IIE Transactions (2015) **47**, 1267–1286

ISSN: 0740-817X print / 1545-8830 online
DOI: 10.1080/0740817X.2015.1011355

Online appointment sequencing and scheduling

S. AYCA ERDOGAN[1,*], ALEXANDER GOSE[2] and BRIAN T. DENTON[3]

[1]*Department of Industrial and Systems Engineering, San Jose State University, San Jose, CA 95192-0085, USA*
E-mail: ayca.erdogan@sjsu.edu
[2]*Graduate Program of Operations Research, North Carolina State University, Raleigh, NC 27695-7913, USA*
[3]*Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI 48109-2117, USA*

Received December 2011 and accepted December 2014

We formulate and solve a new stochastic integer programming model for dynamic sequencing and scheduling of appointments to a single stochastic server. We assume that service durations and the number of customers to be served on a particular day are uncertain. Customers are sequenced and scheduled dynamically (online) one at a time as they request appointments. We present a two-stage stochastic mixed integer program that uses a novel set of non-anticipativity constraints to capture the dynamic multi-stage nature of appointment requests as well as the sequencing of customers. We describe several ways to improve the computational efficiency of decomposition methods to solve our model. We also present some theoretical findings based on small problems to help motivate decision rules for larger problems. Our numerical experiments provide insights into optimal sequencing and scheduling decisions and the performance of the solution methods we propose.

Keywords: Appointment scheduling, sequencing, stochastic programming, health care

1. Introduction

Many service systems provide appointments to customers in advance of their arrival. However, because service times are uncertain, the amount of time to allocate between customer arrivals is a challenging decision. Short inter-arrival times can lead to high service system utilization but at the expense of long customer wait times. Long inter-arrival times, on the other hand, tend to reduce customer waiting but at the expense of lower resource utilization. Achieving a balance between these competing criteria can be challenging because simple rules, such as longest processing time first sequencing and setting the mean service time for customer inter-arrivals, often perform poorly (Denton and Gupta, 2003; Denton *et al.*, 2007).

When the number of customers to be scheduled is known in advance, schedules can be designed using stochastic optimization models (Denton *et al.*, 2007; Gul *et al.*, 2011) or through experimentation with simulation models (Ho and Lau, 1992; Robinson and Chen, 2003) or queuing models (Soriano, 1966; Mercer, 1973; Sabria and Daganzo, 1989). However, in many service systems appointment scheduling is complicated by the fact that the exact number of customers to be scheduled is not known in advance. Instead, customers request appointments sequentially over time, and appointments are quoted *online*; i.e., sequentially at the time of each appointment request. Since rescheduling of appointments is uncommon in most service industries it is necessary to make these online scheduling decisions in such a way that schedules are adaptable to variation in customer demand.

In health care delivery systems, achieving this balance is particularly important because of the high cost of resources, including human and physical resources. In this context uncertainty in demand often arises due to the inherently uncertain nature of urgent care. In outpatient clinics, for instance, customers are often classified into groups such as *routine* and *urgent*. Routine patients are scheduled in advance, often weeks or months in advance. Urgent patients, on the other hand, are scheduled on much shorter notice, typically days or hours in advance of the first patient arrival and may have a higher priority for service. Such patients are often referred to as *add-ons*. Furthermore, due to the nature of urgent patients, the exact number to be scheduled is not known with certainty. Therefore, routine appointment scheduling must be done in a way that anticipates the potential future need to schedule additional urgent patients.

In this article we describe a stochastic integer programming model for dynamic sequencing and scheduling of appointments to a single stochastic server. The model is a generalizable representation of the appointment scheduling process for many kinds of service systems (e.g., consulting services, visa services, accounting services). Appointment

*Corresponding author

requests are received sequentially, one at a time, for a given future day of service. Requests are probabilistic and therefore the exact number of requests that will be received is not known with certainty. At the time of each appointment request, the scheduler (e.g., clinical assistant in primary care clinic or experienced nurse in an outpatient surgery center) must decide on the appointment time to quote to the customer. By setting the appointment time for each customer the scheduler sets both the sequence of arrivals and the inter-arrival times between customers. In this article we study how to simultaneously optimize both of these decisions when they must be made sequentially with imperfect information about the total demand on the system. The solutions resulting from the model we present can be used to construct a scheduling template that defines the appointment time to assign for each customer request. The objective is to minimize a weighted sum of the expected cost of direct waiting and waiting until time to appointment and overtime. Direct waiting is the time spent waiting beyond the assigned appointment time, whereas waiting until *time to appointment* is the time spent waiting from the start of the day (earliest possible appointment time) and the appointment time. The latter waiting time is relevant for customers who have an urgent need for access such as urgent surgeries that arise on short notice and need to be incorporated into the daily schedule.

We discuss the special structure of our model and several ways to improve the computational efficiency of the L-shaped method to solve our model. We present some theoretical findings based on a special case of the problem and use these to provide some insight into optimal scheduling decisions for larger problems. We present a series of numerical experiments that provide insights into optimal sequencing and scheduling decisions as well as the performance of the solution methods we propose. Drawing on our computational experiments we classify problems into those that are easy to solve and those that are computationally challenging.

The remainder of this article is organized as follows: In Section 2 we provide background and literature review on appointment sequencing and scheduling. In Section 3 we present a detailed problem definition and model formulation. Section 4 describes structural properties of the model and solution methodology, and Section 5 presents the experimental results. Finally, in Section 6 we discuss our main findings and future research directions.

2. Background and literature review

Since much of the literature is in the context of health care we use the terms *patient* and *customer* interchangeably. Most of the previous work on dynamic appointment scheduling has assumed a fixed First-Come-First-Served (FCFS) queue discipline. Figure 1 illustrates the evolution of an online appointment schedule over time for a specific example in which up to five customers are scheduled. Figure 1(a) illustrates the FCFS policy in a dynamic scheduling environment. Figure 1(b) illustrates the more general case, which we consider in this article, in which the sequence is not fixed *a priori*. Note that customers are scheduled in order of their appointment requests; however, their appointment times on the day of service do not necessarily follow that order.

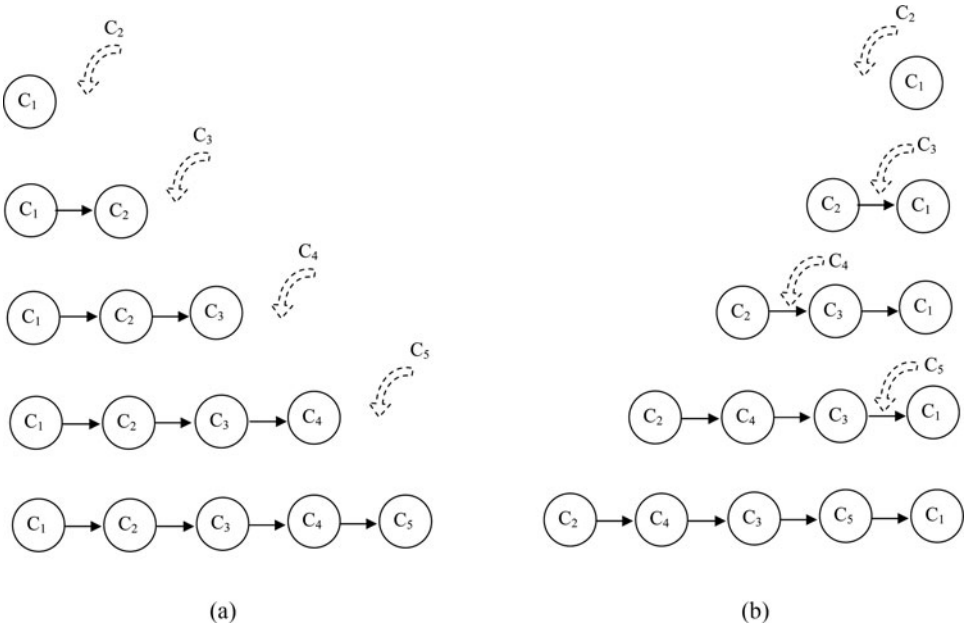

Fig. 1. Illustration of the online scheduling problem for scheduling up to five customers. Figure (a) illustrates the case in which the sequence of appointments is FCFS and (b) illustrates the general case in which the FCFS sequence is relaxed.

We study the general online scheduling problem because there are a number of health care environments in which this problem arises. Given the dynamic nature of most scheduling environments, in practice schedulers must consider the relative importance of customers when assigning appointment times. Since scheduling is done sequentially and rescheduling is uncommon in most service systems, sequencing decisions are an important part of setting appointments. When each sequencing and scheduling decision is made, the possibility of future uncertain arrivals of patients, perhaps with varying priority for service, must be considered.

In the static (off-line) context, scheduling appointments with multiple patient classes has received recent attention from several researchers. Previous studies have considered patient classifications according to characteristics such as new/returning patients, child/adult patients, or according to service durations (e.g., high versus low variance). In the context of surgery scheduling, for example, surgeries are often classified in two categories: elective and urgent. For elective cases, surgery may be planned well in advance (e.g., months) to be performed on a future date. For non-elective cases, on the other hand, the surgery is unanticipated. These cases must be *worked in* to the existing schedule, either by using intentionally reserved or otherwise available space in the schedule or by creating space by canceling previously scheduled elective cases.

In some health care environments threshold policies are applied. According to these policies, lower-priority patients (outpatients) are scheduled until a capacity threshold is reached. Remaining capacity is reserved for higher-priority patients that may arrive in the future. For example, Green *et al.* (2006) considered appointment scheduling in the context of a diagnostic medical facility in the presence of two types of demand, inpatients and outpatients, both of which must be served by the same resources. They formulated a Markov Decision Process (MDP) model and used it to determine dynamic priority rules for admitting patients. An alternative strategy, used by some hospitals, is to allocate separate capacity for emergencies and add-ons. This is common in the context of surgical practices where one or more operating rooms (ORs) may be reserved for surgeries that arise on short notice. Another strategy is to reserve slack time in the schedule for urgent patients (Gerchak *et al.*, 1996; Klassen and Rohleder, 2003; Torkki *et al.*, 2006).

Wang (1993) studied a dynamic scheduling problem in which an additional customer is scheduled after an initial batch of customers has been scheduled. He used phase-type distributions to investigate the transient solution of a Markovian server to determine the optimal start times for each customer. To find the appointment time of the new customer, the schedule was divided into intervals according to the currently scheduled appointments, and a set of nonlinear equations was solved for each interval. The placement of the new customer was determined by the interval that has the minimum objective function value after an initial schedule had been developed. However, the author assumed a single additional customer and did not attempt to find the optimal schedule in light of the possibility of additional customer arrivals, which is the problem considered in this article.

Cayirli *et al.* (2006) developed a simulation model to determine the sequence and schedule for new and returning patients. The authors tested several sequencing rules including FCFS, alternating between new and returning patients, sequencing new patients at the beginning, and sequencing returning patients at the beginning. In addition to these sequencing rules, several scheduling rules to determine the appointment allowances were tested. These rules included allocating equal intervals between patients, double-booking the first two patients (Bailey's Rule), and scheduling two patients at a time with equal intervals. They concluded that sequencing decisions have more impact on the performance of the system than the appointment scheduling rules. In another study, Cayirli *et al.* (2008) considered different environmental characteristics such as no-show rates, the ratio of new patients to returning patients, and walk-ins. They concluded that FCFS is not necessarily optimal when there are multiple patient classes. They found that different sequencing and scheduling rules should be selected depending on the environmental characteristics. Unlike previous studies based on simulation models that compare the performances of predetermined sequencing and scheduling rules we aim to find optimal dynamic appointment schedules using a novel two-stage stochastic integer programming formulation of the multi-stage decision process.

More recently, Zonderland *et al.* (2010) studied the trade-off between cancelation of scheduled elective surgeries to accommodate urgent arrivals and the unused OR time that is reserved for uncertain urgent surgeries. The authors used an infinite-horizon MDP to determine the number of slots to be reserved for urgent arrivals. They found that when the cost of canceling elective surgeries is higher than the cost of OR idle time, the optimal policy is to reserve appointment slots for a certain number of urgent arrivals in advance but postpone the remaining urgent surgeries. They found that when the cost of OR idle time is high, the optimal policy is to cancel elective surgeries to accommodate urgent surgeries.

A number of studies have considered dynamic scheduling with the aim of finding the optimal daily scheduling policy in the presence of no-shows and cancelations (Hassin and Mendel, 2008; Kolisch and Sickinger, 2008; Muthuraman and Lawley, 2008; Robinson and Chen, 2010; Lin *et al.*, 2011). Liu *et al.* (2010) also studied dynamic appointment scheduling, however, with the aim of finding the optimal future appointment day depending on the no-show probability of the requesting patient. In these studies, the patients were classified according to no-show probabilities. One exception is the work by Kolisch and Sickinger (2008) that also considered different patient classes including outpatient, inpatient, and emergency. The service durations

in the above studies were assumed to be either deterministic (Kolisch and Sickinger, 2008; Robinson and Chen, 2010) or exponential (Hassin and Mendel, 2008; Muthuraman and Lawley, 2008), which is not a limitation in our model presented in this article.

The studies by Muthuraman and Lawley (2008), Zeng *et al.* (2010), and Lin *et al.* (2011) are most relevant to this article. In Muthuraman and Lawley (2008), the authors implemented a myopic policy that schedules patients to slots sequentially as requests are received, until the profit function of revenue and waiting and overtime costs starts decreasing. Later, Zeng *et al.* (2010) developed heuristics for the problem defined in Muthuraman and Lawley (2008) that also considers heterogenous patient types depending on no-show behavior. In Lin *et al.* (2011), the authors developed an MDP to generate the optimal appointment scheduling of patients to time slots while considering the dynamic nature of appointment requests (i.e., add-ons). The solution is divided into two parts: the off-line part that determines the optimal schedule before appointment requests start and the online part that retrieves the optimal schedule and implements the scheduling decision as appointments arrive. The authors used backward induction to find the optimal solution for small instances and Approximate Dynamic Programming (ADP) methods that utilize state aggregation and simulation for larger instances. They found that a "semi-optimal policy", which accepts the patients with high no-show probability until demand reaches the point at which over-booking becomes necessary and transfers the additional high no-show probability patients to later days after overbooking, performs better than the "myopic policy", which maximizes the profit function that rewards the patients served and penalizes overtime and waiting time.

Our work differs from Lin *et al.* (2011) in several ways: first, we do not discretize the appointment scheduling process. In addition to waiting time our model also considers a cost associated with a fixed time (e.g., start of day). This allows us to consider urgent services (e.g., emergencies) and reservation of capacity during the day to accommodate these uncertain requests. Second, we formulate the problem as a stochastic integer program and focus on a method for obtaining optimal solutions or tight optimality gaps. The ability to compute optimality gaps is one of the major benefits of our math programming approach (versus ADP methods, such as those used by Lin *et al.* (2011), for which there is very little known about how to compute error bounds). The model by Lin *et al.* (2011) also is not restricted by service time distributions. However, it computationally relies on numerical integration and is viable only for certain distributions, such as exponential and gamma distributions.

Another related study is that of Erdogan and Denton (2011), which presented two stochastic programming models for two variants of the appointment scheduling problem. The first model was a static scheduling problem with no-shows. The second model was a dynamic appointment scheduling problem in which the customers are sequentially allocated to an appointment time as they request appointments. The appointment times are allocated on an FCFS basis. In contrast, in this article we relax the assumption of an FCFS sequence and we consider the total waiting (direct and time to appointment) cost measured from the start of the day. These differences lead to a unique discrete and stochastic model that is more realistic for service systems that serve customers with varying priority (e.g., hospital-based surgical practices, urgent care clinics) but also much more challenging to solve. The dynamic appointment scheduling problem assuming FCFS sequence introduced in Erdogan and Denton (2011) was formulated as a multi-stage stochastic program and solved with the nested Bender's decomposition method via decomposing the problem into several Linear Programs (LPs). The LPs at each stage had special structures such that a solver was not required to find the optimal solution. The master LPs at each iteration were solved using CPLEX or alternatively with an algorithm that exploited the geometry of the simple two-variable structure, and subproblem LPs were solved without a solver using dual information. Relaxing the FCFS assumption, on the other hand, introduced a sequencing aspect to the dynamic scheduling problem. This required the introduction of binary sequencing variables, transforming the model to a mixed-integer stochastic program. Thus, it required fundamentally different solution techniques such as a branch-and-cut algorithm to solve the mixed-integer master problems at each stage during the decomposition as opposed to the methods used in Erdogan and Denton (2011). The new problem presented in this article also inherits the multi-stage structure to capture the dynamic nature of the problem caused by each appointment request. However, in order to avoid solving a mixed-integer program at each iteration and at each stage, we present a compact re-formulation of the model to represent a multi-stage problem as a two-stage stochastic Mixed-Integer Program (MIP), which eliminates the need to use a nested decomposition method. This allows the L-shaped method to be used to decompose the problem into an MIP in the first stage and an LP in the second stage; moreover, this allows us to exploit the special structure of the problem in several ways that we describe in Section 4.

3. Model formulation

We begin by presenting a standard model for the static appointment scheduling problem (Denton and Gupta, 2003). The static problem as described in the previous section aims to find the optimal start times for a given number of customers, n, to visit a stochastic server. Service times are random variables and the objective is to minimize a weighted sum of expected customer waiting time and expected overtime with respect to an established session length, d.

Commonly considered criteria include customer waiting time, server idle time, and overtime, which can be written as follows:

$$
\begin{aligned}
w_1(\omega) &= 0,\\
w_i(\omega) &= (w_{i-1}(\omega) + Z_{i-1}(\omega) - x_{i-1})^+, i = 2, \ldots, n,\\
s_i(\omega) &= (-w_{i-1}(\omega) - Z_{i-1}(\omega) + x_{i-1})^+, i = 2, \ldots, n,\\
\ell(\omega) &= (w_n(\omega) + Z_n(\omega) + \sum_{i=1}^{n-1} x_i - d)^+.
\end{aligned}
$$

The variable $w_i(\omega)$ denotes waiting time of customer i, $s_i(\omega)$ denotes server idle time immediately prior to customer i's arrival, x_i denotes the customer allowance (inter-arrival time between customer i and $i+1$), and $Z_i(\omega)$ denotes the random service duration for customer i under random duration scenario ω. (Note that the expression $(\cdot)^+$ indicates $max(\cdot, 0)$.) The optimization problem can be written as

$$
\min_{\mathbf{x}} \left\{ \sum_{i=1}^{n} (c_i^w E_\omega[w_i(\omega)] + c^s E_\omega[s_i(\omega)]) + c^\ell E_\omega[\ell(\omega)] \right\}, \tag{1}
$$

where c^w, c^s, and c^ℓ denote the costs of waiting time, idle time, and overtime, respectively.

In the dynamic scheduling context, appointment decisions are made one at a time as customers request appointments. Figure 2 depicts a simple case with two customers. The first customer requests an appointment with probability one and the second customer requests with probability q (with probability $1-q$ a second customer does not request an appointment). We assume $d = 0$; thus, overtime corresponds to makespan, and Z_1 and Z_2 are independent and identically (i.i.d.) distributed service durations. For this special case, Erdogan and Denton (2011) proved that it is optimal to schedule customers in FCFS order (as opposed to scheduling the second (add-on) customer first referred to as Add-On-First-Served (AOFS)) when the waiting costs for two customers are also identical. The model we consider in this article is more general than the model discussed in Erdogan and Denton (2011). We use it to establish conditions under which FCFS or AOFS may be optimal.

We formulate the general online appointment sequencing and scheduling problem as a stochastic MIP with binary decision variables representing patient sequencing decisions and continuous decision variables representing inter-arrival times and appointment times. The appointment scheduling process is as follows. Customers request appointments for a specific day of service and requests arise probabilistically over time up any time prior to the day of service until some cutoff time at which the schedule is closed (e.g., 5 pm the day before the day of service) or a maximum of n appointments is reached. Customers are quoted their appointment times online as requests arise over time. The sequence of appointments may change over time as the appointment schedule evolves; however, once an appointment time is quoted for a given customer it cannot be changed. The sequential nature of this process can be formulated as a multi-stage stochastic program with stages representing each customer request. We formulate this multi-stage problem as a two-stage stochastic MIP with constraints that enforce non-anticipativity of the sequence of appointment scheduling decisions. We use the following notation, where upper case indicates random variables and boldface indicates vectors.

Model parameters:

- n : number of customers to be scheduled
- ω : index for service duration scenarios
- p_j : probability of exactly j customers requesting an appointment

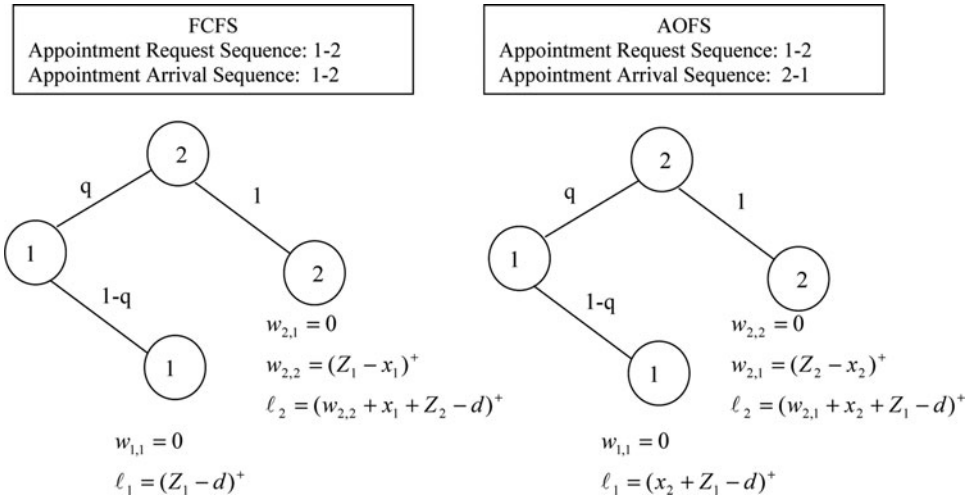

Fig. 2. Example of the dynamic scheduling problem for scheduling two customers according to FCFS and add-on first served sequencing rules ($w_{j,i}$ represents waiting time of customer i when there are j customers in the system).

$\mathbf{Z}(\omega)$: vector of random service durations for n customers
d : session length to complete all customers before overtime occurs
$\mathbf{c}^w$: vector of direct waiting time cost coefficients for n customers
$\mathbf{c}^a$: vector of appointment time (or waiting until time to appointment) cost coefficients for n customers
c^ℓ : cost coefficient for overtime
c^s : cost coefficient for idle time

Decision variables:

$o_{j,i,i'}$: binary sequencing variable where $o_{j,i,i'} = 1$ if customer i immediately precedes i' at stage j, and $o_{j,i,i'} = 0$ otherwise (first-stage decision variable)
$x_{j,i,i'}$: time allowance for customer i given that i immediately precedes i' (appointment inter-arrival time for customer i and i') at stage j (first-stage decision variable)
$a_{j,i,i'}$: arrival time of customer i', given that i immediately precedes i' at stage j (first-stage decision variable)
$w_{j,i,i'}(\omega)$: waiting time of customer i' given that customer i immediately precedes i' at stage j under duration scenario ω (second-stage decision variable)
$s_{j,i,i'}(\omega)$: server idle time between customer i and i', given that i immediately precedes i' at stage j under duration scenario ω (second-stage decision variable)
$\ell_j(\omega)$: overtime at stage j with respect to session length d under duration scenario ω (second-stage decision variable)

The index j denotes the stage of the decision-making process, which is defined by the arrival of customer j's appointment request. The decision variables defined above are denoted by vectors $\boldsymbol{o}$, $\boldsymbol{x}$, $\boldsymbol{a}$, $\boldsymbol{w}$, $\boldsymbol{s}$, which are sequence-dependent at each stage. Furthermore, the sequence may change from one stage to the next as customer requests arrive. This is due to the fact that in a given stage, when a new customer requests an appointment, the customer may be scheduled between two previously scheduled customers. For instance, as depicted in Fig. 1(b), at stage $j = 2$, two customers have already requested appointments and the assigned sequence is 2–1. When the next customer requests an appointment, at stage $j = 3$, the new sequence could be 2–3–1 if customer 3 is sequenced between customers 2 and 1. Thus, one of the previously established immediate precedence relationships in the sequence might be broken at a later stage. The probability of having j customers request appointments, p_j, can be written as follows:

$$p_j = (1 - q_{j+1}) \prod_{i=1}^{j} q_i, \text{ for all } j = 1, \ldots, n-1,$$

$$p_n = \prod_{i=1}^{n} q_i,$$

where q_i is the probability that customer i requests an appointment, given that customer $i-1$ has requested. In other words, it is the probability that at least an additional customer will request an appointment, given that $i-1$ customers have already requested appointments. Note that we assume that $q_1 = 1$; i.e., there is always at least one customer in the system.

In our model formulation, for each stage j, we include two dummy customers, customer 0 and customer $j+1$. Customer 0 is always at the beginning of the sequence, and customer $j+1$ is always at the end of the sequence. This simplifies the formulation by ensuring each customer (except dummy customers) is preceded and followed by another customer. A valid sequence of appointments at any stage j is one that begins with the dummy customer 0 and ends with the dummy customer $j+1$. Between successive stages, the sequence of customers does not change, except for the possibility that the jth customer will be inserted between two customers in the previous stage's sequence or appear immediately before the dummy customer $j+1$.

The problem described above is by nature a multi-stage decision process, with the customer appointment requests defining the stages. However, multi-stage stochastic integer programs are widely regarded as computationally intractable. Therefore, we formulate our model as a two-stage stochastic program (2-SLP) in which binary (sequencing) decisions are dependent on the appointment request scenario and appear in the first stage. We use a novel set of constraints to enforce non-anticipativity of the appointment sequencing decisions across stages. This formulation has the benefit of a continuous and convex recourse function in the second stage, which allows for the application of the L-shaped method, which we discuss in Section 4.

The 2-SLP formulation of the on-line appointment sequencing and scheduling problem can be written as follows:
(D-ASSP)

$$\min \sum_{j=1}^{n} p_j \left[\sum_{i=1}^{j} \sum_{i'=1}^{j} c^a_{i'} a_{j,i,i'} \right] + Q(\mathbf{o}, \mathbf{x}, \mathbf{a}), \tag{2}$$

s.t.

$$\sum_{i'=1}^{j} o_{j,0,i'} = 1, \qquad \forall j, \tag{3}$$

$$\sum_{i'=1}^{j} o_{j,i',j+1} = 1, \quad \forall j, \tag{4}$$

$$\sum_{\substack{i'=1 \\ i \neq i'}}^{j+1} o_{j,i,i'} = 1, \qquad \forall j, i = 1, 2, \ldots, j, \tag{5}$$

$$\sum_{\substack{i'=0 \\ i \neq i'}}^{j} o_{j,i',i} = 1, \qquad \forall j, i = 1, 2, \ldots, j, \tag{6}$$

$$\sum_{\substack{i'=0\\ i\neq i'}}^{j+1}\sum_{i=0}^{j+1} o_{j,i,i'} = j+1, \qquad \forall j, \quad (7)$$

$$o_{j,i,j} + o_{j,j,i'} \geq 2(o_{j-1,i,i'} - o_{j,i,i'}), \qquad \forall j, \forall i, i' < j, \quad (8)$$

$$x_{j,i,i'} \leq M_1 o_{j,i,i'}, \qquad \forall j, i, i', \quad (9)$$

$$a_{j,i,i'} \leq M_1 o_{j,i,i'}, \qquad \forall j, i, i', \quad (10)$$

$$\sum_{\substack{i'=1\\ i\neq i'}}^{j+1} x_{j,i,i'} = \sum_{\substack{i'=1\\ i\neq i'}}^{j+1} a_{j,i,i'} - \sum_{\substack{i'=0\\ i\neq i'}}^{j+1} a_{j,i',i}, \qquad \forall j, \ i, \quad (11)$$

$$\sum_{\substack{i'=1\\ i\neq i'}}^{j+1} a_{j,i',i} = \sum_{\substack{i'=1\\ i\neq i'}}^{j} a_{j-1,i',i}, \qquad \forall j, \ i, \quad (12)$$

$$x_{j,i,i'}, \ a_{j,i,i'} \geq 0, \ o_{j,i,i'} \in \{0,1\}, \qquad \forall j, i, i', \quad (13)$$

where

$$Q(\mathbf{o}, \mathbf{x}, \mathbf{a}) = E_\omega[Q(\mathbf{o}, \mathbf{x}, \mathbf{a}, \omega)], \quad (14)$$

and $Q(\mathbf{o}, \mathbf{x}, \mathbf{a}, \omega)$ defines the second-stage scenario subproblem:

$$\min \quad \sum_{j=1}^{n} p_j \left[\sum_{i=1}^{j}\sum_{i'=1}^{j} (c_{i'}^{w} w_{j,i,i'}(\omega) + c^{s} s_{j,i,i'}(\omega)) + c^{\ell} \ell_j(\omega) \right], \quad (15)$$

s.t.

$$w_{j,i,i'}(\omega) \leq M_2(\omega) o_{j,i,i'}, \qquad \forall i, i', j, \omega, \quad (16)$$

$$s_{j,i,i'}(\omega) \leq M_3(\omega) o_{j,i,i'}, \qquad \forall i, i', j, \omega, \quad (17)$$

$$-\sum_{i'=1}^{j} w_{j,i',i}(\omega) + \sum_{i'=1}^{j} w_{j,i,i'}(\omega) - \sum_{i'=1}^{j} s_{j,i,i'}(\omega) = Z_i(\omega) - \sum_{i'=1}^{j} x_{j,i,i'}, \qquad \forall i, j, \omega, \quad (18)$$

$$\ell_j(\omega) \geq \sum_{i=1}^{j}\sum_{i'=1}^{j} s_{j,i,i'}(\omega) + \sum_{i=1}^{j} Z_i(\omega) + \sum_{i'=1}^{j} x_{j,0,i'} - d, \quad \forall j, \omega, \quad (19)$$

$$w_{j,i,i'}(\omega), s_{j,i,i'}(\omega) \geq 0, \qquad \forall j, i, i', \omega, \quad (20)$$

$$\ell_j(\omega) \geq 0, \qquad \forall j, \omega. \quad (21)$$

We refer to the above problem as the Dynamic Appointment Sequencing and Scheduling Problem (D-ASSP). In our two-stage formulation, the vectors of time allowances, **x**, appointment times, **a**, and binary sequencing variables **o**, are first-stage decisions. The random service time durations vector, $\mathbf{Z}(\omega)$, with support $\Xi \in \Re^n$, depends on outcomes indexed by $\omega \in \Omega$. Customer waiting time, $\mathbf{w}(\omega) \in \Re^{n^3}$, server idle time $\mathbf{s}(\omega) \in \Re^{n^3}$, and overtime $\ell(\omega) \in \Re^n$ denote the second-stage (recourse) decisions made after the first-stage decisions and the observation of random service duration scenario, ω. Service times for all customers scheduled on a particular day are observed simultaneously at the start of the day. Although this is an approximation of the true sequential observation process, it results in no inaccuracy in the model due to the assumption that customers are not rescheduled on the day of service.

The first-stage constraints in the above formulation define feasible appointment schedules with respect to sequencing decisions. In the above formulation, constraint set (3) ensures that dummy customer 0 is always at the beginning of the sequence at each stage. Constraint set (4) ensures that dummy customer $j+1$ is always at the end of the sequence at stage j. Constraint sets (5) and (6) imply that each (non-dummy) customer is part of a feasible sequence i.e., each customer comes before another and followed by another within a given stage. Constraint set (7) ensures that $j+1$ precedence relationships exist at each stage j including the precedence relationships with dummy customers.

Treating $\boldsymbol{x}$, $\boldsymbol{a}$, and $\boldsymbol{o}$ as first-stage decisions implies that they are made with perfect information about the number of future appointment arrivals. To correct this we add non-anticipativity constraints (Birge and Louveaux, 1997). Standard non-anticipativity constraints require that decisions are the same for any decisions that share the same history of the appointment request process. However, this typically results in a very large number of constraints. Instead, we use a problem-specific set of constraints, constraint set (8). These constraints require each stage's sequencing decisions are made only based on the information available at that stage and that they are feasible with respect to the sequencing decisions made in the earlier stages. We provide the following proposition to prove the validity of the D-ASSP formulation.

Proposition 1. *A sequence of appointments at stage* $j = 1, \ldots, n$ *is valid if and only if Constraints (3) to (8) are satisfied.*

Proof. See the Appendix. ■

Constraints (9) and (10) ensure that corresponding time allowances, $x_{j,i,i'}$, and appointment times, $a_{j,i,i'}$, may be non-zero only if customer i precedes i' at stage j. M_1 is chosen to be sufficiently large to be an upper bound on the optimal values of decisions $x_{j,i,i'}$ and $a_{j,i,i'}$. Constraint (11) implies that the allowance for each customer is equal to the time difference between the appointment time of that customer and the appointment time of the following customer in the sequence. Constraint (12) enforces the appointment time for a customer to be preserved in the future stages. In

other words, Constraint (12) ensures that the arrival time of customer i remains the same at each stage even though his or her position in the sequence may change.

In the second stage, constraint sets (16) and (17) ensure that waiting and idling times, $w_{j,i,i'}$ and $s_{j,i,i'}$, will be non-zero only if customer i directly precedes customer i' at stage j. Constraint set (18) determines the sequence-dependent waiting times for each customer. A customer's waiting time depends on the waiting time, allowance, and the service time of the preceding customer. Similarly, constraint set (19) determines the overtime at each stage j, which depends on the total idle time between customers and total service durations of all customers. Note that the expression $\sum_{i'=1}^{j} x_{j,0,i'}$ denotes the time before the first customer's arrival at stage j and the expression $\sum_{i=1}^{j}\sum_{i'=1}^{j} s_{j,i,i'}(\omega)$ represents the total idle time between customers.

4. Problem structure and solution methodology

In this section we first present a special case of the problem that provides some insight into the tradeoff between the cost of delaying customers (waiting until time to appointment) and the stochastic nature of online arrivals. Next, we discuss several special properties of our model that can be exploited to achieve computational efficiency.

4.1. *A special case*

Consider the case in which $n = 2$, with one routine customer requesting an appointment with probability one, followed by an urgent add-on customer that requests an appointment with probability q. We analyze this case to give insight into the patterns we observe in the optimal schedules for larger problems studied in Section 5. We begin by assuming the two customers have identical deterministic services times. We define the two alternative sequencing decisions as follows:

FCFS: The first customer in the appointment request sequence is scheduled to arrive first. A second (add-on) customer requests an appointment with probability q, and this customer is scheduled to arrive after the first customer.

AOFS: The second (add-on) customer in the appointment request sequence requests an appointment with probability q after the first customer requests an appointment. However, the second customer is scheduled to arrive first.

We impose the deterministic service time assumption by defining $Z_i = \mu$ with probability one for $i = 1, 2$. The following assumptions are made about the time to appointment and direct waiting costs. First, we assume $c_2^a = c_2^w$; i.e., that the cost of waiting for the add-on customer is the same whether it is direct or time to appointment waiting. Second, we assume $c_1^a = 0$; i.e., there is no cost of waiting until the time-to-appointment for the routine customer. We further assume the session length $d = 0$; i.e., we consider the common case of minimizing a weighted sum of makespan and indirect waiting costs. These assumptions are consistent with many health care environments such as surgery and primary care practices in which urgent add-on customers have high waiting costs, and routine customers arrive at their assigned appointment time and therefore only accrue costs for direct waiting.

The decision process for the appointment scheduling problem described above is illustrated in Fig. 3. We note that when the sequence is FCFS it is clearly optimal to assign the first customer to arrive at time 0 since a non-zero appointment time would result in unnecessary additional waiting cost. When the sequence is AOFS, on the other hand, we denote the arrival time of the second (routine) customer by a_1. It is straightforward to show that $a_1 = 0$ if $qc_1^w \le (1-q)c^\ell$ and $a_1 = \mu$ if $qc_1^w > (1-q)c^\ell$. This follows from the fact that the optimal appointment time for the AOFS, denoted by a_1^{AOFS}, can be expressed as

$$a_1^{AOFS} = \operatorname{argmin}\{q(2c^\ell\mu + c_1^w(\mu - a_1)^+) + (1-q)c^\ell(a_1 + \mu)\}.$$

Given this, the following decision rules are optimal:

- AOFS is optimal if
  - $qc_1^w \le (1-q)c^\ell$ and $c_1^w \le c_2^a$, which corresponds to the case where customers are double booked at time 0,
    or
  - $qc_1^w > (1-q)c^\ell$ and $qc_2^a \ge (1-q)c^\ell$.
- FCFS is optimal if
  - $qc_1^w \le (1-q)c^\ell$ and $c_1^w > c_2^a$, which corresponds to the case where customers are double booked at time 0,
    or
  - $qc_1^w > (1-q)c^\ell$ and $qc_2^a < (1-q)c^\ell$.

The above decision rule is consistent with intuition in a number of ways. For example, if q and c_2^a are low—i.e., the likelihood of the add-on customer arriving and the direct cost of waiting if the add-on customer does arrive are low—then it tends to be optimal to schedule the add-on customer after the routine customer. Conversely, when the add-on customer is likely to request an appointment and/or the cost of waiting until appointment is high, it tends to be optimal to schedule the add-on customer first in the sequence.

The above example helps provide insight into the tradeoff involved in the optimal sequencing decision for the routine and add-on customer. Next, we consider the case in which service times are stochastic. We assume that the service times of the routine and add-on customers, Z_1 and Z_2, respectively, are independent and identically distributed (iid) random variables with mean μ. All other assumptions are the same as the above example. The optimal objective

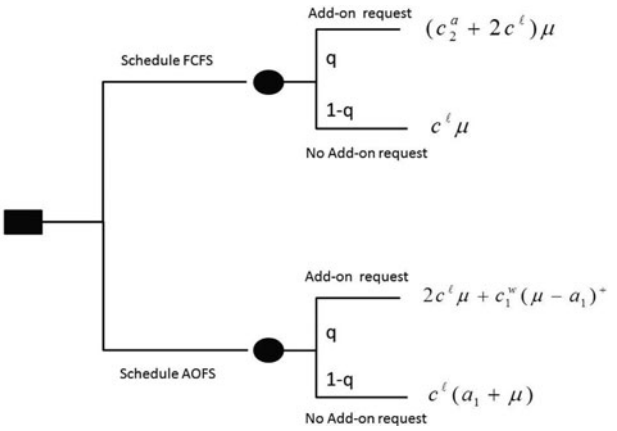

Fig. 3. Illustration of the decision process for the case of $n = 2$ with one routine customer, and one add-on customer arriving with probability q.

function for FCFS can be written as follows:

$$F^*_{FCFS} = E_Z[(1-q)c^\ell Z_1 + q(c_2^a Z_1 + c^\ell Z_1 + c^\ell Z_2)] = (1-q)c^\ell\mu + q(c_2^a + 2c^\ell)\mu. \quad (22)$$

The expression for F^*_{FCFS} follows from the fact that it is optimal to schedule the routine customer at time 0. Note that the optimal FCFS solution is obtained by assuming the appointment time for the add-on customer is also zero, which follows trivially from the fact that direct waiting and waiting until appointment time are equal ($c_2^a = c_2^w$). The optimal objective function for AOFS can be written as follows:

$$\begin{aligned} F^*_{AOFS} &= E_Z[(1-q)c^\ell(a_1^{AOFS} + Z_1) + q(c_1^w(Z_2 - a_1^{AOFS})^+ \\ &\quad + c^\ell((Z_2 - a_1^{AOFS})^+ + a_1^{AOFS} + Z_1))] \\ &= c^\ell a_1^{AOFS} + c^\ell\mu + q(c_1^w + c^\ell)E_Z[(Z_2 - a_1^{AOFS})^+]. \end{aligned} \quad (23)$$

Using the above definitions we state the following proposition.

Proposition 2. *If Z_1 and Z_2 are i.i.d. and*

$$c_2^a \geq c_1^w \quad (24)$$

then the optimal sequence is AOFS.

Proof. See the Appendix. ■

This proposition states that Equation (24) is a sufficient condition for the optimal sequence to be AOFS. Although we prove this only for the special case of $n = 2$ in this section, we provide evidence in Section 5 that this simple condition provides a useful rule of thumb for larger problems.

4.2. *Special structure of the model*

The D-ASSP model formulation expressed in Equations (2) to (21) is a two-stage stochastic MIP, with binary decisions in the first stage and a continuous second-stage LP. In this section we present several properties of the dynamic appointment sequencing and scheduling model that we introduced in the previous section.

4.2.1. *Solution to scenario subproblems*

The D-ASSP model has *complete recourse* since the recourse problem, $Q(\mathbf{o}, \mathbf{x}, \mathbf{a}, \omega)$ has a feasible solution for any choice of $\mathbf{o}, \mathbf{x}, \mathbf{a}$. Given a first-stage solution with a feasible sequence and feasible appointment times and allowances, the optimal second-stage solution can be computed easily by computing the corresponding waiting time, idle time, and overtime variables. For instance, assuming that a first-stage solution to a three-customer problem is 2–3–1. Table 1 includes the sequences and corresponding second-stage variables at each of the three decision stages based on this first-stage solution. At stage 1, the system only has customer 1 and the two dummy customers (customers 0 and 2). At stage 2, customer 2 is also included, and according to the optimal sequence, this customer precedes customer 1 since the optimal sequence at stage 2 is 0–2–1–3. Knowing that customer 2 precedes customer 1, the waiting time of customer 1 can be found using the waiting time of preceding customer 2. Similarly, at the third stage, given that the sequence is 0–2–3–1–4, the waiting time of customer 1 can be found by using the waiting time of the preceding customer 3, which in turn is determined by the waiting time of customer 2. In our implementation of the L-shaped method, at each iteration, for each scenario, the subproblem solution is obtained as described above. This eliminates the need to solve the subproblem LP (e.g., using the simplex method). The optimal basis of the primal problem can be used to directly compute the dual solution. Thus, much less computational effort is expended in computing the optimal solution to the second-stage subproblems.

4.2.2. *Big M values*

Both the first-stage and second-stage problems given in Section 3 have *big M* values in their formulations. These values must be chosen carefully because having unnecessarily large M values can cause computational disadvantages in solving MIPs since they lead to a weak LP relaxation.

In our formulation, big M values provide upper bounds on the values of first-stage decision variables, $\boldsymbol{x}$ and $\boldsymbol{a}$, and second-stage decision variables, $\boldsymbol{w}$ and $\boldsymbol{s}$. For the first stage constraints, (9) and (10), we let

$$M_1 = \max_{\omega} \sum_{i=1}^{n} Z_i(\omega). \quad (25)$$

Table 1. Second-stage variables for waiting, idling, and overtime for a three-customer problem given a fixed set of sequencing decisions is known

Stage	*Sequence*	*Corresponding second-stage variables*
1	0–1–2 (0 and 2 dummy)	$w_{1,0,1} = (Z_0 - x_{1,0,1})^+$
		$w_{1,1,2} = (w_{1,0,1} + Z_1 - x_{1,1,2})^+$
		$s_{1,0,1} = (x_{1,0,1} - Z_0)^+$
		$s_{1,1,2} = (x_{1,1,2} - w_{1,0,1} - Z_1)^+$
		$\ell_1 = (s_{1,0,1} + s_{1,1,2} + Z_0 + Z_1 - x_{1,0,1} - d)^+$
2	0–2–1–3 (0 and 3 dummy)	$w_{2,0,2} = (Z_0 - x_{2,0,2})^+$
		$w_{2,2,1} = (w_{2,0,2} + Z_2 - x_{2,2,1})^+$
		$w_{2,1,3} = (w_{2,2,1} + Z_1 - x_{2,1,3})^+$
		$s_{2,0,2} = (x_{2,0,2} - Z_0)^+$
		$s_{2,2,1} = (-w_{2,0,2} - Z_2 + x_{2,2,1})^+$
		$s_{2,1,3} = (-w_{2,2,1} - Z_1 + x_{2,1,3})^+$
		$\ell_2 = (s_{2,0,2} + s_{2,2,1} + s_{2,1,3} + Z_0 + Z_1 + Z_2 - x_{2,0,2} - d)^+$
3	0–2–3–1–4 (0 and 4 dummy)	$w_{3,0,2} = (Z_0 - x_{3,0,2})^+$
		$w_{3,2,3} = (w_{3,0,2} + Z_2 - x_{3,2,3})^+$
		$w_{3,3,1} = (w_{3,2,3} + Z_3 - x_{3,3,1})^+$
		$w_{3,1,4} = (w_{3,3,1} + Z_1 - x_{3,1,4})^+$
		$s_{3,0,2} = (-Z_0 + x_{3,0,2})^+$
		$s_{3,2,3} = (-w_{3,0,2} - Z_2 + x_{3,2,3})^+$
		$s_{3,3,1} = (-w_{3,2,3} - Z_3 + x_{3,3,1})^+$
		$s_{3,1,4} = (-w_{3,3,1} - Z_1 + x_{3,1,4})^+$
		$\ell_3 = (s_{3,0,2} + s_{3,2,3} + s_{3,3,1} + s_{3,1,4} + Z_0 + Z_1 + Z_2 + Z_3 - x_{3,0,3} - d)^+$

This is a valid bound because if the allowances or the appointment times are larger than the maximum sum of the service durations over all duration scenarios, it will result in an avoidable increase in idle time and/or overtime.

The M values in the second stage, $M_2(\omega)$ and $M_3(\omega)$, are scenario-dependent. Thus, bounds on $M_2(\omega)$ and $M_3(\omega)$ can take advantage of information about service times represented by ω. We use the fact that none of the waiting time variables can take values larger than the sum of the service durations of all customers for each scenario, ω. This is true since it is not possible for a customer to wait more than the sum of completion times of all customers (which corresponds to arrival of all customers at time 0). This bound can be tightened further by making the bound customer-specific. Each customer i's waiting time must be less than or equal to the total service durations of all customers that could be sequenced prior to customer i. The new bound can be achieved by setting

$$M_2(\omega) = \sum_{i=1, i\neq i'}^{n} Z_i(\omega). \tag{26}$$

Next we consider $M_3(\omega)$ that upper bounds idle time. The allowance for each customer is bounded above by the maximum of sum of service durations of all customers over all scenarios because the idle time between two customers, say i and i', can never exceed this bound minus the duration of the customer i (given that i precedes i'). Thus, $M_3(\omega)$ can be written as follows:

$$M_3(\omega) = \max_{\omega} \sum_{k=1, k\neq i}^{n} Z_k(\omega) - Z_i(\omega). \tag{27}$$

Note that $M_3(\omega)$ is dependent on customer i, but the dependency is suppressed for simplicity in the formulation.

4.2.3. *Valid inequalities*

To improve convergence, we include additional cuts to the first-stage problem. The goal is to provide a tighter bound on θ, a surrogate variable representing the recourse function $Q(\mathbf{o}, \mathbf{x}, \mathbf{a})$ in the first-stage problem, using the mean value problem. To construct this mean-value-based subproblem, the random scenario duration Z_i is replaced with its mean value, μ_i, in a single scenario subproblem. By Jensen's inequality, the solution to this mean value subproblem provides a lower bound on the value of the recourse problem (see Birge and Louveaux (1997) for a discussion of this and other relevant properties of stochastic programs). This bound on the value of θ is as follows:

$$\theta \geq Q(\mathbf{o}, \mathbf{x}, \mathbf{a}, \bar{\boldsymbol{\xi}}).$$

New auxiliary variables, $\bar{\boldsymbol{w}}$, $\bar{\mathbf{s}}$, and $\bar{\ell}$ represent the waiting time, idle time, and overtime variables in this mean-value-based subproblem. This approach is the same as that described in Batun *et al.* (2011) and Erdogan and Denton

(2011). The set of cuts based on the mean value relaxation is as follows:

$$\theta \geq \sum_{j=1}^{n} p_j \left[\sum_{i=1}^{j} \sum_{i'=1}^{j} (c_{i'}^{w} \bar{w}_{j,i,i'}(\mu) + c^{s} \bar{s}_{j,i,i'}(\mu)) + c^{\ell} \bar{\ell}_j(\mu) \right] \tag{28}$$

$$\bar{w}_{j,i,i'}(\mu) \leq M_2(\mu) o_{j,i,i'}, \qquad \forall i, i', j, \tag{29}$$

$$\bar{s}_{j,i,i'}(\mu) \leq M_3(\mu) o_{j,i,i'}, \qquad \forall i, i', j, \tag{30}$$

$$-\sum_{i'=1}^{j} \bar{w}_{j,i',i}(\mu) + \sum_{i'=1}^{j} \bar{w}_{j,i,i'}(\mu) - \sum_{i'=1}^{j} \bar{s}_{j,i,i'}(\mu) = \mu_i - \sum_{i'=1}^{j} x_{j,i,i'}, \qquad \forall i, j, \tag{31}$$

$$\bar{\ell}_j(\mu) \geq \sum_{i=1}^{j} \sum_{i'=1}^{j} \bar{s}_{j,i,i'}(\mu) + \sum_{i=1}^{j} \mu_i + \sum_{i'=1}^{j} x_{j,0,i'} - d, \ \forall j, \tag{32}$$

$$\bar{w}_{j,i,i'}(\mu), \bar{s}_{j,i,i'}(\mu) \geq 0, \qquad \forall j, i, i', \tag{33}$$

$$\bar{\ell}_j(\mu) \geq 0, \qquad \forall j. \tag{34}$$

4.2.4. *L-shaped method*

We solved the D-ASSP using the L-shaped method, which is an iterative decomposition method that proceeds by improving an approximation (relaxation) of the first-stage problem (master problem) by adding supporting hyperplanes (optimality cuts; see Birge and Louveaux (1997) for a detailed description of the L-shaped method). After finding an integer-feasible solution to the master problem, all subproblems are solved and a new optimality cut is generated from the dual solutions of the subproblems. The optimality cut is added to improve the master problem solution, which is subsequently re-solved. This continues until the stopping criteria have been met.

Our implementation of the L-shaped algorithm is summarized in the following pseudocode.

L-Shaped Algorithm

1. $\nu = 1$ (iterations), $\omega = 1$ (scenario), initialize M_1, $M_2(\omega)$ and $M_3(\omega)$, $\forall \omega$
2. Initialize L-shape tolerance = 0.01
3. **if** option=0
4. Use formulation in (3)–(12) for the master problem
5. **else if** option=1
6. Add mean value based cuts to the master problem
7. Initialize optimality tolerance for MIP solver
8. **While** ((L-shape gap > L-shape tolerance) and (Current time < Time limit) **do**
9. $\nu \leftarrow \nu + 1$
10. Solve master problem ν and obtain current obj. value
11. Solve subproblem for each scenario ω
12. Add optimality cut to the master problem
13. L-shape gap = 100 (best obj. value – obj. value ν)/(best obj. value)
14. **end While**

The L-shaped gap is a percentage that is calculated as the ratio of the difference between the best objective function value found and the current objective function value to the best objective function value found.

In our numerical experiments, we tested several standard ways (using CPLEX 12.0) to improve the solution performance of the MIP in the first-stage problem. We utilized presolve to eliminate redundant variables and constraints. We also experimented with warm starting by using the optimal solution of the MIP in the master problem of the previous iteration as a starting solution for the MIP in the current iteration's master problem. We experimented with adding many types of MIP cuts. We added generalized upper bound cover cuts and implied bound cuts to the first stage problem (Wolsey, 1998). We also evaluated the solution performance with several search strategies including branch-and-cut and dynamic search offered by CPLEX 12.0. We also tested different variable selection strategies, such as strong branching.

5. Results

In this section, we first present results illustrating the structure of the optimal sequence and schedule for some specific examples. Then, we present the results of experiments to evaluate computational performance of our L-shaped method implementation on a series of larger test problems. All experiments were done on a Intel Core2Quad CPU, Q6600 2.39 GHz, with 4GB RAM. The methods were implemented in C++ with the CPLEX 12.0 callable library. We sampled 1000 random service duration scenarios for each model instance. All solutions reported are based on a tolerance of 1%.

5.1. *Examples of the structure of the optimal solution*

We use two examples to illustrate the structure of the optimal solution. First, we consider a model instance with five customers ($n = 5$). All customers are assumed to have identical cost coefficients for waiting times ($c_i^w = 4, \forall i$) and appointment times ($c_i^a = 2, \forall i$), probabilities of requesting appointments ($q_i = 0.5, \forall i$), and service time distributions ($Z_i \sim U(30, 40), \forall i$). The cost of overtime is $c^{\ell} = 10$, and the cost of idle time is $c^s = 5$. The optimal sequence and appointment times for this problem are presented in Table 2 (the non-dummy customers are written in bold font). Results in Table 2 indicate that FCFS is optimal for this particular example. Note that this is consistent with the sufficient condition in Proposition 4.1, which was proven

Table 2. Optimal solution of a five- customer problem instance with identical characteristics, $c_i^w = 4$, $c_i^a = 2$, $q_i = 0.5$, $\forall i$, $c^\ell = 10$, $c^s = 5$, $Z_i \sim U(30, 40)$, $\forall i$, $d = 115$. Note that a_i, $i = 1, \ldots, 5$ is an abbreviated form of the appointment decision variable used to denote the appointment time for customer i

Sequence	*Appointment times*	*Allowances*
Stage 1: $0 - \mathbf{1} - 2$	$a_1 = 0$	$x_{5,1,2} = 30.22$
Stage 2: $0 - \mathbf{1} - \mathbf{2} - 3$	$a_2 = 30.22$	$x_{5,2,3} = 33.01$
Stage 3: $0 - \mathbf{1} - \mathbf{2} - \mathbf{3} - 4$	$a_3 = 63.22$	$x_{5,3,4} = 35.05$
Stage 4: $0 - \mathbf{1} - \mathbf{2} - \mathbf{3} - \mathbf{4} - 5$	$a_4 = 98.23$	$x_{5,4,5} = 33.44$
Stage 5: $0 - \mathbf{1} - \mathbf{2} - \mathbf{3} - \mathbf{4} - \mathbf{5} - 6$	$a_5 = 131.65$	

Bold numbers indicate non-dummy patients

for the special case of $n = 2$ but also holds for this larger problem.

Next we present results for a five-customer problem instance with two different customer types: three routine and two add-ons. Routine customers are known to request appointments with certainty ($q_i = 1$ for $i = 1, 2, 3$). Add-on customers request appointments with probability 0.5. Thus, $q_i = 0.5$ for $i = 4, 5$. The waiting time cost for routine customers is $c_i^w = 4$, and for add-on customers it is $c_i^w = 10$. The waiting until appointment time cost for routine customers is $c_i^a = 0$, and for add-on customers it is $c_i^a = 10$. This problem instance is motivated by health care environments in which add-on patients have a high cost of direct waiting and waiting until appointment time. For instance, in surgery scheduling, urgent add-on patients sometimes cannot afford to wait, thus, they are scheduled early in the day. Routine patients, on the other hand, can be scheduled at any time, but have a cost associated with direct waiting. The service time distribution, cost of overtime, and cost of idle time are the same as the previous experiments ($c^\ell = 10$, $c^s = 5$, $Z_i \sim U(30, 40)$, $\forall i$). The results in Table 3 show that the optimal sequence places add-on customers at the beginning of the schedule (if they request appointments) due to their high cost of appointment times. Note that the first routine customer is also scheduled to arrive at time 0 along with the add-on customers (if they request appointments). Thus, this customer will be served first if the add-on customers do not request appointments. Note that for this problem instance, the optimal sequence is AOFS, which is consistent with the sufficient condition in Proposition 2.

Table 3. Optimal solution of a three routine + two add-on customer problem instance, $c_i^w = 4$ $\forall i = 1, 2, 3$, $c_i^w = 10$ $\forall i = 4, 5$, $c_i^a = 0$ $\forall i = 1, 2, 3$, $c_i^a = 10$ $\forall i = 4, 5$, $c^\ell = 10$, $c^s = 5$, $q_i = 1$, $\forall i = 1, 2, 3$, $q_i = 0.5$, $\forall i = 4, 5$, $Z_i \sim U(30, 40)$, $\forall i$, $d = 115$. Note that a_i, $i = 1, \ldots, 5$ is an abbreviated form of the appointment decision variable used to denote the appointment time for customer i

Sequence	*Appointment times*	*Allowances*
Stage 1: $0 - \mathbf{1} - 2$	$a_5 = 0$	$x_{5,5,4} = 0$
Stage 2: $0 - \mathbf{1} - \mathbf{2} - 3$	$a_4 = 0$	$x_{5,4,1} = 0$
Stage 3: $0 - \mathbf{1} - \mathbf{2} - \mathbf{3} - 4$	$a_1 = 0$	$x_{5,1,2} = 35.87$
Stage 4: $0 - \mathbf{4} - \mathbf{1} - \mathbf{2} - \mathbf{3} - 5$	$a_2 = 35.87$	$x_{5,2,3} = 34.38$
Stage 5: $0 - \mathbf{5} - \mathbf{4} - \mathbf{1} - \mathbf{2} - \mathbf{3} - 6$	$a_3 = 70.25$	

Bold numbers indicate non-dummy patients

5.2. *Sensitivity to service time variance*

In addition to the above experiments, we experimented with cases in which customers have different variances for their service durations. In the context of static scheduling of a fixed number of customers, previous research indicated that scheduling customers with higher variance later in the schedule minimizes the potential impact of waiting time for the later customers in the schedule (Weiss, 1990; Denton *et al.*, 2007). Intuitively, such sequences limit the amount of disruption that high-variance customers can cause for the remainder of the scheduled customers. In our next experiment, five customers having the same mean duration but different variances were considered. The cost of the appointment time for each customer was assumed to be $c_i^a = 0$ to prevent its effect on the sequencing decisions. Service durations were chosen as follows: $Z_1 \sim U(25, 35)$, $Z_2 \sim U(15, 45)$, $Z_3 \sim U(20, 40)$, $Z_4 \sim U(23, 37)$, $Z_5 \sim U(10, 50)$. Thus, service durations have a fixed mean of 30 for each customer, however, variances differ. The variances of the service durations of the customers were as follows: $\sigma_1^2 = 8.33$, $\sigma_2^2 = 75$, $\sigma_3^2 = 33.3$, $\sigma_4^2 = 16.33$, $\sigma_5^2 = 133.33$.

Table 4 provides the results for the above-defined model instance. The results indicate that for the problems with stochastic arrivals of customers with probability $q_i = 0.5, \forall i$, the customers are sequenced in FCFS order regardless of the changes in the cost coefficients. This is due to the fact that the probability of having additional customers is low when $q_i = 0.5, \forall i$, since scheduling a customer with a low probability to request an appointment before a higher-probability customer is not beneficial. For example, the probability of having one, two, three, four, or five customers in this experiment are $p_1 = 0.5$, $p_2 = 0.25$, $p_3 = 0.125$, $p_4 = 0.0625$, or $p_5 = 0.0625$, respectively. As the conditional probability gets higher ($q_i = 0.9$ and $q_i = 1, \forall i$), in the last two rows of Table 4, the uncertainty in appointment requests is reduced and the effect of having variances on the sequence becomes more prominent.

Table 4. Optimal sequencing rules for varying cost parameters and conditional probabilities q in the presence of customers with different variances. VAR denotes sequence in increasing order of variance. (VAR* indicates one exception in the increasing variance sequence)

Parameters	*Optimal sequence*	*Sequencing rule*
$c^w = c^s = c^\ell = 1$		
$q_i = 0.5 \; \forall \; i = 1, \ldots, 5$	$1-2-3-4-5$	FCFS
$c^w = 10, c^s = c^\ell = 1$		
$q_i = 0.5 \; \forall \; i = 1, \ldots, 5$	$1-2-3-4-5$	FCFS
$c^w = c^s = 1, c^\ell = 10$		
$q_i = 0.5 \; \forall \; i = 1, \ldots, 5$	$1-2-3-4-5$	FCFS
$c^w = 10, c^s = c^\ell = 1$		
$q_i = 0.9 \; \forall \; i = 1, \ldots, 5$	$1-3-4-2-5$	VAR*
$c^w = 10, c^s = c^\ell = 1$		
$q_i = 1 \; \forall \; i = 1, \ldots, 5$	$1-3-4-2-5$	VAR*

Note that $q_i = 1, \forall i$ corresponds to the static scheduling problem. In these cases, the optimal sequence approaches a schedule in which customers are in increasing order of variance. Scheduling lower-variance customers first has been observed to be near-optimal in the context of static appointment sequencing and scheduling (Denton *et al.*, 2007) because it prevents accumulation of high waiting times later in the schedule.

5.3. *Sensitivity to direct waiting and waiting until appointment time costs*

We experimented with a larger model instance with seven routine + three add-on patients with varying cost parameters for add-on patients. The conditional probability of requesting an appointment is one for routine customers and 0.5 for add-on customers. The service time distribution is assumed to be $Z_i \sim U(30, 40)$, and the length of the day is assumed to be 275. Both waiting time costs for routine customers are fixed to $c_i^a = 0$, and $c_i^w = 1$, respectively. For add-on customers waiting time to appointment and direct waiting time costs are varied between 0.01 and 1000. For each experiment 10 model instances were generated using a different random number generator seed to sample scenarios.

Results in Table 5 show that, as the direct and time to appointment waiting costs for add-on patients increase, the sequence changes from FCFS to AOFS. It is interesting that none of the instances of 4.2 could be solved to within the tolerance of 1% within the 15 000-second time limit. For these 10 instances the mean optimality gap achieved at the time of termination was 5.61% and the worst-case gap was 7.01%. The sequence varied considerably in the best solution obtained, with add-ons appearing variously at the beginning, end, and middle of the sequence.

To investigate instance 4.2 further we solved the 10 instances (same problem instance with 10 different seeds) with a computation time limit of 50 000 seconds. None of the 10 instances were solve to optimality within the increased time limit. According to the results at the time limit, none of the sequences follow FCFS or AOFS but indicate a mixed sequence of routine and urgent customers. The objective function value for each instance at the time limit is provided in Table 6. The table also provides the objective function value for the same instance if the sequence was fixed to FCFS. According to the best results achieved within the time limit using the D-ASSP model, a mixed sequence of routine and add-on customers results in a schedule that is on average 13.2% less costly than a sequence based on the FCFS sequence. Note that the average gap at the time limit for D-ASSP solution is 1.7% as opposed to the 1% for the FCFS solution.

Table 5. Optimal sequencing rules for varying direct/time to appointment cost parameters. Problems were solved to a tolerance of 1% with a maximum time of 15 000 seconds. An asterisk (*) indicates that the model instance could not be solved to the specific tolerance within the time limit

Instance no.	c^a *Routine*	c^w *Routine*	c^a *Add-on*	c^w *Add-on*	c^L	c^s	*Optimal sequence*	*CPU time* *Ave*	*CPU time* *Max*	*Number of iterations* *Ave*	*Number of iterations* *Max*
4.1	0	1	0.1	0.1	10	5	R-R-R-R-R-R-R-A-A-A	12 295.5	14 980	55.2	598
4.2	0	1	1	1	10	5	*	*	*	*	*
4.3	0	1	10	10	10	5	A-A-A-R-R-R-R-R-R-R	1174.8	1852	163.5	209
4.4	0	1	50	50	10	5	A-A-A-R-R-R-R-R-R-R	418.2	613	94.9	122
4.5	0	1	100	100	10	5	A-A-A-R-R-R-R-R-R-R	257.6	522	67.4	112
4.6	0	1	250	250	10	5	A-A-A-R-R-R-R-R-R-R	117.2	290	36	73
4.7	0	1	500	500	10	5	A-A-A-R-R-R-R-R-R-R	52.5	112	18.1	36
4.8	0	1	750	750	10	5	A-A-A-R-R-R-R-R-R-R	28.1	48	10.3	17
4.9	0	1	1000	1000	10	5	A-A-A-R-R-R-R-R-R-R	19.4	30	7.1	10

Table 6. Comparison of the D-ASSP solution with FCFS for a 10 problem instances of Experiment 4.2. (Experiment set 4.2 could not be solved with D-ASSP within 50 000-second time limit. The D-ASSP results presented in this table are for the best solutions obtained within the time limit)

Instance no.	c^a *Routine*	c^w *Routine*	c^a *Add-on*	c^w *Add-on*	c^L	c^s	*Sequence obtained for D-ASSP*	*Obj. func. value of D-ASSP solution*	*Obj. func. value of FCFS solution*
4.2.1	0	1	1	1	10	5	R-R-A-A-R-R-R-R-A-R	316.06	362.53
4.2.2	0	1	1	1	10	5	R-A-R-A-R-R-R-R-A-R	312.41	357.75
4.2.3	0	1	1	1	10	5	R-A-R-A-R-A-R-R-R-R	305.13	351.41
4.2.4	0	1	1	1	10	5	R-A-R-A-R-R-A-R-R-R	313.19	358.57
4.2.5	0	1	1	1	10	5	R-A-R-R-A-R-A-R-R-R	303.45	353.76
4.2.6	0	1	1	1	10	5	R-A-R-R-A-R-R-A-R-R	315.39	361.93
4.2.7	0	1	1	1	10	5	R-R-A-R-R-R-R-A-R-A	298.41	341.32
4.2.8	0	1	1	1	10	5	R-A-R-R-R-A-R-A-R-R	305.37	354.07
4.2.9	0	1	1	1	10	5	A-A-R-R-R-R-A-R-R-R	309.08	356.99
4.2.10	0	1	1	1	10	5	R-A-R-R-R-R-R-A-A-R	311.99	363.04

Next, we present an experiment set to evaluate the changes in the structure of the optimal sequence as both the cost ratio of direct waiting time to the cost of time until appointment ($c^w/c^a = 2, 5, 10$) and the appointment request probability ($q = 0.3, 0.5, 0.7$) change. For this experiment, it is assumed that the direct waiting cost (c^w) is the same for routine and add-on customers. Table 7 shows the sequence of the customers for these runs. For some experiments, the 10 random seeds returned slightly different optimal sequences. Thus, all sequences generated are also provided in the table. According to the results, customers are scheduled in FCFS order when q is low and c^w/c^a is high; i.e., the uncertainty in total number of customers is high and direct waiting is costlier than the waiting until time to appointment. As q increases, the uncertainty in the number of customers decreases and the sequence incorporates one add-on customer early in the schedule (following at least one routine customer). On the other hand, when c^w/c^a decreases from 10 to two, the relative importance of time to appointment increases compared with direct waiting the sequence becomes a mixture of add-on and routine customers.

5.4. *Value of the stochastic solution*

In this section we present results to evaluate the benefit of solving a stochastic programming model compared with solving a deterministic problem using the mean of the random service times for the scheduling problem. This relative benefit is called the Value of Stochastic Solution (VSS). It provides a measure of the value of the model relative to the commonly used approach in practice of scheduling according to the mean appointment time. We present the VSS for 10 patient problems with varying routine and add-on customers (with $q_i = 0.5$) and varying costs. According to the results presented in Table 8, VSS increases as the cost of waiting time for add-on customers increases. This is due to the fact that compared with the optimal sequence provided by stochastic programming solution, the mean value solution tends to place the add-on customer later in the schedule, which significantly increases the total cost due to high waiting time cost. Therefore, as the relative importance of the add-on customer increases, solving the stochastic program becomes more and morebeneficial. Furthermore,

Table 7. Results for the problem instances with 10 customers with varying cost parameters (fixed overtime, idle time, and time to appointment costs: $c^\ell = 10$, $c^s = 5$, $c_i^a = 0\ \forall i = 1\ldots7$, $c_i^a = 1\ \forall i = 7, 8, 9$), $q_i = 1\ \forall i = 1\ldots7$, $Z_i \sim U(30, 40)$

	$q_i = 0.3$	$q_i = 0.5$	$q_i = 0.7$
	R-A-R-R-A-R-R-R-R-A		
$\frac{c^w}{c^a} = 2$	R-R-R-R-R-A-R-R-A-A	R-A-R-R-R-R-R-A-R-A	R-A-R-R-R-R-R-R-A-A
	FCFS	R-A-R-R-R-R-R-R-A-A	R-R-A-R-R-R-A-R-R-A
$\frac{c^w}{c^a} = 5$	FCFS	R-A-R-R-R-R-R-R-A-A	R-A-R-R-R-R-R-R-A-A
$\frac{c^w}{c^a} = 10$	FCFS	FCFS	R-A-R-R-R-R-R-R-A-A

Table 8. VSS for 10 customer problems with varying cost coefficients

Instance	c^a *Routine*	c^w *Routine*	c^a *Add-on*	c^w *Add-on*	c^L	c^s	*Ave. obj. func. value of D-ASSP solution*	*Ave. obj. func. value of mean value solution*	*Ave. VSS (%)*
9 R + 1 A	0	1	1	1	10	5	786.93	829.19	5.03
	0	1	1	10	10	5	787.09	845.80	6.82
	0	1	1	100	10	5	787.09	1023.99	21.79
8 R + 2 A	0	1	1	1	10	5	578.52	609.59	4.64
	0	1	1	10	10	5	598.34	645.87	7.17
	0	1	1	100	10	5	631.50	899.41	28.09
7 R + 3 A	0	1	1	1	10	5	401.15	425.36	5.67
	0	1	1	10	10	5	422.79	456.67	7.34
	0	1	1	100	10	5	459.26	615.78	25.02

the minimum VSS across all instances is 4.64%, suggesting that incorporation of uncertainty in the scheduling process is generally important.

We also evaluated the benefit of solving stochastic programming problem which considers the dynamic arrival process compared with using a schedule based on expected number of customers. In this approach, the total cost of the stochastic programming problem with "R routine + A add-on customers" is compared to the expected total cost of a schedule that was initially optimized for the expected number of routine and add-on customers. In this case, even though the optimal schedule for the expected number of customers is found, additional requests of customers if they arrive have to be addressed without rescheduling. It is also possible that none of the A add-on customers request appointments, which leaves the time allocated for add-on customers idle.

Table 9 compares the $7R + 3A$ customer problem solved with D-ASSP for 10 different random seeds, with the expected total cost of scheduling the mean number of routine and and add-on customers as described above. The VSS that measures the impact of uncertainty in customer demand for this particular problem instance reaches 38% compared with the VSS of 25% presented in Table 8 that considers the duration uncertainty. This highlights the importance of capturing the stochastic nature of the appointment requests.

5.5. *Computational performance of proposed methods*

We performed *ad hoc* experiments to test several implementations of the L-shaped method to solve D-ASSP instances. We found the presolve option in CPLEX was effective in eliminating redundant variables and constraints

Table 9. VSS for the 10-customer problem considering the stochastic appointment request (c^a routine = 0, c^w routine = 1, c^a add-on = 1, c^w add-on = 100, $c^L = 10$, $c^s = 5$)

		Expected no. of customers problem					
	7R+3A SP solution	*7 requests* $p = 0.5$	*8 requests* $p = 0.25$	*9 requests* $p = 0.125$	*10 requests* $p = 0.125$	*Expected total cost of 7,8,9,10 request problems*	*VSS %*
SEED 1	456.11	551.09	118.99	921.10	1675.43	629.86	38.09
SEED 2	450.80	542.08	115.76	953.07	1716.83	633.72	40.58
SEED 3	454.04	557.79	108.39	894.35	1650.99	624.16	37.47
SEED 4	461.74	547.85	115.79	911.35	1668.95	625.41	35.45
SEED 5	472.11	569.34	129.78	920.69	1686.26	642.98	36.19
SEED 6	452.83	589.21	136.54	956.10	1711.38	662.18	46.23
SEED 7	450.83	545.67	116.49	904.11	1662.53	622.79	38.14
SEED 8	452.80	529.13	109.51	908.75	1670.41	614.33	35.67
SEED 9	455.60	537.19	108.77	906.63	1674.71	618.45	35.74
SEED 10	462.28	558.69	108.85	919.14	1671.19	630.35	36.36
Average	456.91	552.80	116.89	919.53	1678.87	630.42	37.97

and reducing computation time. We also tested warm start using the optimal solution of the MIP in the master problem from the previous iteration as a starting solution for the MIP in the next iteration. However, we observed little benefit of warm start when presolve is utilized. We further tested standard valid inequalities. From our experiments we observed that adding generalized upper bound cover cuts and implied bound cuts improved solution performance. Addition of other cuts such as mixed-integer rounding cuts, clique cuts, fractional cuts, and flow cover cuts had little or no effect on the solution time. We observed that varying search techniques between traditional branch-and-cut and CPLEX's dynamic search made no significant difference on the solution time. Also, using different variable selection strategies such as strong branching did not have a significant effect on the solution time. In our experiments CPLEX was able to solve the instances 4.3, 3.6, and 4.6. The gap at the time limit for problems 3.5 and 4.5 were 92.9%, and 87.54%, which are much worse than the reported gap for the L-shaped method provided below.

We evaluated the computational performance of our L-shaped method implementation in terms of number of iterations, CPU time, and optimality gap achieved in a fixed time limit. All of the model instances used in the experiments of this section were created by sampling using 10 different random seeds. We sampled 1000 random service duration scenarios for each model instance. The results are presented in terms of the average and maximum CPU time, and average and maximum number of iterations across the 10 replications for each model instance.

Two different service time distributions were used for the experiments: uniform and lognormal. Uniform is considered since it is a common test distribution in the appointment scheduling literature, and lognormal because it is a common distribution for modeling service durations for medical procedures (e.g., endoscopy clinics as in Berg *et al.* (2010)). The results of the experiments with uniformly distributed service durations are presented in Table 10. Instances 3.1, 3.3, and 3.5 are dynamic scheduling problems including a single customer type with $Z_i \sim U(30, 40)$, $q_i = 0.5$, $c_i^a = 2$, $c_i^w = 4$, $\forall i$. Instances 3.2, 3.4, and 3.6 include two customer types, routine and add-on. Routine customers are scheduled with certainty ($q_i = 1$) and add-on customers request appointments dynamically with $q_i = 0.5$. The cost coefficients for add-on customers are $c_i^w = 8$, and $c_i^a = 6$. Experiment sets 4.1, 4.2, 4.3, 4.4, 4.5, and 4.6 present the results of the same experiments with lognormal service durations ($Z_i \sim$ lognormal(3.2, 0.5)).

All of the experiments with five patients (instances 3.1, 3.2, 4.1, and 4.2) were solved within the time limit of 15 000 seconds to the predetermined 1% optimality gap. Among the seven-customer experiments, instance sets 3.3, 3.4, and 4.4 are all solved to optimality, but instance set 4.3 reached the time limit and terminated before finding the optimal solution. None of the instance sets for problems with 10 patients were solved to optimality within the time limit of 15 000 seconds except one instance of 3.5. The results indicate that as the number of patients increase the problems become much harder to solve.

The quality of the solution at the time of termination is also considered. Table 11 includes the optimality gap

Table 10. Computational performances of solution methods with uniformly distributed service times ($Z_i \sim U(30, 40)$) and lognormally distributed service times ($Z_i \sim$ lognormal(3.2, 0.5)). Problems were solved to a tolerance of 1% with a maximum time of 15 000 seconds. An asterisk (*) indicates cases in which no model instances were solved to the specified tolerance. Times are reported in seconds

		Uniform service distribution					*Lognormal service distribution*			
Problem size (patients)	*Customer class*	*Instance no.*	*CPU time*		*Number of iterations*		*Instance no.*	*CPU time*		*Number of iterations*	
			Average	*Max.*	*Average*	*Max.*		*Average*	*Max.*	*Average*	*Max.*
5	5 Identical Patients	3.1	69.9	91	25.4	31	4.1	127.9	167	123.6	153
	3 Routine + 2 Urgent	3.2	29.5	39	22.5	30	4.2	23.3	32	71.4	88
7	7 Identical Patients	3.3	2312.9	3156	89.2	92	4.3	*	*	279.6	289
	4 Routine+ 3 Urgent	3.4	1112.4	1460	188.2	225	4.4	4961	6510	478.1	627
10	10 Identical Patients	3.5	13 017	13 017	8	8	4.5	*	*	9.5	10
	7 Routine+ 3 Urgent	3.6	*	*	474	487	4.6	*	*	396.6	381

Table 11. L-shaped gap at the time of termination for the instances that are not solved to optimality

Problem size	*Instance no.*	*Customer type*	*L-shape gap*		
			Worst	*Best*	*Average*
7 (lognormal)	4.3	Identical Patients	18.2	13.78	15.33
10 (uniform)	3.5	10 Identical Patients	8.45	3.19	7.18
	3.6	7 Routine 3 Urgent	0.23	0.13	0.18
10 (lognormal)	4.5	10 Identical Patients	33.93	29.63	32.22
	4.6	7 Routine 3 Urgent	8.19	5.93	7.01

at the time of termination for the instances that could not be solved to optimality. The results presented in the table are the worst, best, and average gaps found within 10 replication of the instances with different random seeds. The percentage gap is calculated as the ratio of the difference between best objective function value and current objective function value to the best objective function value found.

The inclusion of mean value cuts had a significant influence on the optimality gap obtained within the time limit of 15 000 seconds. The smallest gap at the time of termination for the seven-customer model instances (instances 2.3, 2.4, 3.3, and 3.4) without mean value cuts was 107.2%; the same set of instances were solved to within 1% when the mean value cuts are added. For the model instances with 10 customers (instances 2.5, 2.6, 3.5, and 3.6), the best gap found without mean value cuts was 240.11%, whereas with mean value cuts, some instances terminated with optimality gaps as small as 2%.

It is important to note that the results that we presented in this section are for instances that are particularly difficult to solve. We generally found instances that have the same time to appointment waiting cost for patients the most challenging to solve. We also solved instances in which first-stage costs, c_i^a, and second-stage waiting time cost, c_i^w, are different for each customer. Table 12 provides two examples of problems similar to those reported in Table 10. Again, 10 randomly generated problem instances were solved. All of the instances were solved to optimality within the time limit.

Table 12. Results for the problem instances with seven and 10 patients that are solved to optimality. Parameters are $c^\ell = 10$, $c^s = 5$, $c_i^w = i$, $c_i^a = i^2$, $Z_i \sim U(30, 40)$ for uniformly distributed service durations, $Z_i \sim$ lognormal$(3.2, 0.5)$ for lognormally distributed service durations

Problem size	*Distribution type*	*L-shaped method*	
		CPU time	*Number of iterations*
7 Customers	Uniform	41.17	2
	Lognormal	436.65	21
10 Customers	Uniform	215.57	2
	Lognormal	322.16	3

6. Conclusions

We formulated the online appointment sequencing and scheduling problem using a novel formulation of the multi-stage problem as a two-stage stochastic integer program. The special case of two customers was used to develop some insight into the tradeoff between the cost of waiting until appointment time and the likelihood of additional customers arriving. We discussed a number of structural properties of the model and we presented the results of numerical experiments for two alternative implementations of the L-shaped method. We also provided insights into the types of model instances that are most computationally challenging. Our numerical experiments illustrated a number of properties of optimal online appointment schedules from which managerial insights can be drawn.

Our numerical experiments indicated that problems for which the cost of waiting until appointment time is low are the most challenging to solve. For these problems we observed that as the problem size grows (e.g., seven to 10 customer model instances), some of the instances could not be solved to a tolerance of 1% within 15 000 seconds. However, we observed that adding mean-value-based cuts to the master problems produced significant improvements in the optimality gap. For instance, the smallest gap at the time of termination for the seven-customer model instances was 107.2%, compared with 1% when the mean value cuts were added. For the model instances with 10 customers, the best gap found without mean value cuts was 240.11%, compared with gaps as small as 2% with the mean value cuts. Thus, we conclude that adding mean value cuts significantly improves computational efficiency of the L-shaped method for this problem.

Our results showed that cost parameters, c^w, c^a, c^ℓ, appointment request probabilities, q_i, and customer service time distributions, can all significantly influence the structure of the optimal online schedule. We found that, when all customers have the same costs and service time distributions, FCFS is often a good rule of thumb. In general we observe that when waiting until appointment time costs are high for add-on customers (relative to idle time and overtime costs) add-ons should all be sequenced first. In other words, the scheduler should reserve capacity at the beginning of the schedule for add-on cases. When waiting until appointment time costs are low (or zero) add-on customers

should all be scheduled last. The sufficient condition for FCFS scheduling derived for the case of two customers appears to provide a reasonable sequencing heuristic for larger problems. We observed that as q_i increases from 0.3 to 0.7, and as c^w/c^a decreases, the schedule shows a mixed sequence of routine and add-on customers, usually allowing capacity for one add-on customer at the beginning, one at the end, and one closer to the middle. We also observed that when service time distributions varied among customers and q_i increases, the customers with lower variances tend to be scheduled early in the schedule. Thus, we conclude that scheduling customers in increasing variance order is recommended when arrivals are nearly deterministic, i.e., q_i is close to one for all i. From a practical perspective the model we present is quite data-intensive. One challenge is in estimation of service time distributions, which requires access to large numbers of samples of customer services times, the availability of which varies depending on the particular application setting. Another challenge is the estimation of demand distributions for routine and add-on customers. This can be achieved using historical data on the number and types of customers scheduled. This may require a large number of observations to estimate the condition probability of customer arrivals since such demand distributions may vary by day of week, for example. Finally, estimation of cost coefficients (e.g., waiting costs for routine versus add-on customers) requires input from decision-makers.

There are some limitations of our model which present opportunities for future research. For example, in some scheduling environments no-shows can be a problem. Our model is readily adapted to this case and future extensions could explore the influence of this additional source of uncertainty. Our model also assumes a single server, but many service systems, particularly in health care environments, involve multiple servers working in parallel, and multiple stages of service. Finally, our model considers a single day of service and therefore does not explicitly consider customer preferences for different days of service. Thus, it is primarily applicable either as an exact method for environments in which customer preferences do not apply an important role (e.g., scheduling of outpatient surgery) or as a heuristic. We believe our model provides a basis for development of more complex models in the future. The methods we have developed for the single-server problem provide a foundation for the development of exact decomposition methods and/or heuristics for larger more realistic problems.

Acknowledgements

The authors are grateful for comments from the Associate Editor and three anonymous reviewers that helped improve the final version of this manuscript. This article is also based in part upon work supported by the National Science Foundation under grant number CMMI 0844511. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Batun, S., Denton, B., Schaefer, A. and Huschka, T. (2011) Operating room pooling and parallel surgery processing under uncertainty. *Informs Journal on Computing*, **23**(2), 220–237.

Berg, B., Denton, B., Nelson, H., Balasubramanian, H., Rahman, A., Bailey, A. and Lindor, K. (2010) A discrete event simulation model to evaluate operational performance of a colonoscopy suite. *Medical Decision Making*, **30**(3), 380–387.

Birge, J.R. and Louveaux, F. (1997) *Introduction to Stochastic Programming*, Springer, New York, NY.

Cayirli, T., Veral, E. and Rosen, H. (2006) Designing appointment scheduling systems for ambulatory care. *Health Care Management Science*, **9**, 47–58.

Cayirli, T., Veral, E. and Rosen, H. (2008) Assessment of patient classification in appointment system design. *Production and Operations Management*, **17**(3), 338–353.

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

Denton, B., Viapiano, J. and Vogl, A. (2007) Optimization of surgery sequencing and scheduling decisions under uncertainty. *Health Care Management Science*, **10**(1), 13–24.

Erdogan, S.A. and Denton, B.T. (2011) Dynamic appointment scheduling of a stochastic server with uncertain demand. *Informs Journal on Computing*, **25**(1), 116–132.

Gerchak, Y., Gupta, D. and Henig, M. (1996) Reservation planning for elective surgery under uncertain demand for emergency surgery. *Management Science*, **42**(3), 321–334.

Green, L.V., Savin, S. and Wang, B. (2006) Managing patient service in a diagnostic medical facility. *Operations Research*, **54**(1), 11–25.

Gul, S., Denton, B.T., Fowler, J. and Huschka, T.R. (2011) Bi-criteria scheduling of surgical services for an outpatient procedure center. *Production and Operations Management*, **20**(3), 406–417.

Hassin, R. and Mendel, S. (2008) Scheduling arrivals to queues: a single-server model with no-shows. *Management Science*, **54**(3), 565–572.

Ho, C. and Lau, H. (1992) Minimizing total cost in scheduling outpatient appointments. *Management Science*, **38**(12), 1750–1764.

Klassen, K.J. and Rohleder, T.R. (2003) Outpatient appointment scheduling with urgent clients in a dynamic, multi period environment. *International Journal of Service Industry Management*, **15**(2), 167–186.

Kolisch, R. and Sickinger, S. (2008) Providign radiology health care services to stochastic demand of different customer classes. *OR Spectrum*, **30**, 375–395.

Lin, J., Muthuraman, K. and Lawley, M. (2011) Optimal and approximate algorithms for sequential clinical scheduling with no-shows. *IIE Transactions on Healthcare Systems Engineering*, **1**, 20–36.

Liu, N., Ziya, S. and Kulkarni, V. (2010) Dynamic scheduling of outpatient appointments under patient no-shows and cancelations. *Manufacturing & Service Operations Management,* **12**(2), 347–364.

Mercer, A. (1973) Queues with scheduled arrivals: a correction, simplification and extension. *Journal of the Royal Statistical Society. Series B*, **35**(1), 104–116.

Muthuraman, K. and Lawley, M. (2008) A stochastic overbooking model for outpatient clinical scheduling with no-shows. *IIE Transactions*, **40**, 820–837.

Robinson, L.W. and Chen, R.R. (2003) Scheduling doctors' appointments: optimal and empirically-based heuristic policies. *IIE Transactions*, **35**, 295–307.

Robinson, L.W. and Chen, R.R. (2010) A comparison of traditional and open-access policies for appointment scheduling. *Manufacturing and Service Operations Management*, **12**(2), 330–346.
Sabria, F. and Daganzo, C.F. (1989) Approximate expressions for queuing systems with scheduled arrivals and established service order. *Transportation Science*, **23**(3), 159–165.
Soriano, A. (1966) Comparison of two scheduling systems. *Operations Research*, **14**(3), 388–397.
Torkki, P.M., Alho, A.I., Peltokorpi, A.V., Torkki, M.I. and Kallio, P.E. (2006) Managing urgent Surgery as a process: case study of a trauma center. *International Journal of Technology Assessment in Health Care*, **22**(2), 255–260.
Wang, P.P. (1993) Static and dynamic scheduling of customer arrivals to a single-server system. *Naval Research Logistics*, **40**, 345–360.
Weiss, E.N. (1990) Models for determining estimated start times and case orderings in hospital operating rooms. *IIE Transactions*, **22**(2), 143–150.
Wolsey, L.A. (1998) *Integer Programming*.Wiley, New York.
Zeng, B., Turkcan, A., Lin, J. and Lawley, M. (2010) Clinical scheduling models with overbooking for patients with heterogeneous no-show probabilities . *Annals of Operations Research*, **178**(1), 121–144.
Zonderland, M.E., Boucherie, R.J., Litvak, N. and Vleggeert-Lankamp, C.L. (2010) Planning and scheduling of semi-urgent Surgeries. *Health Care Management Science*, **13**(3), 256–267.

Appendix

Proof of Proposition 1. Proof is by induction. At stage $j = 1$, the sequence is required to be $0 \to 1 \to 2$. This comes from the fact that $o_{1,0,1} = 1$ by Equation (3), $o_{1,1,2} = 1$ by Equation (4), and $o_{1,i,i'} = 0$ for all other values of $i, i' = 0, 1, 2$ by Equation (7). This is obviously a valid sequence for stage 1. Suppose that the above constraints hold for any valid sequence of appointments at stage $j = k - 1$, and no constraints are violated for any such valid sequence. We will show that this implies a valid sequence for stage $j = k$, completing the proof.

Let an arbitrary valid sequence of appointments at stage $j = k - 1$ be

$$i_0 \to i_1 \to \ldots \to i_{k-1} \to i_k.$$

Since this is assumed valid, we know that $i_0 = 0$ and $i_k = k$. We also know that

$$o_{k-1,i_0,i_1}, o_{k-1,i_1,i_2}, \ldots, o_{k-1,i_{k-1},i_k} = 1,$$

and $o_{k-1,i,i'} = 0$ for all other values of $i, i' = 0, \ldots, k$.

Given that all of the constraints hold at stage $j = k$, we observe that no more than one variable in the set

$$S = \{o_{k,i_s,i_{s+1}} | s = 0, \ldots, k-2\}$$

can be zero. Otherwise, from Equation (8) with $j = k$ and the assumption that the sequence at stage $j = k - 1$ is valid, there are two distinct values t and s such that

$$o_{k,i_s,k} + o_{k,k,i_{s+1}} \geq 2(o_{k-1,i_s,i_{s+1}} - o_{k,i_s,i_{s+1}}) = 2,$$
$$o_{k,i_t,k} + o_{k,k,i_{t+1}} \geq 2(o_{k-1,i_t,i_{t+1}} - o_{k,i_t,i_{t+1}}) = 2.$$

However, this implies that the variables on the left-hand side of these inequalities are all equal to one. In particular, this means that

$$o_{k,i_s,k} + o_{k,i_t,k} = 2,$$

which violates Equation (6) when $i = j = k$. This observation leads to two cases.

In the first case, if all of the variables in S are equal to one, then Equation (7) ensures that exactly two stage $j = k$ variables outside of this set are equal to one. In other words, there are indices a, b, c, and d, where $o_{k,a,b}, o_{k,c,d} \notin S$ and $o_{k,a,b}, o_{k,c,d} = 1$. Customer k and dummy customer $k + 1$ must be involved in any valid sequence at stage $j = k$. However, $o_{k,i,i'} \in S$ implies that $i, i' \neq k, k + 1$. From Equation (4), we know that the third index of one of these two non-zero variables must be $k + 1$, and from Equation (5) with $j = k$, we know that the second index of one of these non-zero variables must be $i_k = k$. Similarly, Equation (6) with $j = k$ tells us that the third index of one of the non-zero variables must be k. The only possibility is $o_{k,i_{k-1},k}, o_{k,k,k+1} = 1$. This corresponds to the valid sequence:

$$i_0 = 0 \to i_1 \to \ldots \to i_{k-1} \to i_k = k \to k + 1.$$

This sequence corresponds to placing the kth customer at stage k after all previously scheduled customers.

In the second case, if all but one variable in S is equal to one, then from Equation (8), there is a t such that

$$o_{k,i_t,k} + o_{k,k,i_{t+1}} \geq 2(o_{k-1,i_t,i_{t+1}} - o_{k,i_t,i_{t+1}}) = 2.$$

This means that the variables on the left-hand side of the inequality are both equal to one. This fact, along with constraint (4) at $j = k$, yields the valid sequence

$$i_0 \to i_1 \to \ldots \to i_t \to i_k = k \to i_{t+1} \to \ldots \to i_{k-1} \to k + 1.$$

This sequence corresponds to placing the kth customer at stage k after the (i_t)th and before the (i_{t+1})th scheduled customers.

These two cases represent all possible valid sequences at stage $j = k$, and no constraints are violated by these sequences. ■

Proof of Proposition 2. The proof follows from optimality considering the objective function for the FCFS and AOFS cases as follows:

$$\begin{aligned} F^*_{AOFS} &= E_Z[(1-q)c^\ell(a_1^{AOFS} + Z_1) + q[c_1^w(Z_2 - a_1^{AOFS})^+ \\ &\quad + c^\ell((Z_2 - a_1^{AOFS})^+ + a_1^{AOFS} + Z_1))] \\ &\leq F_{AOFS} \qquad \text{(A1)} \\ &= c^\ell\mu + q(c_1^w + c^\ell)\mu \qquad \text{(A2)} \\ &\leq F^*_{FCFS}, \qquad \text{(A3)} \end{aligned}$$

where inequality (A1) follows from setting $a_1^{AOFS} = 0$ and the fact that a specific solution such as $a_1^{AOFS} = 0$ results in

a worse solution than the optimal solution. The inequality (A3) follows from the sufficient condition $c_2^a \geq c_1^w$ in the proposition. ■

Biographies

Ayca Erdogan is an assistant professor in the Industrial and Systems Engineering Department at San Jose State University. Previously, she was a visiting assistant professor at Daniel J. Epstein Department of Industrial and Systems Engineering at the University of Southern California, and a Postdoctoral Research Fellow at Stanford University School of Medicine. Her main research interest is decision making under uncertainty with an emphasis on problems related to health care. She is interested in operations management problems in health care delivery systems and individual- and population-level medical decision-making problems related to cancer screening. She holds a Ph.D. in Operations Research from North Carolina State University and a B.S. in Industrial Engineering from Istanbul Technical University, Turkey.

Alex Gose is an entrepreneur and business consultant. He holds a Ph.D. in Operations Research from North Carolina State University. His research interests include decision making under uncertainty, stochastic programming, and semiconductor manufacturing.

Brian Denton is an associate professor in the Department of Industrial and Operations Engineering at the University of Michigan in Ann Arbor, Michigan. Previously, he was an associate professor in the Department of Industrial and Systems Engineering at North Carolina State University, a senior associate consultant at the Mayo Clinic, and a senior engineer at IBM. He is past president of the INFORMS Health Applications Section and he serves as Secretary on the INFORMS Board of Directors. His primary research interests are in optimization under uncertainty with applications to health care delivery and medical decision making. He completed his Ph.D. in Management Science at McMaster University, his M.Sc. in Physics at York University, and his B.Sc. in Chemistry and Physics at McMaster University in Hamilton, Ontario, Canada.