{y} \sum_{t=1}^T y_t^i + C_W \sum_{i=1}^{N_S} (y_t - y_t^i) + C_D y_T \\
y_t \geq y_{t-1} + \sum_{i=1}^{N_S} \gamma_{t,i}(\omega) z_{t,i} + \beta_t(\omega^o) - 1 & \text{for } 1 \leq t \leq T \text{ with } y_0 = 0, \\
y_t \geq y_{t-1} - 1 & \text{for } T < t \leq T, \\
y_t^i \geq y_{t-1}^i + \sum_{i=1}^{N_S} \gamma_{t,i}(\omega) z_{t,i} - 1 & \text{for } 1 \leq t \leq T \text{ with } y_0^i = 0, \\
y_t^i \geq y_{t-1}^i - 1 & \text{for } T < t \leq T; \\
y_t, y_t^i \in \mathbb{Z}_+ & \text{for } 1 \leq t \leq T.
\end{array} \right.
\]

Let \( y = \begin{bmatrix} y \\ y^i \end{bmatrix} \), \( c = (C_W, ..., C_W, C_W + C_D, C_W, ..., C_W, 1 - C_W, ..., 1 - C_W)^{tr} \) be a \( 2T \) dimensional vector where all the first \( T \) elements are \( C_W \) except for \( T \)th element and the last \( T \) elements are \( 1 - C_W \), \( U = \begin{bmatrix} U & 0 \\ 0 & U \end{bmatrix} \), \( M(\omega) = \begin{bmatrix} M(\omega) \\ M(\omega) \end{bmatrix} \), and \( d(\omega) \) be a \( 2T \) dimensional vector where the first \( T \) elements are \( \beta_t(\omega) - 1 \) and other elements are \(-1\).

Then we will have the matrix form

\[
\begin{align*}
\min_z & \quad \mathbb{E}_\omega[Y(z, \omega)] \\
Wz &= c, \\
z &\in \{0, 1\},
\end{align*}
\]
where
\[
\Upsilon(z, \omega) := \begin{cases} 
\min \ y & c'y \\
Uy \geq M(\omega)z + d(\omega), \\
y \in \mathbb{Z}_+.
\end{cases}
\]

Notice that \( U \) is also totally unimodular, so the analysis follows that in Section 5.2.

D. LP Relaxation of (T2-SAA)

If we relax the 0-1 integer constraints for \( z \) in (T2-SAA), we obtain its LP relaxation as follows.

\[
\min_{y(\omega), z} \sum_{\omega \in \Omega} \left[ c' y(\omega) \right] \quad \text{(T2-LP)}
\]

\[
Wz = e,
\]

\[
Uy(\omega) - M(\omega)z \geq d(\omega) \quad \text{for any } \omega,
\]

\[
y(\omega) \geq 0 \quad \text{for any } \omega,
\]

\[
z \geq 0.
\]

However, when \( N_S \geq 3 \) and \( |\Omega| \geq 3 \), the coefficient matrix

\[
\begin{bmatrix}
0 & 0 & 0 & \cdots & 0 & W \\
U & 0 & 0 & \cdots & 0 & M(\omega_1) \\
0 & U & 0 & \cdots & 0 & M(\omega_2) \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 0 & M(\omega_3)
\end{bmatrix}
\]

is not necessarily totally unimodular. To see that, take the \( t^{th} \) row of \( M(\omega_1), M(\omega_2) \) and \( M(\omega_3) \) and their \( ((t-1) \times N_S + 1)^{th}, ((t-1) \times N_S + 2)^{th}, ((t-1) \times N_S + 3)^{th} \) columns, then the square submatrix is

\[
\begin{bmatrix}
\gamma_{t,1}(\omega_1) & \gamma_{t,2}(\omega_1) & \gamma_{t,3}(\omega_1) \\
\gamma_{t,1}(\omega_2) & \gamma_{t,2}(\omega_2) & \gamma_{t,3}(\omega_2) \\
\gamma_{t,1}(\omega_3) & \gamma_{t,2}(\omega_3) & \gamma_{t,3}(\omega_3)
\end{bmatrix}.
\]

As \( \gamma_{t,i}(\omega) \) can either be 0 or 1, this submatrix may not be unimodular. For instance, the determinant of \( \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \) is 2. Thus, the LP relaxation of (T2-SAA) may not be exact.

E. Additional Numerical Results
Figure 5  The Optimal Schedules for Provider 2 ($T = 10$, $N_S = 9$)