Authors are encouraged to submit new papers to INFORMS journals by means of a style file template, which includes the journal title. However, use of a template does not certify that the paper has been accepted for publication in the named journal. INFORMS journal templates are for the exclusive purpose of submitting to an INFORMS journal and should not be used to distribute the papers in print or online or to submit the papers to another publication.

Data-driven appointment-scheduling under uncertainty: The case of an infusion unit in a cancer center

Avishai Mandelbaum, Petar Momčilović, Nikolaos Trichakis
Technion – Israel Institute of Technology, University of Florida, Massachusetts Institute of Technology
Sarah Kadish, Ryan Leib, Craig A. Bunnell
Dana-Farber Cancer Institute

Service systems are often stochastic and pre-planned by appointments, yet implementations of their appointment systems are prevalently deterministic. For instance, at the planning stage of healthcare services, customer punctuality and service durations are often assumed equal their means – and this gap, between planned and reality, motivated our research. Specifically, we consider appointment scheduling and sequencing under a time-varying number of servers, in a data-rich environment where service durations and punctuality are uncertain. Our data-driven approach, based on infinite-server queues, yields tractable and scalable solutions that accommodate hundreds of jobs and servers. We successfully test our approach against near-optimal algorithms, which exist for merely single-servers. This entails a novel algorithm of appointments, via data-based robust-optimization with novel uncertainty sets. To test for practical performance, we leverage a unique dataset from a cancer center, that combines real-time locations, electronic health records and appointments log. Focusing on one of the center’s infusion units (~90 daily appointments, 25+ infusion chairs), we reduce cost (waiting+overtime) by 30%.

Key words: Appointment scheduling; appointment book; data-driven decision making; scheduling under uncertainty

1. Introduction

Appointment scheduling is a ubiquitous operational process in service systems. In Healthcare Delivery Systems (HDS), for example, the majority of services operate under scheduled appointments, ranging from primary care exams to surgeries. Being essentially the process of matching supply and demand, appointment scheduling is a prerequisite for good performance: surplus of supply, i.e., too “few” appointments over a period of time, leads to low server utilization; surplus of demand, i.e., too “many” appointments, leads to customer delays. Such mismatches can have profound negative
consequences in settings where server idling and customer waiting are both costly—for example, in HDS, where servers could correspond to specialized physicians or equipment, and customers to patients. From a methodological standpoint, scheduling (a given number of) appointments, the service durations of which are stochastic, is a notoriously difficult problem. The main reason is that its analysis requires tracking transient performance measures, which renders standard queueing theory inapplicable (Kong et al. 2013). Consequently, solution approaches have so far been developed only for specialized cases; notably for settings where service durations and punctuality are deterministic (Pinedo 2009, Santibáñez et al. 2012), or for settings where service durations are uncertain, but where only a single server is in operation (Denton and Gupta 2003).

A great deal of HDS follow modes of operation, however, that lie beyond the specialized cases amenable to the existing scheduling algorithms. First, they tend to operate by sharing multiple servers. In particular, for cost-cutting and efficiency purposes, there is a recent consolidation trend in the healthcare industry, with large-scale HDS increasingly pooling or sharing their resources, instead of operating specialized and dedicated centers (Dafny et al. 2012, Bravo et al. 2017). For example, in oncology outpatient care, Massachusetts General Hospital (MGH) recently opened a new adult cancer center operating a sixty bed/chair infusion unit. Similarly, Dana-Farber Cancer Institute (DFCI) built a new cutting-edge clinical and research facility (the Yawkey Center), where infusion chairs are shared among different disease centers. Second, state-of-the-art clinical treatment processes are highly variable and subject to inherent uncertainty, owing to myriads of physiological factors and/or emergence of complex and personalized clinical pathways. For instance, infusion appointments in oncology are contingent on the patient’s evolving health status, resulting in their durations being variable and unpredictable (Barysaukas et al. 2016). (We elaborate on this in Section 3.)

In reality, there exists a significant discrepancy between the planned and the actual operations, even at HDS that employ sophisticated scheduling systems and state-of-the-art practices (e.g. DFCI). As a consequence, patients face lengthy waiting times, with their experience often being different from what was scheduled. This mismatch can be traced to the aforementioned assumptions of deterministic service times and punctuality—assumptions routinely made in appointment scheduling implementations, which are capable of handling large-scale operations that involve hundreds of servers and hundreds of customers. To this end, consider DFCI’s infusion appointment operations: Figure 1 depicts histograms of appointment punctuality and actual duration of appointments; the latter were scheduled for two hours, but their actualization illustrates significant variability.

Figure 1 offers yet another perspective: it was compiled using high-resolution data from a Real-Time Locating System (RTLS) deployed at DFCI. Our RTLS tracks both patients (≈850 per
Figure 1  Histograms of punctuality of infusion treatments (left) and of actual durations of infusion treatments scheduled for two hours (right). The graphs are based on data for all DFCI business days between November 2013 and May 2015.

In this paper, we deal with the so-called offline appointment sequencing problem, where one must decide on the appointment times for a given number of jobs—which have random durations—so as to minimize a combination of the expected wait time of jobs and the overtime of servers. Equivalently, one needs to decide on the sequence in which jobs will be scheduled and the time allowances between successive jobs. Problems in which the sequence of jobs is given and the decision only revolves around the time allowances between successive jobs are referred to as appointment scheduling problems. From a practical standpoint, the problem we study here arises, for example, at service systems that manage appointment requests in two steps: in the first step that occurs several weeks or days before service, each incoming appointment request is booked for a particular day/time-window in the future; in the second step, only a few days before service, an exact time is set for all appointments that were booked for a particular day in the first step. This two-step
Figure 2   Snapshot of our RTLS data, namely patient location at DFCI’s Yawkey Center ninth floor infusion unit at 11:17 on XX-YYY-2014. An animation of the entire day is available at youtu.be/e1qHeYg7hfW. Nodes represent specific locations, e.g., infusion chairs and hallways. Patients represented by dots traversing edges are obtaining service at the originating node and reach the destination node upon completion—for example, the patient traversing the edge emanating from Infusion Chair 25 in Room 11 is currently undergoing infusion treatment in that chair.

approach is well accepted both in practice (Mak et al. 2015) and in the literature, where the second step precisely corresponds to the (offline) appointment sequencing problem.

In the present work, we focus on the offline appointment scheduling problem for relatively large service systems. Our systems consist of a (potentially time-varying) large number of servers, who cater to a large number of customers; and the systems are stochastic in the sense that both service-durations and (customer) punctuality are subject to significant uncertainty. Our motivating example is infusion services at DFCI, where a disease center consists of 10’s of chairs (servers) that accommodate over 50 patients (customers) per day. Methodologically, as we argued already, the appointment sequencing problem is notoriously difficult (Kong et al. 2013). It becomes even more challenging in multi-server settings in which jobs are served by shared servers, as opposed to, for example, being pre-assigned to one particular server; this is because, in general, sharing of servers leads to complex queueing dynamics that cannot be expressed in closed form (e.g. à la Lindley recursion for a single server).
Table 1 Applicability of existing approaches in the literature (SO (Denton and Gupta 2003) and DR (Kong et al. 2013, Mak et al. 2015)), our DDR approach and our IS approach in terms of number of servers, and the ability to capture sequencing or punctuality.

<table>
<thead>
<tr>
<th></th>
<th>Number of shared servers</th>
<th>Allows for</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>single</td>
<td>few</td>
<td>many</td>
</tr>
<tr>
<td>Existing approaches: SO, DR</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>DDR</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>IS</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
</tbody>
</table>

Contributions

We develop a novel, data-driven approach to solve the appointment sequencing problem; we accommodate a time-varying number of shared servers, in a tractable and scalable fashion, for problem instances with hundreds of jobs and hundreds of servers. Our approach relies on an infinite-server relaxation, which accounts for uncertainty in service times and punctuality while usefully (e.g., Figures 4 and 6) modeling the complex queueing dynamics of shared-server environment; we refer to it as an Infinite-Server (IS) scheduling approach (Section 5), and to its underlying model as the IS-model or simply IS (Section 5.1). Furthermore, we utilize a CLT approximation for computational purposes. These relaxations render our approach applicable to problems involving a large enough number of servers.

We perform extensive benchmarking of our IS approach in two steps, and find that it yields near-optimal performance with significant reductions in average wait time and overtime cost. These reductions could reach as high as 30% in large-scale HDS such as DFCI, when compared against appointment scheduling implementations usually deployed in such systems. We elaborate on the two steps next.

First, in Section 6 we benchmark IS against certain near-optimal solution approaches established in the literature, specifically Stochastic Optimization (SO) (Denton and Gupta 2003) and Distributionally Robust (DR) approaches (Kong et al. 2013, Mak et al. 2015). As it happens, these approaches are only applicable to single-server environments: Table 1 provides a schematic illustration of their scope and limitations. To enable a comparison nevertheless, we develop another solution approach, dubbed Data-Driven Robust (DDR) that, loosely speaking, lies “in-between”: DDR leverages the tractability of robust optimization and enables one to explicitly model the queueing dynamics for multiple, shared servers (see Table 1), while also leveraging the wealth of operational data by relying on novel uncertainty sets. In our numerical studies, we then compare DDR with near-optimal solution approaches in a single-server setting and find that it performs equally well (if not better). This supports comparison of IS with DDR in a multi-server environment, which reveals that IS performs equally well as (if not better than) DDR.

Second, in Section 7 we benchmark IS against appointment scheduling algorithms that treat service times and punctuality as deterministic (e.g. equal to their mean values), which is the
modus operandi of virtually all implementations in practice (Berg and Denton 2012); we refer to such algorithms as means-based scheduling. We conduct our analysis using DFCI data: as already mentioned, this is a unique real-world data set, of unprecedented resolution, which draws data from both RTLS and EHR systems. To the best of our knowledge, ours is the first work to use RTLS log data (at such scale) to extract operational data. Using DFCI data allows one to evaluate the benefits that our work could yield in a realistic setting. Compared to means-based scheduling, in Section 7 we find that IS provides considerable wait time and overtime cost reductions, in the order of 30% consistently, under a wide range of experimental setups. Our approach is entirely data-driven, with data playing a central role in every step: calibration, validation, testing, and benchmarking.

Managerial Insights

Does variability “average out” in multi-server appointment systems? In particular, one might hypothesize that in such systems, where a large number of jobs are processed simultaneously by shared servers, delays from jobs that run over their expected service time would be offset by jobs that finish earlier, to the extent that stochasticity would cancel out. Such a hypothesis could be used to rationalize the use of means-based scheduling in practice. Our work, however, invalidates this hypothesis, informing HDS managers of the significant gains (18%-42% reductions in overtime plus waiting times) that appointment scheduling processes, which do account for uncertainty, can bear over means-based scheduling in multi-server environments.

RTLS implementations have been very recent and it is not yet well understood how to leverage such systems and their data (neither in an offline nor an online fashion). RTLS systems avert many observational biases that plague other data-gathering processes (see Section 3). Hence another insight that our work affords is that it showcases a way to utilize RTLS data to improve operational procedures, such as appointment scheduling.

More broadly, RTLS could better the efficient use of fixed resources (e.g., exam room space), which is essential to enable clinical growth within existing facilities. Indeed, adding or expanding a facility requires significant financial resources and time, likely years, to implement. RTLS could support methods that optimize schedules, reduce patient wait time, and distribute clinical volume to avoid unsustainable volume peaks. This would lead to a more efficient use of fixed resources and extend the time horizon before demand consistently exceeds available capacity and thus new facilities are required.

2. Brief Literature Review

The literature on appointment scheduling is vast and has dealt with multiple facets of the problem. Excellent surveys are provided by Cayirli and Veral (2003), Gupta and Denton (2008),
Cardoen et al. (2010) and Ahmadi-Javid et al. (2017). Outpatient Procedure Center scheduling is reviewed in Berg and Denton (2012), where it was remarked that, in practice, schedules are commonly based on mean procedure times. Trading of customer waiting vs. server idleness, in the context of appointment systems, is a classical problem that can be traced to Bailey (1952), Jackson (1964), White and Pike (1964). Recent works on appointment scheduling (e.g., see Zacharias and Pinedo (2014), Zacharias and Armony (to appear), Zacharias and Pinedo (2016), including Santibáñez et al. (2012), Goçgun and Puterman (2014), Dunn et al. (2017) that focus on cancer care) consider models with various features (e.g., deadlines) that capture uncertainty of future appointment requests (stochastic arrivals, urgent/elective procedures, cancellations). However, these studies do not address uncertainty in punctuality and treatment durations, i.e., deterministic service times and perfect punctuality is assumed. Below, we only discuss work that is most related to ours.

Work on appointment sequencing and scheduling under uncertainty has almost exclusively dealt with the case where a single server processes all jobs. Another prevalent assumption is that jobs arrive precisely at their scheduled times (perfect punctuality). For the single-server scheduling problem, Denton and Gupta (2003) employ a stochastic linear programming formulation and develop a variant of the standard L-shaped algorithm (e.g., Chapter 5 in Birge and Louveaux (1997)) to obtain optimal solutions. Kaandorp and Koole (2007) show that, under independent and exponentially distributed job durations, the objective is L# convex. Using this property, they develop a local search algorithm guaranteed to converge to the optimal schedule. Begen and Queyranne (2011) are the first to show that, under a general joint discrete probability distribution, the single-server scheduling problem is solvable in polynomial time, although their results are of a theoretical nature and no numerical solutions are presented.

The single-server sequencing problem appears to be intractable (Mak et al. 2015). Several complexity results have been derived (e.g., Mancilla and Storer (2012)), including recent work by Kong et al. (2016), which proves the sequencing problem to be NP-hard even under fixed time allowances. Not surprisingly hence, no exact solution methods are known. A popular heuristic is to sequence jobs in increasing order of variance (OV), which was also found to produce “good” sequences numerically. Optimality of OV has only been shown for two jobs by Weiss (1990) and Denton et al. (2007), assuming independent job durations. The recent work by Kong et al. (2016), however, argues that OV is very unlikely to be optimal for a general number of jobs.

Motivated by lack of data required to fit credible service time probability distributions in some applications, a recent stream of papers proposes the use of robust optimization (e.g., see Ben-Tal et al. (2009)) to minimize the worst-case waiting and overtime costs. Kong et al. (2013) deal with the scheduling problem and show that, under a service time uncertainty model that uses
the complete covariance matrix, it can be solved to optimality as a convex optimization problem. Using a different model of uncertainty that relies on knowledge of only marginal moments, instead of the complete covariance matrix, Mak et al. (2015) provide tractable conic programs for the robust appointment sequencing problem. We re-emphasize that all the above pertains to single-server models.

To our knowledge, when jobs are served by multiple servers, no optimal solution methods exist. In particular, by relying on Lindley’s recursion to express wait times of jobs, none of the aforementioned approaches extend to the case where jobs are served by shared servers. In addition, a direct application of stochastic programming techniques does not scale to practical problem sizes (Castaing et al. 2016).

A handful of recent papers deal with multi-server settings and study how to pre-assign each job to a particular server, which subsequently operates separately and serves only its pre-assigned jobs. In these settings, a job cannot be served by an arbitrary available server, i.e., there is no resource pooling. To deal with this job-to-server pre-assignment problem, chance-constrained optimization (Deng and Shen 2016, Deng et al. 2017) or load-balancing heuristics are employed (Mak et al. 2014). Our work differs by considering shared-server operations. We are not aware of any work that provides a tractable (approximate) solution approach for the multi-server scheduling or sequencing problems, in which a (potentially time-varying) number of servers are being shared to serve jobs with uncertain punctualities and service times.

Finally, we mention a line of data-driven research (Kim et al. 2015, 2017) where the data originates from an appointment-driven system. The approach is descriptive in that they leverage data for developing models of system characteristics (patients arrivals), while their appointment system is fixed. Our approach, on the other hand, is prescriptive in that our models seek to improve the driving appointments-system itself. To be specific, Kim et al. (2015, 2017) use data from an appointment system in a Korean outpatient endocrinology clinic, in order to model patients arrivals. Their models capture variability relative to the underlying appointments which, for example, is due to changes by patients, doctors or inherent in the system itself. To put our research in this perspective, we propose a data-driven model for the occupancy process arising from an appointment scheduling system, and our model supports improvements of the latter; to recapture the contributions, our data-source is very large, models are analytically-tractable and likely to be robust, prescriptions are well-validated and they yield significant reductions in patients delay and doctors overtime.

3. DFCI Infusion Operations and Data

In this section, we briefly describe the operational practices at DFCI, which are very similar to those employed in peer leading cancer centers, for example, the Massachusetts General Hospital (Rieb
2015). Inevitably, due to space limitations, we omit many details, since processes at large-scale cancer centers are complex. Rather, we focus on the infusion operations and the unique operational RTLS+EHR data gathering processes at DFCI.

DFCI cares for approximately 1,000 cancer outpatients on a daily basis, the majority of which receive their care at the Yawkey Center, a state-of-art 14-story building with more than 100 exam rooms and 150 infusion spaces, which are split across seven of the floors. The center is organized into multiple “disease centers” that specialize in specific cancer types. Depending on their size, several disease centers can be co-located on a single floor and share their exam rooms and infusion spaces.

The focus of our work is on DFCI’s infusion operations. Patient flow is schematically shown in Figure 3. Infusion patients visit the center according to an appointment schedule, either by having a same day exam scheduled prior to their infusion (linked appointments), or not (unlinked appointments). Currently, infusion appointment durations are set according to historical averages for patients who followed the same treatment protocol in the past. Actual appointment durations, however, differ from scheduled ones significantly—see, for example, Figure 1. This is because, for most patients, infusion protocol are adjusted based on their updated clinical conditions, determined by their blood draw results and exam. These same-day changes in treatment plans, e.g., addition, change, or removal of procedures, make actual infusion durations highly uncertain and variable. Other sources of variability include the heterogeneity of patients’ clinical condition and clinical trials, with the latter typically requiring non-standardized treatments. Consequently, the infusion process is usually the bottleneck in DFCI’s operations.

**Real-time Locating System: A Data Goldmine**

A key feature of the Yawkey Center is that it has an RTLS system installed. This system consists of a network of more than 900 sensors on the ceilings of eight clinical floors, two non-clinical floors and the parking garage. The sensors use infrared to track badges worn by patients, providers and administrators. Badge locations are recorded every three seconds. On average, approximately 850 daily patient visits are recorded by the RTLS system. RTLS compliance in the infusion units is excellent, facilitated by nurses confirming that patients are properly badged prior to treatment.
A unique feature of the RTLS data-gathering process is that it is fully automated and requires no effort by users to record activities. This automation alleviates an important deficiency of classical solely-EHR-based implementations, which typically require some manual user input. Indeed, manual data collection often suffers from observational bias because, during busy periods, manual tasks, including data logging, are frequently overlooked or not performed on-time.

As a part of our work, RTLS data from the eight clinical floors were “synchronized” with appointment book data, as well as data from the in-house pharmacy. Having access to all these logs, enables us to accurately determine not only locations of patients, but also the activities in which they are engaged. An animation of patients undergoing infusion activities on XX-YYY-2014 at DFCI is accessible at youtu.be/e1qHeYg7hfw (see also Figure 2). The resulting data set provides a representation of infusion operations at DFCI that is of unprecedented fidelity and accuracy, and this motivates our data-driven approach.

4. Appointment Scheduling and Sequencing Problems
There are \( n \) customers to be scheduled for service during a time interval \([0, \bar{T}]\). For exposition purposes, we consider time to be continuous; our solution approach can be readily employed both for continuous and discrete time. The service facility has \( c_t \) available servers at time \( t \in [0, \bar{T}] \). Regular business hours of the facility are \([0, T], T \leq \bar{T}\); any work during the time interval \((T, \bar{T}]\) counts as overtime, which incurs cost of \( \gamma \) per server per unit of time.

The planner needs to choose appointment times \( a_i \) for each customer \( i = 1, \ldots, n \). The quantities \( a_1, \ldots, a_n \) are deterministic. Appointments must be scheduled during regular business hours, that is \( a_i \leq T \). (Appointment times can also be subject to some extra constraints, e.g., \( a_i \leq T_i \), where \( T_i \) is a deadline for customer \( i \). Such simple linear constraint can readily be embedded in our algorithms.) The service time (duration) and punctuality of the \( i \)th customer are \( D_i \geq 0 \) and \( P_i \), respectively. Customer \( i \) arrives to the system at time \( t = a_i + P_i \); \( P_i > 0 \) implies that the customer is late, and conversely for \( P_i < 0 \). The random variables \( D_i \) and \( P_i \) are described by distribution functions \( F_i \) and \( G_i \), respectively; these distributions are known to the planner. Distribution \( F_i \) quantifies the cancellation probability for customer \( i \) via its mass at the origin – cancelled appointments (no-shows) can be thought of as appointments with zero service requirements. It is assumed that \( D_i \) and \( P_i \) are mutually independent (this assumption is not crucial and can be relaxed), as well as independent across customers. In general, the distributions \( F_i \) and \( G_i \) can vary with \( a_i \). With \( S_i \) being the service start time of customer \( i \), waiting time is \( W_i = S_i - a_i - P_i \geq 0 \), and the corresponding server is busy during the service interval \([S_i, S_i + D_i]\). The overtime work due to customer \( i \) is \( O_i = (S_i + D_i - T)^+ \). Customers are served by any available idle server in the order of their arrival. Customer \( i \) enters service at \( S_i \) only if this does not lead to a capacity violation.
during the service interval. That is, a customer that enters service stays in service until service is completed – a decrease in capacity never forces a customer out of service.∗

The appointment sequencing problem is to select the appointment times \(a_1, \ldots, a_n\), so as to minimize the expected cost

\[
E \left[ \sum_{i=1}^{n} (W_i + \gamma O_i) \right].
\]

(1)

Given a permutation \(\pi\) of \((1, \ldots, n)\), the appointment scheduling problem is to select appointment times \(a_1, \ldots, a_n\) in order to minimize (1) subject to \(a_{\pi_1} \leq a_{\pi_2} \leq \cdots \leq a_{\pi_n}\). That is, the scheduling problem is easier than the sequencing problem, since the relative order of appointments is fixed, and the planner needs to choose only the spacing between them.

5. Our Infinite-Server (IS) Approach

We now present our approach to solve the appointment scheduling and sequencing problems. As remarked above, even the scheduling version is a very hard problem. For “single-server” \((c_t = 1\) for all \(t)\) scheduling, although no efficient, general-purpose solution methods exist, there is a plethora of well-performing heuristics (surveyed in Section 2). Unfortunately, none of these approaches extend to multiple-server scheduling and sequencing problems where uncertainty is fully taken into account. Motivated by the challenges facing DFCI, our goal is to propose a solution approach that accommodates multiple-server environments, while utilizing the available wealth of operational data. As already noted, the multiple-server version of the problem is challenging due to the absence of closed-form, tractable expressions for its underlying queueing dynamics. Consequently, even the mapping of a given schedule to actual occupancy, or delays for that matter, remains intractable due to uncertainty of service requirements and punctuality.

We first introduce a model that approximates occupancy in a multiple-server system, given a fixed appointment schedule. Next we perform extensive validation — using real data from DFCI — that demonstrate the credibility of our model’s approximations for practical situations. Then, we embed this model in an optimization routine that allows us to produce well-performing schedules. We detail these steps in the following sections.

5.1. Infinite-Server Model

In this subsection, we introduce a model for approximating occupancy given a particular schedule. To provide some motivation first, consider Table 2: it depicts a snapshot of the appointment book for the ninth floor infusion unit at DFCI, at the beginning of AA-BBB-2014. How “good” is

∗ Such an assumption is common in analyses of time-varying queueing systems, e.g., see Liu and Whitt (2011). Moreover, it implies that service times are realized upon arrival. At DFCI infusion units, this occurs when patient’s clinical condition is assessed upon arrival, and the corresponding treatment plan is adjusted accordingly.
Table 2 A sample of the infusion-unit appointment book at DFCI; we report neither precise date nor patient id due to privacy considerations.

<table>
<thead>
<tr>
<th>date</th>
<th>time</th>
<th>patient id</th>
<th>duration (min)</th>
<th>link flag</th>
<th>floor id</th>
<th>disease center</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA-BBB-2014</td>
<td>15:00</td>
<td>xxxxx021</td>
<td>60</td>
<td>unlinked</td>
<td>9</td>
<td>breast oncology</td>
</tr>
<tr>
<td>AA-BBB-2014</td>
<td>12:30</td>
<td>xxxxx247</td>
<td>120</td>
<td>unlinked</td>
<td>9</td>
<td>breast oncology</td>
</tr>
<tr>
<td>AA-BBB-2014</td>
<td>10:30</td>
<td>xxxxx083</td>
<td>180</td>
<td>linked</td>
<td>9</td>
<td>genitourinary oncology</td>
</tr>
<tr>
<td>AA-BBB-2014</td>
<td>12:30</td>
<td>xxxxx062</td>
<td>60</td>
<td>linked</td>
<td>9</td>
<td>breast oncology</td>
</tr>
<tr>
<td>AA-BBB-2014</td>
<td>12:00</td>
<td>xxxxx740</td>
<td>120</td>
<td>linked</td>
<td>9</td>
<td>genitourinary oncology</td>
</tr>
<tr>
<td>AA-BBB-2014</td>
<td>07:00</td>
<td>xxxxx741</td>
<td>60</td>
<td>unlinked</td>
<td>9</td>
<td>breast oncology</td>
</tr>
</tbody>
</table>

Figure 4 Examples of actual (blue) and planned (black) infusion bed/chair occupancy for three days at DFCI.

For the corresponding models, the means (solid lines) and plus/minus one standard deviation from the mean (dashed lines) are shown.

this schedule? To answer this question, at the very least one must usefully predict the resulting occupancy (given distributional information on punctualities and service durations).

In Figure 4, the black “brick-shaped” line illustrates the occupancy as planned in the appointment book on AA-BB-2014, plus two other days. This deterministic description of planned occupancy is based on nominal service durations and perfect punctuality (as in Table 2). On the other hand, because of uncertainty in treatment durations and punctuality, actual occupancy can be viewed as a stochastic process, for which only a single realization is observable for a particular day (schedule). The blue lines illustrate the actual, realized occupancy we recovered using RTLS data from these days. Clearly the (deterministic) planned occupancy does not describe the corresponding actual process adequately. We hence develop a model for approximating occupancy based on an Infinite-Server (IS) model. Using this model, we obtain an occupancy approximation that is itself a stochastic process much like actual occupancy: the red lines illustrate its mean (solid) plus/minus a standard deviation (dashed). A key observation is that the IS model provides an excellent fit to observed occupancies, well within confidence intervals. We perform an extensive model validation in Section 5.2 after we formally present the IS model next.
Given a set of \( n \) customers and their appointment times \( a_1, \ldots, a_n \), our approximation is obtained by eliminating the constraint on the number of available servers. In particular, customer \( i \) arrives at time \( t = a_i + P_i \) and leaves the infinite-server system just before \( t = a_i + P_i + D_i \). For notational convenience, we define \( \tilde{F}_i(x) := 1_{\{x \geq 0\}}(1 - F_i(x)) \). Let \( Z_i := \{Z_i(t), t \in \mathbb{R}\} \), where

\[
Z_i(t) := 1_{\{a_i + P_i \leq t < a_i + P_i + D_i\}}
\]

is the indicator of customer \( i \) being present in the system at time \( t \). Then, it follows that

\[
E[Z_i(t)] = E[\tilde{F}_i(t - a_i - P_i)] =: \Omega_i(t) \quad \text{and} \quad \text{Var}(Z_i(t)) = \Omega_i(t) - \Omega_i^2(t).
\]

We use \( Z := \{Z(t), t \geq 0\} \) to denote the total number of customers in the infinite-server system:

\[
Z(t) = \sum_{i=1}^{n} Z_i(t).
\]

Assuming independence of customers, it follows that

\[
\text{Var}(Z(t)) =: \sigma^2(t) = \sum_{i=1}^{n} \Omega_i(t) (1 - \Omega_i(t)). \quad (2)
\]

Let \( X := \{X(t), t \in \mathbb{R}\} \) be the occupancy process under a fixed capacity profile \( (c_t) \) and a given schedule \( \{a_i, i = 1, \ldots, n\} \). The corresponding infinite-server process \( Z \) is expected to approximate the actual occupancy \( X \) well for situations where capacity violations do not occur frequently (light-traffic regime). In general, however, \( Z \) serves as a lower bound only: \( X(t) \geq Z(t) \), for all \( t \). Nonetheless, as we shall discover, using the infinite-server process proves useful for scheduling purposes.

Before using real DFCI data to validate our IS model approximations, we present an illustrative example with synthetic data. This example is our first demonstration that the IS model provides a credible approximation of the actual occupancy process for scheduling purposes.

**Example 1 (Homogeneous customers).** Consider a system serving \( n = 100 \) homogeneous customers. In particular, for all customers \( i = 1, \ldots, n \), the service distribution is exponential with rate \( \beta = 1 \), \( \tilde{F}_i(x) = 1_{\{x \geq 0\}}e^{-\beta x} \), and the punctuality distribution \( G_i \) is Laplace, defined by the density \( \lambda e^{-\lambda |x|}/2 \), with \( \lambda = 10 \). Capacity \( c_t \) is constant throughout.

We consider two possible service schedules, A and B illustrated in the upper right portion of Figure 5. Schedule A can be thought of as the result of a “means-based” scheduling process, akin to the deterministic planning we alluded to in the beginning of this section: service durations and punctuality are assumed to equal their expected values (unit and zero, respectively). Specifically, Schedule A entails five customer batches. Each batch has twenty customers (as many as available servers) and arrives every one unit of time (as long as service is expected to last). Schedule B has
an initial batch of 25 customers arriving at $t = 0.05$, after which the 75 customer scheduled times are uniformly spread out between $t = 0.3$ and $t = 4$.

Using our IS model, one has

$$\Omega_i(t + a_i) = \begin{cases} \lambda \frac{e^\lambda t}{2(\lambda + \beta)} e^\lambda t, & t < 0, \\ \lambda \frac{e^{-\beta t}}{2(\lambda + \beta)} e^{-\beta t} + \lambda \frac{e^{-\beta t} - e^\lambda t}{2(\lambda - \beta)}(e^\lambda t - e^{-\beta t}), & t \geq 0. \end{cases}$$

In the left column of Figure 5, $\tilde{F}_i$ and $G_i$ are plotted on the upper plot. The lower plot depicts $\Omega_i$ (blue line) and $(\Omega_i - \Omega_i^2)^{1/2}$ (dashed line), i.e., the standard deviation of $Z_i$, for $a_i = 0$. On the lower plot, the corresponding $\Omega_i$ under perfect punctuality ($P_i = 0$) and deterministic service ($ED_i = 1$) is plotted as well (black line).

In the center column, for Schedule A (shown on the top), we plot $\mathbb{E} X(\cdot)$ under three different capacity levels ($c_t$ is constant equal to 20, 22 and 24) and $\mathbb{E} Z(\cdot)$ ($c_t = \infty$); the corresponding standard deviations are shown with dashed lines. The lower plot also depicts the IS process under the assumption of unit deterministic service times and perfect punctuality (black line), whereby no capacity violation occurs for the considered capacity levels. In the right column, we depict the associated quantities for Schedule B.

The example above suggests that the higher the capacity level, the better the IS model approximates the finite-capacity system (as seen in Figure 5). Interestingly, even for $c_t = 20$, the IS process still provides a viable approximation of the occupancy process under both schedules. In particular, the IS model readily reveals that Schedule A would lead to a less efficient demand-capacity match, as compared to Schedule B. Indeed, the IS approximation accurately reflects a “shape” of the demand curve that matches capacity more closely under Schedule B rather than under Schedule A, as seen in Figure 5.

5.2. Model Validation

Before using our infinite-server model to develop a scheduling algorithm, we validate it within the DFCI operating environment. This is done with real data, and the analysis yields our first important insight: despite the complexity of DFCI operations, our basic IS model provides an excellent approximation of the intricate underlying dynamics. In particular, actual occupancy at DFCI infusion units (blue lines in Figure 4) is driven by myriad factors and is the outcome of interdependent processes that are intractable to analyze. Yet, we find that the IS model provides approximation bands that accurately characterize occupancy: as discussed above, Figure 4 depicts these bands (red) for a particular DFCI floor (infusion unit) on three specific days, with actual occupancy (blue) falling well within them. Furthermore, we find the high fidelity of the IS approximation to be consistent. Specifically, Figure 6 depicts the actual and IS-model occupancies averaged across multiple days, alongside the corresponding standardized errors; we observe an excellent fit. We detail our studies in the rest of the present subsection.
Figure 6 The average number of occupied infusion beds/chairs at DFCI for the training (left) and the testing (center) sets based on RTLS data ("actual", blue line) and appointment book data ("planned", black line); the average of model means ("model", red line) is shown as well. A histogram of normalized errors is shown along with the corresponding normal density (right).

Model focus and granularity We apply our model to a single floor (infusion unit) at DFCI separately, rather than to all floors simultaneously, since (i) operation of different floors is semi-autonomous, (ii) a disease center is located on a single floor, and (iii) statistical properties of disease centers (patients undergoing treatment, operations) can differ significantly. Moreover, we consider different days of the week separately, as provider schedules differ from one day of the week to another, resulting in different daily patient loads and populations (e.g., Fridays are not as “busy” as other days of the week). Given a floor and a day of the week, we estimate bed/chair time distributions and punctuality distributions. In particular, we estimate “service” distributions conditional on (i) the disease center, (ii) the scheduled infusion duration, (iii) the type of the
appointment (linked/unlinked), and (iv) the time of the day (morning vs. afternoon); these parameters are extracted from the appointment books of DFCI. As far as punctualities are concerned, we estimate conditional punctuality distributions based on (i) the disease center, (ii) the time of the day (scheduled infusion start times), since average delays change during a day, and (iii) the type of the appointments (linked/unlinked). Hence, the function $\Omega_i$ for the $i$th patient depends on (i) the scheduled treatment (infusion) start time (via the $a_i$ and the punctuality/service distribution that corresponds to $a_i$), (ii) the scheduled treatment duration (via the corresponding distribution that corresponds to the treatment duration), and (iii) the type of the appointment. Given these estimated distribution functions, an infinite-server model is constructed for a given day (schedule). The model mean and variance can be computed from $\Omega_i$’s based on the previous subsection.

“Modeling” Wednesdays on the ninth floor We illustrate the described procedure by an analysis of Wednesdays on the ninth floor. A time period containing 37 Wednesdays in 2014 was considered (from February 19 to October 29). During this time period, the infusion unit was shared between two disease centers (breast oncology and genitourinary oncology). Three days were excluded from our analysis (March 12, June 11 and August 20) due to RTLS system interruptions on those particular days. The remaining 34 days were divided into the training (20 Wednesdays, from February 19 to July 16) and testing (14 Wednesdays, from July 23 to October 29) sets. All schedules for the considered days are different. The number of patients varies from 62 to 101 a day (the average is 83.3, while the standard deviation is 7.7). The number of scheduled (actual) infusion hours ranges from 122.0 (153.4) to 188.5 (223.8) a day (the average is 153.6 (190.3), while the standard deviation is 17.2 (17.8)).

The training set was used to estimate conditional punctuality and service distributions. All analysis was conducted on a five-minute scale, i.e., the unit of time is five minutes. Punctuality distributions were estimated based on values of appointment times. In particular, based on a statistical analysis, eight time intervals were considered: $[00:00-09:00)$, $[09:00-10:00)$, $[10:00-11:00)$, $[11:00-12:00)$, $[12:00-14:00)$, $[14:00-15:00)$, $[15:00-16:00)$, and $[16:00-24:00)$. Similarly, service distributions were estimated based on scheduled appointment durations being in one of eight intervals (in minutes): $[5,25)$, $[25,45)$, $[45,75)$, $[75,105)$, $[150,210)$, $[210,270)$, $[270,330)$, $[330,420)$, $[420,540)$, $[540,660)$, and $[660,1000)$. The estimated distributions were used to construct infinite-server models for all considered days.

Validation results In Figure 6, we plot the number of occupied infusion beds/chairs averaged across the training (left) and the testing (center) sets. In both graphs, the “actual” and “planned” curves are plotted based on RTLS data and data derived from the appointment book, respectively. We note that an infinite-server model for a given day (conditional on its schedule, i.e., appointment
book) is a random object. Thus, in Figure 6, the red lines corresponding to the model represent the average (across days) of model means. In Figure 4, we illustrate the relationship between the actual/planned number of occupied infusion beds/chairs and the model for three particular days. For the model, we show the means (the solid red lines) as well as the levels corresponding to plus/minus one standard deviation from the means (dashed red lines). The actual number of occupied beds/chairs can be viewed as a single realization of a random process conditional on the schedule. Given a day in a testing set and a time of the day, we define a normalized error as a difference between the actual number of occupied beds/chairs and the model mean, scaled by the standard deviation of the model. A histogram of normalized errors for the testing set and specific times of the day (10:00, 11:00, 12:00, 13:00, 14:00, 15:00, and 16:00) is shown in Figure 6 (right); a corresponding standard normal density is plotted as well. Note that the distribution of normalized errors would be standard normal if the occupancy process were indeed an infinite-server process with a large traffic intensity. In this case, the sample average and standard deviation are equal to 0.06 and 0.98, respectively. These numbers demonstrate the excellent fit the IS model provides, as we discussed at the beginning of this section.

5.3. Infinite-Server Sequencing Heuristics

We now use the infinite-server model to propose a heuristic for the appointment scheduling and sequencing problems. For a function \( r: \mathbb{R} \to [0, \infty) \), we define a cost of a sample path of \( Z \) as

\[
Q(Z) := \int_{-\infty}^{\infty} r(Z(t) - c_t) \, dt + \tilde{\gamma} \int_{T}^{\infty} Z(t) \, dt,
\]

for some \( \tilde{\gamma} > 0 \); the two terms approximate costs of waiting time and overtime. Define \( \tilde{Z} = \{ \tilde{Z}(t), t \geq 0 \} \) as a CLT-based approximation of \( Z \):

\[
\tilde{Z}(t) := \mathbb{E}Z(t) + \xi(t) \sigma(t),
\]

where \( \xi(t) \) is a standard normal random variable and \( \sigma(t) \) is given by (2). Consequently, \( Q(\tilde{Z}) \) is an approximation of \( Q(Z) \), which gives rise to the following problem:

\[
\min_{\{a_i \in [0,T]\}} \mathbb{E}Q(\tilde{Z}). \tag{4}
\]

For example, if \( r(x) = x^+ \), one has

\[
\mathbb{E}Q(\tilde{Z}) = \int_{-\infty}^{\infty} \sigma(t) (\varphi(\psi(t)) + \psi(t) \Phi(\psi(t))) \, dt + \tilde{\gamma} \int_{T}^{\infty} \mathbb{E}\tilde{Z}(t) \, dt, \tag{5}
\]

where

\[
\psi(t) := \frac{\mathbb{E}Z(t) - c_t}{\sigma(t)};
\]
here, we use the common notation for the standard normal density and distribution function (\( \varphi \) and \( \Phi \), respectively). In the case when \( G_i \) does not vary with \( a_i \), (5) implies

\[
\frac{\partial}{\partial a_i} \mathbb{E}(\tilde{Z}) = \int_{-\infty}^{\infty} \left[ \frac{\Omega_i(t) - 1/2}{\sigma(t)} \varphi(\psi(t)) - \Phi(\psi(t)) \right] d\Omega_i(t) + \tilde{\gamma} \Omega_i(T).
\]

The advantage of using approximation (3) is that \( \tilde{Z}(t) \) is fully characterized by its mean and standard deviation only. It is expected that the approximation is relevant whenever the central limit theorem is applicable. In general, solving (4) to optimality is hard, since the objective function is not necessarily convex. However, one can efficiently obtain local optimal solutions for large problem sizes. Candidate starting points can be obtained using the OV heuristic, or efficient means-based scheduling algorithms that ignore uncertainty, or random sampling. By approximately solving the problem for various values of \( \tilde{\gamma} \), one can obtain schedules close to the efficient frontier of wait time versus overtime costs, and eventually select the one that trades off \( \gamma \) units of wait time with one unit of overtime.

The non-convexity of the IS heuristic objective (4) is a limitation of the IS approach, since one needs to resort to local-search algorithms. One can overcome this limitation by formulating a well-behaved objective that leads to a simpler (or standard) optimization framework. However, such an alternative comes at the expense of neglecting key (stochastic) features of the underlying system. Our design choice is to incorporate essential stochastic aspects of the system (along with the corresponding distributional data) in the model by sacrificing convexity.

6. Performance Evaluation: Comparison with Robust Appointment Scheduling

We benchmark our IS sequencing heuristic, which accommodates multiple servers, against the state-of-the-art solution approaches in the literature that deal with single-server instances. To this end, we develop a novel Data-Driven Robust (DDR) solution approach that “bridges” the two: it deals with any capacity vector \( c_i \), albeit for the appointment scheduling problem where customers’ order is fixed (see Table 1). Existing approaches in the literature assume perfect punctuality, that is, \( P_i = 0 \), for all \( i \); we henceforth also make this assumption in the present section. (The perfect punctuality case can be used to consider settings where punctualities are non-zero but constant.)

Numerical studies on the scheduling problem reveal that the performance of our IS approach is at least as good as (if not superior to) the DDR approach for the multiple-server problem, while the performance of the latter is at least as good as (if not superior to) the state-of-the-art heuristics in the literature for the single-server problem.
6.1. Data-Driven Robust (DDR) Scheduling

We develop a robust optimization approach to solve the appointment scheduling problem under perfect punctuality. It is applicable to any capacity specification $c_t$ and considers discrete time; for the single-server case, it can also be readily adapted to continuous time. Our approach differs from classical approaches in robust optimization by being grounded in data—a point elaborated upon in the uncertainty set description below.

At a high-level, the robust optimization approach employs two stages. In the first one, we select a schedule $a_1 \leq a_2 \leq \cdots \leq a_n$; it minimizes the cost incurred in the second stage, in which “nature” draws the service durations from an uncertainty set so as to maximize the incurred cost.

**Uncertainty set of service durations** Uncertainty sets in classical robust optimization rely on limited information about the uncertain variables, which in our case are the service times. In particular, they are usually calibrated using only the support of the uncertain variables, for example, or their means, or their higher-order moments, etc. Consequently, they tend to perform well in settings where there is a scarcity of data.

In our setting, on the other hand, a wealth of data is available. Thus, instead of extracting only partial support and/or moment related information, we propose a novel construction of uncertainty sets that leverage all the available data explicitly, specifically by “sampling” from the available empirical distributions. (Hence, we refer to our approach as data-driven.) In particular, consider $n$ independent uniformly distributed random variables, $U_i$, $i = 1, \ldots, n$. Samples from these random variables can be readily used to obtain samples of the service time durations of the $n$ customer appointments:

$$D_i = F_i^{(r)}(U_i), \quad i = 1, \ldots, n;$$

here, $F_i^{(r)}$ is the (left continuous) inverse of $F_i$: $F_i^{(r)}(x) = \inf\{s : F_i(s) \geq x\}$. Since the $U_i$’s are independent and identically distributed, we propose to use CLT-style constraints to form the uncertainty sets that they lie in. Such constraints were introduced and found to perform well in recent papers in the robust optimization literature, see, e.g., Bandi and Bertsimas (2012).

To formalize the construction of our proposed uncertainty set, we need to “discretize” our random variables. In particular, consider $J$ point values of the $i$th service duration’s inverse distribution function:

$$\delta_{ij} := F_i^{(r)}\left(\frac{j}{J+1}\right), \quad i = 1, \ldots, n, \quad j = 1, \ldots, J.$$

Let the associated uniform random variable $U_i$ take the corresponding discrete values $j/(J+1)$, $j = 1, \ldots, J$. We can then express the service durations as

$$d_i = \sum_{j=1}^{J} \delta_{ij}u_{ij}, \quad i = 1, \ldots, n,$$  

(6)
where \( u_{ij} \) is the indicator variable

\[
u_{ij} = \begin{cases} 
1 & \text{if } U_i = j+1, \\
0 & \text{otherwise.}
\end{cases}
\]

Viewed from a different angle, the variables \( u \) can be interpreted as assignment variables. That is, the service durations have a range of possible values \( \delta \) that they can take. The variables \( u \) assign to the service durations one of their possible values.

The indicator variables lie in the uncertainty set

\[
\mathcal{U} := \left\{ u \in \{0,1\}^{n \times J} : \sum_{j=1}^{J} u_{ij} = 1, \forall i = 1, \ldots, n, \sum_{i=1}^{n} \sum_{j=1}^{J} \frac{2}{\sqrt{12}} u_{ij} - \frac{n}{2} \leq \Gamma \right\},
\]

where \( \Gamma \) is a conservatism parameter. The first constraint in the definition of \( \mathcal{U} \) is an assignment constraint: precisely one of the \( u_{ij} \)’s is equal to one for each \( i \). The second constraint ensures that the normalized sum of the i.i.d. variables \( U_i \) is not larger than \( \Gamma \) or smaller than \(-\Gamma\). For example, if \( \Gamma = 1 \), the sum is within one standard deviation from its mean. Higher values of \( \Gamma \) mean that the sum of service durations can deviate even more from its mean, allowing nature a larger set of possible durations to pick from so as to maximize costs.

**Nature’s problem: Worst-case durations** Given an appointment schedule \( a \) and a conservatism parameter \( \Gamma \), nature’s problem is to pick the service durations that maximize the incurred cost. To derive a strong formulation for nature’s problem, we map each schedule time variable \( a_i \) to \( \bar{T} \) binary variables \( \alpha_{it}, t = 1, \ldots, \bar{T}, \) that indicate whether service \( i \) has been scheduled by time \( t \) or not. Viewed differently, variable \( \alpha_{it} \) indicates whether customer \( i \) is available for service at time \( t \). Clearly, there is a one-to-one mapping between the \( a \) and \( \alpha \) variables:

\[
\alpha_{it} = 1_{\{a_i \leq t\}}, \quad a_i = \sum_{\tau=1}^{\bar{T}} \tau \cdot (\alpha_{i\tau} - \alpha_{i,\tau-1}), \quad i = 1, \ldots, n, \quad t = 1, \ldots, \bar{T},
\]

(7)

where \( \alpha_{i0} := 0 \) for all \( i \).

The main decision variables for nature are the service duration assignment variables \( u \); the actual durations can be retrieved using (6). Let \( b_{it} \) and \( e_{it} \) be auxiliary decision variables that indicate whether service for the \( i \)th customer has begun and ended, respectively, by time \( t \), in a similar fashion as the availability variables \( \alpha \). Nature’s problem can then be formulated as an integer optimization problem:

\[
W(\alpha) := \text{maximize} \quad \sum_{i=1}^{n} \sum_{t=1}^{\bar{T}} (\alpha_{it} - b_{it}) + \gamma \sum_{i=1}^{n} \sum_{t=T+1}^{\bar{T}} (b_{it} - e_{it})
\]

(8a)

subject to \( b \leq \alpha \) \hspace{1cm} (8b)

\( e \leq b \) \hspace{1cm} (8c)
Mandelbaum et al.: Data-driven appointment-scheduling under uncertainty
Management Science 00(0), pp. 000–000, © 0000 INFORMS

\begin{align}
    b_{it} & \leq b_{i,t+1}, \quad i = 1, \ldots, n, \quad t = 1, \ldots, \bar{T} - 1 \quad (8d) \\
    e_{it} & \leq e_{i,t+1}, \quad i = 1, \ldots, n, \quad t = 1, \ldots, \bar{T} - 1 \quad (8e) \\
    b_{it} & \leq b_{i-1,t}, \quad i = 2, \ldots, n, \quad t = 1, \ldots, \bar{T} \quad (8f) \\
    \sum_{i=1}^{n} (b_{it} - e_{it}) & \leq c_t, \quad t = 1, \ldots, \bar{T} \quad (8g) \\
    \sum_{i=1}^{n} (b_{it} - e_{it}) & \geq f_t c_t, \quad t = 1, \ldots, \bar{T} \quad (8h) \\
    \sum_{i=1}^{n} (\alpha_{it} - b_{it}) & \leq n f_t, \quad t = 1, \ldots, \bar{T} \quad (8i) \\
    \sum_{i=1}^{\bar{T}} (b_{it} - e_{it}) & = \sum_{j=1}^{J} \delta_{ij} u_{ij}, \quad i = 1, \ldots, n \quad (8j) \\
    u & \in \mathcal{U} \quad (8k) \\
    b, e, f & \text{ binary,} \quad (8l)
\end{align}

with aforementioned variables \( u \in \{0, 1\}^{n \times J} \), \( b \in \{0, 1\}^{n \times \bar{T}} \), \( e \in \{0, 1\}^{n \times \bar{T}} \), and variables \( f \in \{0, 1\}^{\bar{T}} \) that indicate whether all \( c_t \) servers are busy at time \( t \). Constraints (8b-8c) ensure that service begins after customers are available and ends after it has started, respectively. Constraints (8d-8e) enforce time connectivity: if service for customer \( i \) has begun (ended) by time \( t \), then \( b_{it} \) (\( e_{it} \)) has a value of one for all later time periods. Constraint (8f) enforces first-come first-served policy. Constraint (8g) is a capacity constraint. In conjunction with it, constraint (8h) ensures that \( f_t = 1 \), if \( c_t \) servers are busy at time \( t \). Constraint (8i) allows waiting only if \( f_t = 1 \). Finally, constraint (8j) sets the duration of each service to be as assigned by variables \( u \).

Note that \( W(\alpha) \) is the optimal value of (8). That is, if the appointment schedule is \( \alpha \), \( W(\alpha) \) is the worst-case waiting plus overtime cost incurred, when the service durations are allowed to take values in the uncertainty set described above. We are now ready to formulate the robust appointment scheduling problem.

**Data-driven robust appointment scheduling** We equivalently use the availability variables \( \alpha \) as the decision variables; schedule times can then be retrieved using (7). The DDR approach picks a schedule that minimizes the worst-case cost incurred:

\begin{align}
    \text{minimize} \quad & W(\alpha) \quad (9a) \\
    \text{subject to} \quad & \alpha_{it} \leq \alpha_{i,t+1}, \quad i = 1, \ldots, n, \quad t = 1, \ldots, \bar{T} - 1 \quad (9b) \\
    & \alpha_{it} \leq \alpha_{i-1,t}, \quad i = 2, \ldots, n, \quad t = 1, \ldots, \bar{T} \quad (9c) \\
    & \alpha_{it} = 1, \quad i = 1, \ldots, n, \quad t = T, \ldots, \bar{T} \quad (9d) \\
    & \alpha \text{ binary.} \quad (9e)
\end{align}
Table 3  Summary of mean percentage differences (in %) between DDR and SO solutions as well as between DDR and DR solutions. The summary is across service distributions (lognormal/gamma) and utilization levels (0.85/1.00/1.15).

<table>
<thead>
<tr>
<th></th>
<th>SO</th>
<th></th>
<th></th>
<th>DR</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Min. gap</td>
<td>+0.8</td>
<td>Avg. gap</td>
<td>+2.0</td>
<td>Max. gap</td>
<td>+3.7</td>
<td></td>
</tr>
<tr>
<td>Avg. gap</td>
<td>-0.1</td>
<td>Avg. gap</td>
<td>-2.5</td>
<td>Max. gap</td>
<td>-4.6</td>
<td></td>
</tr>
</tbody>
</table>

Constraint (9b) enforces time connectivity, (9c) enforces the scheduling sequence, and (9d) ensures that all appointments are scheduled during regular business hours. Problem (9) can be solved to optimality using Benders decomposition; we refer the reader to Appendix A for more details.

6.2. Numerical comparison of existing approaches with DDR

We conduct numerical experiments to assess the performance of our data-driven robust approach. We compare it against existing appointment scheduling algorithms that deal with single-server instances under uncertainty. We use an experimental setup akin to Kong et al. (2013). In particular, we consider different cases, where in each case we vary the number of customers, service distributions, utilization and overtime cost rate. For concrete parameter choices, the reader is referred to Appendix A.2.

We compare our DDR approach with Stochastic Optimization (SO) and a Distributionally-Robust (DR) approach. SO minimizes expected cost having access to full distributional information. It is used as a benchmark, since it is known to produce near optimal solutions for the small problem sizes we consider here (Denton and Gupta 2003). Table 3 provides an overview of our results. Not surprisingly, SO produces lower costs than our approach; however, our approach yielded costs that were only 2% higher on average across all cases, and at most 3.7%. We provide a more detailed comparison in Appendix A.2.

The DR approach we use follows Kong et al. (2013) and Mak et al. (2015), by minimizing worst-case cost, over all distributions that have the same mean and variance as the true service distributions. That is, the DR approach leverages only partial information about service times. It is used as a benchmark, since it has been shown to perform very well for realistic problem sizes that include more than 40 customers (still only with a single server). In comparison, our DDR approach yielded costs that were 2.5% lower on average across all cases, occasionally 4.6% lower. Note that our approach yields superior performance by utilizing more distributional information (as we remarked above).† For additional comparison details, see Appendix A.2.

† Note, however, that this advantage comes at the cost of having to solve integer optimization problems (as opposed to second-order cone problems in the DR approach). On the positive side, our approach can also deal with multiple-server problems.
6.3. Numerical Comparison of IS with DDR

Having established that the DDR approach exhibits near-optimal performance for single-server scheduling problems, we now benchmark our IS approach against it for multiple-server scheduling problems. In particular, we consider the problem of scheduling \( n = 40 \) customers to 15 servers \((c_t = 15, t = 1, \ldots, T, \text{ where } T = 90)\). The per-unit overtime cost is \( \gamma = 2 \), and overtime contributes to the overall cost after \( T = 50 \). All customers have the same distribution type for their service requirements: lognormal or gamma (as in Kong et al. (2013)). However, the mean and the coefficient of variation for each customer are sampled from a uniform distribution. In particular, the mean service requirement \( d_i \) of customer \( i \) is uniform on \([2\bar{d}/3, 4\bar{d}/3]\), where \( \bar{d} \) is determined by a desired mean utilization \( nd/(ctT) \); the coefficients of variation are uniform on \([2/3, 4/3]\). For each distribution type (lognormal or gamma) and three utilization levels (0.85, 1.00 and 1.15; this corresponds to \( \bar{d} \approx 0.32T, 0.38T \) and \( 0.43T \)), we generate (sample means and coefficients of variation) 20 cases: these are solved by the two proposed algorithms (parameters \( \Gamma \) and \( \tilde{\gamma} \) were calibrated for each group of 20 cases). The two solutions (for each case) are then fed into a simulator to obtain estimates of the mean of the total cost (based on \( 10^6 \) samples). For the IS approach we use \( r(x) = x^+ \), a choice we retain for all subsequent experiments. Note that both the IS and the DDR approaches use information on distributions of customer requirements. In order to gain additional insights, we also solve all 120 cases by feeding the algorithms with a “wrong” type of a distribution with correct means and variances. For example, if the lognormal distribution was used to generate service requirements, in the algorithms it is assumed that the distribution is gamma (with the correct first two moments for each customer).

Our numerical results are summarized in Figure 7 and Table 4. Note that the IS approach delivers a uniformly superior performance to DDR across all our experimental setups. When the underlying distributions are accurately specified, the cost reduction that IS achieves over DDR is 3.6%, on average. Interestingly, IS holds on to its advantage, even under distributional mis-specifications, lowering cost by 2.8%, on average. In summary, our experiments suggest that the IS approach is uniformly slightly superior to the DDR approach, with the latter found to be near-optimal for the single-server problem setting. Recall moreover that, unlike DDR, the IS approach is capable of handling sequencing, uncertainty in punctualities, as well as handling practice-relevant problem sizes.

7. Performance Evaluation: Scheduling at DFCI

In this section, we compare our IS appointment sequencing approach with means-based scheduling (detailed below) in the context of DFCI infusion operations. The latter is commonly used in outpatient settings (Berg and Denton 2012). In particular, we use our high-resolution RTLS data
Figure 7  Mean costs of DDR (blue) and IS (red) appointment scheduling solutions for different distribution (actual/assumed) pairs and mean utilization levels (0.85, 1.00 and 1.15). The number of servers and customers are 15 and 40, respectively.

Table 4  Mean percentage differences (in %) between IS and DDR solutions for means and percentiles. The pair of distributions refers to actual/assumed distributions, where the L and G stand for lognormal and gamma, respectively.

<table>
<thead>
<tr>
<th>Distributions</th>
<th>Mean utilization</th>
<th>Performance measure</th>
<th>L/L</th>
<th>L/G</th>
<th>G/G</th>
<th>G/L</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.85</td>
<td>Mean</td>
<td>-5.6</td>
<td>-6.8</td>
<td>-3.0</td>
<td>+0.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>75% percentile</td>
<td>-4.5</td>
<td>-5.3</td>
<td>-1.7</td>
<td>+1.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>85% percentile</td>
<td>-1.5</td>
<td>-2.5</td>
<td>+2.1</td>
<td>+4.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>95% percentile</td>
<td>+2.8</td>
<td>+1.9</td>
<td>+7.4</td>
<td>+10.0</td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>Mean</td>
<td>-4.3</td>
<td>-4.1</td>
<td>-4.1</td>
<td>-3.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>75% percentile</td>
<td>-3.4</td>
<td>-3.3</td>
<td>-3.4</td>
<td>-2.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>85% percentile</td>
<td>-2.9</td>
<td>-2.9</td>
<td>-2.7</td>
<td>-1.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>95% percentile</td>
<td>-2.5</td>
<td>-2.6</td>
<td>-1.8</td>
<td>-0.9</td>
<td></td>
</tr>
<tr>
<td>1.15</td>
<td>Mean</td>
<td>-2.1</td>
<td>-2.1</td>
<td>-2.2</td>
<td>-1.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>75% percentile</td>
<td>-2.8</td>
<td>-2.8</td>
<td>-2.8</td>
<td>-2.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>85% percentile</td>
<td>-2.9</td>
<td>-2.9</td>
<td>-3.0</td>
<td>-2.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>95% percentile</td>
<td>-2.8</td>
<td>-2.8</td>
<td>-2.7</td>
<td>-2.2</td>
<td></td>
</tr>
</tbody>
</table>

to generate multiple experimental scenarios, whereby daily infusion appointment scheduling tasks, as faced by DFCI, are carried out by the two approaches. Typical scenarios for a DFCI infusion unit involve approximately \( n = 90 \) daily appointments that are served by 25+ infusion beds/chairs during business hours.

The comparison of the two approaches yields the second important insight of this work: our approach, which accounts for variability in a real-world multiple-server system, significantly outperforms the means-based approach. In particular, the analysis here suggests that deployment of
our IS appointment sequencing approach at DFCI could decrease total cost (costs of waiting plus overtime) by roughly 30%. Next, we provide a detailed description of our analysis.

### 7.1. Experimental Setup

We setup our experiments as follows. Consider a specific historical day for which one can observe the original appointments that were scheduled, including their “type:” the latter is defined by day/time of the week, appointment’s duration, associated disease center and whether a preceding linked exam was scheduled. Not having access to the true statistical properties of the appointments’ durations/punctuality for that day (we only observe a single sample path), we use a bootstrapping procedure to “reconstruct” these properties from the data by using observed punctuality/service samples of same-type appointments on other days. In particular, we obtain empirical values for punctuality (duration) by sampling from a set of punctuality (duration) of appointments that occurred on the same day of the week and interval of the day (morning vs. afternoon), were scheduled for the same nominal duration, were from the same disease center, and had the same exam linkage.

The availability of infusion chairs is specified so that it mimics DFCI nurse-staffing policies. In particular, there is no available capacity before 7:00. Capacity then increases linearly between 7:00 and 10:00 (in intervals of 30 minutes), remains at a constant number (maximum) between 10:00 and 17:00, before decreasing linearly between 17:00 and 19:00 (in intervals of 30 minutes) to reach five available chairs at 19:00 until the end of the day. Overtime starts at 19:00. Thus, the capacity profile is parametrized by a single value: maximum capacity (between 10:00 and 17:00). This value is set based on a desired utilization computed over business hours (7:00-19:00). To this end, we compute the average day-demand in infusion hours (based on the original schedule), and then reduce the maximum capacity until a desired value of utilization is reached. As in Section 5.2, the unit of time is chosen to be five minutes. Appointments can be made between 7:00 and 17:00.

### 7.2. Experiments and Results

Within this experimental setup, we produce schedules using our IS sequencing approach and a standard means-based sequencing approach. The latter is an approach where all random quantities are treated as deterministic and equal to their expected values, and state-of-the-art mixed-integer programming techniques are employed to optimize the schedule. In Appendix B, we elaborate on the exact means-based sequencing algorithm we used. Computed schedules were simulated and averages are obtained based on $10^6$ samples. Numerical results are presented in Figure 8. In particular, on the left plot, we provide an overtime-waiting tradeoff for XX-YYY-2014 (which was a Wednesday), on the ninth floor of DFCI. On that day, 87 patients had infusion appointments and received treatment. The overtime-waiting tradeoff for the IS approach (blue line) is obtained
Figure 8  Overtime-waiting tradeoffs for two days on two different DFCI floors. The higher curves correspond to the higher utilization level.

![Graph showing overtime-waiting tradeoffs]

Table 5  Percentage decrease (in %) for total mean cost, between offline IS and means-based solutions at DFCI.

<table>
<thead>
<tr>
<th>Utilization</th>
<th>Per-unit overtime cost $\gamma$</th>
<th>1/3</th>
<th>1/2</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Floor 9</td>
<td>$&gt; 0.85$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$&gt; 0.95$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Floor 8</td>
<td>$&gt; 0.85$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$&gt; 0.95$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

by varying the parameter $\tilde{\gamma}$ (see Section 5.3); and for the means-based approach (black line), it is obtained by scaling capacity (see Appendix B for details).

In addition, we performed the same analysis for ZZ-WWW-2014 (also a Wednesday, and not precisely reported due to privacy considerations) for the eighth floor (72 scheduled patients) using the same time period. Since the two floors are occupied by different disease centers, their statistical properties are not identical. In particular, in the considered data set, the average (across patients) mean duration and cancellation probabilities are similar (141 minutes vs. 136 minutes and 1.52% vs. 1.3% for floors eight and nine, respectively), but the average coefficient of variation is higher for the eighth floor (0.77 vs. 0.62).

Table 5 reports the percentage cost decrease that our approach achieves, compared with the means-based approach for different overtime rates $\gamma$. We observe that it delivers uniformly a significant improvement, ranging between 18% and 42% across both infusion units and different ratios of overtime to waiting time costs.

Finally, we remark that, in this paper, we focused on the offline appointment scheduling problem: the list of patients that are to be scheduled for a given day is known in advance. As future work, it
would also be of practical interest to consider an online version of the problem, where patients make appointments asynchronously. In Figure 8, we also plot the overtime-waiting tradeoff (red curve) for the myopic (online) version of the infinite-server heuristic. That is, patients are scheduled in a greedy manner one by one, without any ability to modify appointment times of already scheduled patients. The presented averages are based on 4000 samples, where for each sample patients arrive in random order. The gap between the offline and myopic versions of the algorithm is due to the non-anticipatory nature of the myopic algorithm. (The average relative increase in total cost, compared to the offline version, is approximately 32%.) This gap can be hopefully reduced by employing an online algorithm that anticipates future appointment demand – we leave this topic for a natural and worthy followup research. Additionally, one could also evaluate the impact of patient preferences on the overall performance of appointment-based systems. Such preferences were elicited from patients in Liu et al. (2017), in order to model their willingness to wait for their preferred doctor, as opposed to an earlier appointment by an alternative doctor.

Acknowledgments
The authors acknowledge DFCI support of the Service Enterprise Engineering (SEE) Laboratory at the Technion, as well as laboratory researchers: Dr. Valery Trofimov, Ella Nadjharov, Igor Gavako and Dr. Arik Senderovich. The work of A.M. has been partially supported by BSF grant 2014180 and ISF grant 1955/15. The work of P.M. has been partially supported by NSF grant CMMI-1362630 and BSF grant 2014180.

References


Deng Y, Shen S, Denton B (2017) Chance-constrained surgery planning under uncertain or ambiguous surgery duration, working paper. 2


Appendix A: Data-Driven Robust Scheduling

A.1. Technical Details on Solution Approach

We outline our solution approach for Problem (9). To ease notation, let $\mathcal{A}$ be the set of feasible solutions for (9), that is,

$$
\mathcal{A} = \{ \alpha : \alpha \text{ satisfy constraints (9b–9e)} \}.
$$

We use a Benders decomposition procedure to solve (9). In particular, given a solution $\bar{\alpha}$ to (9), let $d(\bar{\alpha})$ be worst-case durations that are optimal for nature’s problem (8). We apply a Benders cut $z \geq \beta_\alpha(\alpha)$, where

\[
\begin{align*}
\beta_\alpha(\alpha) &:= \min \sum_{i=1}^{n} \sum_{t=1}^{T} (\alpha_{it} - b_{it}) + \gamma \sum_{i=1}^{n} \sum_{t=\tau_{\alpha}+1}^{T} (b_{it} - b_{i,t-d_i(\bar{\alpha})}) \\
\text{subject to} & \quad b \leq \alpha \\
& \quad b_{it} \leq b_{i,t+1}, \quad i = 1, \ldots, n, \quad t = 1, \ldots, \bar{T} - 1 \\
& \quad b_{it} \leq b_{i-1,t}, \quad i = 2, \ldots, n, \quad t = 1, \ldots, \bar{T} \\
& \quad \sum_{i=1}^{n} (b_{it} - b_{i,t-d_i(\bar{\alpha})}) \leq c_t, \quad t = 1, \ldots, \bar{T} \\
& \quad b \text{ binary},
\end{align*}
\]

where $b_{it} := 0$, for all $i = 1, \ldots, n$, and $t \leq 0$. Below we provide a detailed outline of the procedure we use, together with a proof of its convergence.

**Proposition 1.** Procedure 1 terminates after a finite number of steps.

**Proof.** We first show that

$$
W(\alpha) \geq \beta_\alpha(\alpha), \quad \forall \alpha, \bar{\alpha} \in \mathcal{A}.
$$

To this end, fix some $\alpha, \bar{\alpha} \in \mathcal{A}$, and let $b$ be an optimal solution to (10). We now construct a feasible solution to (8) that achieves an objective value equal to $\beta_\alpha(\alpha)$. In particular, let

$$
e_{it} = \begin{cases} 
\delta_i(\bar{\alpha}) + 1, & t = d_i(\bar{\alpha}) + 1, \ldots, \bar{T} \\
0, & \text{otherwise},
\end{cases} \quad \text{and} \quad u_{ij} = \chi_{\delta_i(\bar{\alpha})}, \quad j = 1, \ldots, J,
$$

for all $i = 1, \ldots, n$. We argue that there exists an $f \in \{0, 1\}^\bar{T}$ such that $(u, b, e, f)$ is feasible for (8). Feasibility of $b$ for (10) implies (8b, 8d, 8f). By the definition of $e$, we also get that (8c, 8e, 8g) hold trivially. To show that there exists an $f \in \{0, 1\}^\bar{T}$ such that $(u, b, e, f)$ is feasible for (8), we need to argue that, for all $t = 1, \ldots, \bar{T}$,

$$
\sum_{i=1}^{n} (\alpha_{it} - b_{it}) > 0 \quad \text{implies} \quad \sum_{i=1}^{n} (b_{it} - e_{it}) = c_t.
$$

For the sake of reaching a contradiction, assume that this condition fails for some time $k$. Then,

$$
\sum_{i=1}^{n} (b_{ik} - e_{ik}) < c_k,
$$

and there exists a service index $I$ such that

$$
\alpha_{Ik} = 1 \quad \text{and} \quad b_{Ik} = 0.
$$
**Procedure 1** Benders decomposition approach to solve (9)

**Input:** initial $\alpha^0 \in A$

**Output:** optimal solution $\alpha^*$ to (9)

1. $\bar{z} \leftarrow -\infty$, $k \leftarrow 0$

   while $W(\alpha^k) > \bar{z}$ do

   solve

   $$
   \begin{align*}
   & \text{minimize} & & z \\
   & \text{subject to} & & z \geq \beta_{\alpha,j}(\alpha), \quad j = 0, \ldots, k \\
   & & & \alpha \in A
   \end{align*}
   $$

2. $k \leftarrow k + 1$

3. let $\tilde{\alpha}$, $\tilde{z}$ be an optimal solution to (11)

4. $\alpha^k \leftarrow \tilde{\alpha}$

5. $\bar{z} \leftarrow \tilde{z}$

end while

$\alpha^* \leftarrow \alpha^k$

If multiple such indices exist, let $I$ be the smallest among them. In other words, the $I$th customer is available at time $k$, but he starts getting serviced only at some later time $\tau > k$. That is,

$$
\tau = \min \{ t : b_{It} = 1 \}.
$$

Consider now $\tilde{b}$ such that customer $I$ starts getting serviced at $\tau - 1$ instead, that is

$$
\tilde{b}_{It,\tau-1} = 1 \quad \text{and} \quad \tilde{b}_{It} = b_{It}, \quad i \neq I, t \neq \tau - 1.
$$

Note that $\tilde{b}$ is feasible for (10). Its first constraint is satisfied since $\alpha_{I,\tau-1} \geq \alpha_{Ik} = 1$, which follows from $\alpha \in A$ and $\tau > k$. The second constraint follows from the feasibility of the original solution $b$. The third is relevant only if $I > 1$ and follows from $b_{I-1,\tau-1} \leq b_{I-1,k} = 1$, which is true since $I$ was picked as the smallest index for which $b_{ik} = 0$. Finally, the capacity constraint is satisfied since there is slack capacity at $k, \ldots, \tau - 1$, given that no new service starts at these periods. Given that $\tilde{b}$ achieves a strictly lower objective value for (10), this contradicts the optimality of $b$.

We now show that

$$
W(\alpha) = \beta_\alpha(\alpha), \quad \alpha \in A.
$$

To this end, let $W(\alpha, \bar{d})$ be the optimal value of (8), where $\bar{d}$ is a service duration vector that corresponds to some assignment vector $\bar{u} \in U$, when $u$ is constrained to equal $\bar{u}$. Similarly, let $\beta(\alpha, \bar{d})$ be the optimal value of (10), when $\bar{d}(\alpha)$ is equal to $\bar{d}$. As a first step, note that $W(\alpha, \bar{d}) = \beta(\alpha, \bar{d})$. To see this, recall we argued
above that, for any $\alpha \in \mathcal{A}$, if $b$ is feasible for (10) then there exists a feasible solution to (8) that achieves the same objective, with the same service duration vector. Thus, $W(\alpha, \bar{d}) \geq \beta(\alpha, d)$. Conversely, if $(\bar{u}, b, e, f)$ is feasible for (8), then we have that $e_{it} = b_{i,t+d_i}$ by (8). Thus, $b$ is feasible for (10), with the same durations vector, achieving the same objective value. Consequently, $W(\alpha, \bar{d}) = \beta(\alpha, d)$. To complete our argument, note that

$$W(\alpha) = W(\alpha, d(\alpha)) = \beta(\alpha, d(\alpha)) = \beta_\alpha(\alpha).$$

The statement of the proposition follows from Theorem 2 in Hooker and Ottosson (2003). □

A.2. Benchmarking for Single-Server Instances

We employ an experimental setup similar to Kong et al. (2013). In particular, we consider scheduling $n = 7, 12$ or $15$ customers to be served by a single server. Punctuality is perfect, i.e., $P_i = 0$ with probability one. Service durations are random and follow either a lognormal or gamma distribution. The mean duration of the $i$th customer, $d_i$, is uniformly drawn over $[2/3, 4/3]$. Conditional on its mean $d_i$, the standard deviation of each duration is uniformly drawn over $[2d_i/(3\sqrt{3}), 4d_i/(3\sqrt{3})]$, so that the coefficient of variation is uniform on $[2/3, 4/3]$ (as in Kong et al. (2013)). Business hours are taken to be $T = \sum_{i=1}^{n} d_i / \rho$, where $\rho$ is a utilization parameter taking values $\rho = 0.85, 1.00$ or $1.15$. Overtime cost rate is $\gamma = 2$ and $\gamma = 3$. The DDR parameters are $\Gamma = 0.5$ and $J = 7$. For each possible combination of distribution, number of customers $n$, utilization parameter $\rho$, and overtime cost rate $\gamma$, we generate 20 instances and produce schedules according to the DDR, SO and DR approaches. Various statistics for resulting costs of each approach were obtained by simulating $10^6$ paths. Table 6 displays a relative comparison of DDR’s mean, 75-, 85- and 95-percentile costs with SO’s and DR’s, only for the case of $\gamma = 2$; results for $\gamma = 3$ are very similar. As remarked in Section 6.3, our experiments suggest that the IS approach is slightly superior to the DDR approach.

Appendix B: Means-based Sequencing Approach

Assume given mean service durations for each customer, denoted by $d_1, \ldots, d_n$, and a capacity vector $c_t$, $t = 1, \ldots, \bar{T}$. At a high level, our means-based approach decides on schedule times $a_1, \ldots, a_n$, so as to minimize makespan, namely the time it takes to serve all customers. Such a schedule would be tailored towards minimizing overtime costs. To obtain solutions that would trade off overtime with waiting costs, we scale down by a constant factor the available capacity during business hours $c_t$, $t = 1, \ldots, T$, and re-solve. In so doing, the re-optimized schedule allows for slack capacity to mitigate waiting at the expense of larger makespan (i.e., possible overtime).

Our means-based sequencing formulation models queueing dynamics in a similar fashion to our DDR:

$$\text{maximize} \quad \sum_{t=1}^{\bar{T}} x_t$$
$$\text{subject to} \quad \alpha_{it} \leq \alpha_{i,t+1}, \quad i = 1, \ldots, n, \quad t = 1, \ldots, \bar{T} - 1$$
$$\alpha_{it} = 1, \quad i = 1, \ldots, n, \quad t = T, \ldots, \bar{T}$$
$$\sum_{i=1}^{n} (\alpha_{it} - \alpha_{i,t-d_i}) \leq c_t, \quad t = 1, \ldots, \bar{T}.$$
Table 6  Mean percentage differences (in %) between DDR and SO solutions (DR solutions) for means and percentiles.

<table>
<thead>
<tr>
<th>Mean utilization</th>
<th>Performance measure</th>
<th>Distribution</th>
<th>Number of jobs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>0.85</td>
<td>Mean</td>
<td>Lognormal</td>
<td>+1.8 (-0.1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+3.3 (-4.6)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+3.7 (-3.9)</td>
</tr>
<tr>
<td></td>
<td>Gamma</td>
<td></td>
<td>+1.9 (-0.2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+3.0 (-4.3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+2.6 (-3.7)</td>
</tr>
<tr>
<td>75% percentile</td>
<td>Lognormal</td>
<td></td>
<td>+2.3 (+1.5)</td>
</tr>
<tr>
<td></td>
<td>Gamma</td>
<td></td>
<td>+2.1 (+2.0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+2.6 (-0.6)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+1.8 (-1.0)</td>
</tr>
<tr>
<td>85% percentile</td>
<td>Lognormal</td>
<td></td>
<td>+2.0 (+3.8)</td>
</tr>
<tr>
<td></td>
<td>Gamma</td>
<td></td>
<td>+3.6 (+4.8)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+4.3 (+6.1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+2.7 (+4.0)</td>
</tr>
<tr>
<td>95% percentile</td>
<td>Lognormal</td>
<td></td>
<td>+1.4 (+2.7)</td>
</tr>
<tr>
<td></td>
<td>Gamma</td>
<td></td>
<td>+3.4 (+3.3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+6.1 (+12.5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+4.0 (+9.5)</td>
</tr>
<tr>
<td>1.00</td>
<td>Mean</td>
<td>Lognormal</td>
<td>+1.4 (-3.6)</td>
</tr>
<tr>
<td></td>
<td>Gamma</td>
<td></td>
<td>+1.6 (-1.4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+2.0 (-2.6)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+2.8 (-1.7)</td>
</tr>
<tr>
<td>75% percentile</td>
<td>Lognormal</td>
<td></td>
<td>+1.4 (+5.4)</td>
</tr>
<tr>
<td></td>
<td>Gamma</td>
<td></td>
<td>+0.6 (+4.5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+1.9 (+1.5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+1.8 (+1.4)</td>
</tr>
<tr>
<td>85% percentile</td>
<td>Lognormal</td>
<td></td>
<td>+1.3 (+9.7)</td>
</tr>
<tr>
<td></td>
<td>Gamma</td>
<td></td>
<td>+0.3 (+6.8)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+2.1 (+5.3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+1.9 (+4.6)</td>
</tr>
<tr>
<td>95% percentile</td>
<td>Lognormal</td>
<td></td>
<td>+0.8 (+8.3)</td>
</tr>
<tr>
<td></td>
<td>Gamma</td>
<td></td>
<td>−0.0 (+5.3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+1.8 (+5.9)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+1.4 (+5.2)</td>
</tr>
<tr>
<td>1.15</td>
<td>Mean</td>
<td>Lognormal</td>
<td>+0.8 (-1.4)</td>
</tr>
<tr>
<td></td>
<td>Gamma</td>
<td></td>
<td>+1.0 (-0.9)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+1.4 (-2.0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+1.5 (-2.0)</td>
</tr>
<tr>
<td>75% percentile</td>
<td>Lognormal</td>
<td></td>
<td>+0.6 (+1.8)</td>
</tr>
<tr>
<td></td>
<td>Gamma</td>
<td></td>
<td>+0.4 (+1.2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+0.9 (+1.5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+0.1 (+0.7)</td>
</tr>
<tr>
<td>85% percentile</td>
<td>Lognormal</td>
<td></td>
<td>+0.4 (+3.2)</td>
</tr>
<tr>
<td></td>
<td>Gamma</td>
<td></td>
<td>+0.1 (+1.7)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+0.6 (+3.5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>−0.7 (+2.1)</td>
</tr>
<tr>
<td>95% percentile</td>
<td>Lognormal</td>
<td></td>
<td>+0.0 (+2.7)</td>
</tr>
<tr>
<td></td>
<td>Gamma</td>
<td></td>
<td>−0.2 (+1.0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+0.5 (+3.1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>−0.9 (+1.6)</td>
</tr>
</tbody>
</table>

\[ x_t \leq x_{t+1}, \quad t = 1, \ldots, \bar{t} - 1 \]
\[ x_t \leq \frac{1}{n} \sum_{i=1}^{n} \alpha_{i,t-d_i}, \quad t = 1, \ldots, \bar{t} \]
\[ \alpha, x \text{ binary;} \]

here \( \alpha_{it} := 0 \), for all \( i = 1, \ldots, n \), and \( t \leq 0 \). Variables \( \alpha \in \{0, 1\}^{n \times T} \) have the same interpretation as before. Variables \( x \in \{0, 1\}^{T} \) indicate whether all customers have been served by time \( t \).