Practical Online Policies for Scheduling Patients at a Scanning Facility Under High Demand

Simran Lakhani[a,*], Ashutosh Mahajan[a], Akshay D Baheti[b], Suyash Kulkarni[b]

[a] *Indian Institute of Technology Bombay, IEOR building, IIT Area, Powai, Mumbai, 400076, Maharashtra, India*
[b] *Tata Memorial Hospital, Homi Babha Building, Parel, Mumbai, 400012, Maharashtra, India*

[*] Corresponding author
Email addresses: simran@iitb.ac.in (Simran Lakhani)

Preprint submitted to Computers & Industrial Engineering *April 16, 2024*

Practical Online Policies for Scheduling Patients at a Scanning Facility Under High Demand

Abstract

We study appointment scheduling in a resource-constrained, high-demand hospital scanning facility in which machine idleness is undesirable. We contend with two uncertainties: unforeseeable patient arrivals and random variations in scan durations across different patient types. The objective is to balance the competing goals of minimizing patient waiting time and idle time of the machine (desired to be near zero). Based on the data available from the facility, patients are divided into three types, and three different practical policies are proposed for scheduling: the first one fills all slots one by one, the other uses the best-fit variant of online bin-packing, and the third one assigns high variance patients to later slots while maintaining predefined probability thresholds. To accommodate scan time variability, some slots are overbooked. The proposed appointment policies are simulated and compared against the existing one. Additionally, an offline optimization model is solved to find the optimal schedule under the assumption that the arrival stream of patients was known beforehand. Chance-constrained programming is used to account for random service times. Our simulations show that the third online policy performs almost as well as the optimal offline policy and achieves a considerable reduction in the average waiting times of patients.

Keywords: Patient scheduling, Uncertain service times, Chance-constrained optimization, Discrete event simulation.

1. Introduction

We consider appointment scheduling for a Magnetic Resonance Imaging (MRI) machine in a tertiary care cancer hospital with a high patient load. The MRI machine serves patients from all departments in the hospital and is capable of performing different types of scans. A stream of patients arrives at the appointment desk, and the secretary allots time slots to patients one by one, without having the foresight of the type of patients that may arrive subsequently. The secretary makes two decisions: (i) the date of the scan based on availability and type of patient, and (ii) the time slot for the selected day. This study is limited to the latter only (time slot selection). Given the high load and the hospital's objective of serving as many patients at subsidized prices as possible and the variability in scan times, the patients are called early to avoid the machine being idle. As a result, the patients have long waiting times on the day of their scan. The objective of this study is to balance the competing goals of minimizing patient waiting time and negligible idle time of the machine.

The day of the scan is usually based on the doctor's recommendation, for example, a cancer patient may require a scan after a prescribed number of weeks for a follow-up visit. This decision is out of scope of this study. The time slot of the scan is selected by the secretary based on how many patients have already been scheduled on that day. This decision is thus affected by the arrival sequence of patients at the appointment desk. Another source of randomness (Gupta and Denton, 2008) that should be accounted for while assigning a time slot is the variability in the scan times. This randomness is observed during the day of the scan due to the patient's condition, specific recommendations of the doctor, the condition of the machine, and other factors.

In our study, we focus on the secretary's problem of assigning patients to pre-defined identical-length slots, known as block scheduling (Cayirli and Veral, 2003). We contend with two uncertainties, the arrival stream of patients for booking an appointment, and randomness in service times on the day of scan. We first classify the patients based on their scan times distribution. We then propose policies for alloting time slots and evaluate them through simulation. We also present an optimization model to minimize patient waiting times and capture machine idleness using chance constraints. The model is designed to find the optimal schedule if the complete arrival stream of patients was hypothetically known beforehand. This means we know the number of requests for each type of scan, also known as offline scheduling (Pham et al., 2023). Our simulations show that the recommended online policy performs nearly as well as the optimal offline policy, thus effectively mitigating the impact of randomness in the arrival stream of patients at the time of booking appointments.

In the remainder of this paper, we start with an overview of the literature on appointment scheduling, specifically MRI scheduling in Section 2. In Section 3, we describe the current scheme observed by the hospital and propose online policies for scheduling patients into the pre-defined slots. In Section 4, we present a simulation-based study to evaluate the impact of proposed policies and compare it against existing policy. In Section 5, we present the offline optimization model. In Section 6, we compare the online and offline policies based on key performance metrics. Finally, we discuss the implications of our work in Section 7.

2. Literature review

Appointment systems are usually used in hospitals to mitigate overcrowding and long waiting times for patients. Extensive research, as outlined in review papers (Cayirli and Veral, 2003; Gupta and Denton, 2008; Ahmadi-Javid et al., 2017), investigated the domain of medical appointment scheduling. According to Cayirli and Veral (2003), the appointment system consists of several decisions concerning the appointment rule, any patient classification, and the adjustments made in the schedule to cope with environmental factors, such as walk-ins, emergencies, and no-shows. The appointment rule consists of decisions regarding the block size (number of patients to be scheduled in each appointment interval) and the time interval between two successive appointments which can either be fixed or variable. We consider the variable block size while keeping the appointment intervals fixed, in contrast with the pioneering works of Bailey (1952), where two patients are scheduled in the initial block, and the remaining patients are scheduled one at a time at equal time intervals. Later White and Pike (1964); Soriano (1966); Cox et al. (1985) proposed scheduling a fixed number of patients to each appointment slot with appointment intervals kept constant. The majority of these studies address the scheduling of a single-patient class and focus on the impact of policies on both patient and physician waiting times.

We utilize patient classification to simplify the scheduling of the patients. As highlighted in Cayirli and Veral (2003), few studies classify patients based on new/follow-up, variability of service times, and type of procedures. Our classification shares some similarities with Bhattacharjee and Ray (2016), where they consider appointment scheduling for multiple classes of patients, their punctuality, no-show rates, and service time probabilities for an MRI machine. They suggest adjusting the inter-appointment times according to the mean service times of patients. This adjustment is difficult to implement in our system because the patients are allotted slots. Green et al. (2006) consider a diagnostic facility, assuming a constant scan time for all patients. Even under these simplistic settings, finding an optimal appointment policy is hard. More recent studies (Sauré and Puterman, 2014; Cappanera et al., 2019) considered the classification of patients based on wait time targets, and perform priority-based scheduling. We do not prioritize any type of patient and no reservation of slots is done for any class of patients.

The studies by Green et al. (2006); Sauré and Puterman (2014); Cappanera et al. (2019) do not account for randomness in scan times, instead they consider random arrivals. In our case, the average scan time for each type of scan, as well as their standard deviations are known and considered in the study. One of the recent studies by Benjaafar et al. (2023) characterizes an optimal appointment schedule under the constraints on the waiting time of each customer. Moreover, their study is limited to a single class of customers. We on the other hand, have constraints on machine idleness because the system is highly resource constrained. Addis et al. (2014), survey stochastic and robust optimization models for scheduling surgeries taking into account uncertain surgery durations. They also consider only a single patient class. They use the approach suggested by Bertsimas and Sim (2004) that does not require information on probability density functions of service times. We, on the other hand, utilize the method of Ben-Tal and Nemirovski (2000) based on the information available regarding the problem setting to develop an offline optimization model.

As the appointment scheduling problem is inherently complex (Cayirli and Veral, 2003; Gupta and Denton, 2008), a fixed policy or set of rules may not always be effective (Patrick et al., 2008). Hence, a dynamic and flexible approach needs to be designed to allocate patients to the pre-designed slots in an online fashion. Pham et al. (2023) reviews several papers that consider uncertainty in service times, but not the uncertain patient requests. In their study, they contend with the stochastic patient arrivals for Radiotherapy Treatment appointments for different classes of patients. The patients are prioritized based on their waiting time targets. They propose an online machine learning based scheduling approach and train their model using a mathematical optimization model. However, their study considers giving a date of the appointment and we propose the appointment time for a given day. Smedira and Shmoys (2022), addressed the online appointment scheduling problem by performing two separate steps of scheduling and rematching slots to avoid overlap. However, in our setting, such an approach would be difficult to implement as it would require informing patients of their updated slots, which can lead to new logistical challenges. Complications in appointment systems increase when the facility also has to cater to walk-ins, no-shows, and unpunctual patients. One of the recent studies (Chen et al., 2018) on intraday scheduling considers the overbooking of slots to mitigate the impact of no-shows using a stochastic optimization model. However, they suggest a dome-dome-dome pattern with alternate long and short time slots, in contrast to our fixed time slots. We do not consider these factors in the present study. Some of these aspects are covered in the works mentioned above.

3. Description of system and online scheduling policies

In the currently followed system at the hospital, patients are assumed to be identical irrespective of the required scan. Block scheduling is implemented with each block of two hours. Patients are scheduled considering the average service time of 45 minutes per scan. To avoid machine idleness, several patients are scheduled in a slot. Having a slot with several patients naturally results in more waiting as compared to calling each patient at a different time (Bailey, 1952; Soriano, 1966). However, the hospital still prefers it because it leads to a lower idle time of the machine, which is important for the limited resources at the hospital. The scanning facility operates for about 22 hours a day, scanning about 30 cancer patients daily. Most of the patients are out-patients with a few in-patients who are given priority. Many patients pay only a subsidized fee at the hospital, and they do not have alternate options for getting a scan. Therefore, some amount of waiting is acceptable in this setting. In the current system, the secretary tries to schedule an equal number of patients in each slot (see Section 3.1).

Parameter	Type A	Type B	Type C
N_p	(11, 20)	(8, 17)	(4, 12)
r_p	0.463	0.334	0.202
μ_p (mins)	36.59	44.82	56.80
σ_p (mins)	15.60	11.65	18.23

Table 1: Parameters for simulation

In our study, we consider on similar lines, S identical and contiguous service slots each of length L (120) minutes. The appointments are scheduled for a single day and the available working time of the machine is T (1320) mins. Patients arrive to book appointments for their respective scans $7-14$ days in advance. The secretary allots each patient to one of these slots. A slot may have several patients. There are about 20 different types of scans each taking different amounts of time. In order to tackle the scheduling problem better, we classify patients into three categories. The scan types are divided into three categories based on their average service times. Let $P := \{A, B, C\}$ be the set of categories of patients. Each arriving patient i is mapped to one of the three categories in P. Let the average scan times and the standard deviation of category p be denoted as μ_p, and σ_p respectively (Table 1). We denote by r_p for $p \in P$ the probability that the next patient arriving for booking an appointment is of category p. We assume that the arrivals of the three patient categories for scheduling a scan are independent of each other.

Given the fixed length slots and an online arrival stream of patients to book appointments, the problem can be viewed as an online covering problem to cover all slots with a small amount of overbooking to avoid the machine being idle (Chen et al., 2018). In an online covering problem, a set of elements must be assigned to a subset of available slots dynamically and sequentially (Alon et al., 2003). Similarly, in our case, at each step, the secretary must allocate a patient to a slot based on the available information of the scan type, without knowing the complete list of future patients. The secretary updates the available slots after each assignment, and the goal is to fill all slots while balancing the waiting time of patients. While scheduling an equal number of patients in each slot may seem like the simplest approach (Cayirli and Veral, 2003), the variable service times for each type of scan and the unpredictability of patient arrivals (for appointments) in an online setting make it difficult. If the number of slots is fixed for each category of patient, there is a possibility of higher idle time for the MRI machine (Cappanera et al., 2019). Therefore, we try to schedule different types of patients in the same slot.

We define the utilization of slot s, denoted by U_s, as the total service time of the patients scheduled in slot s. U_s is random due to uncertainty in service times. The expected utilization of slot s is denoted by, $u_s = \mathbb{E}[U_s]$. Note that U_s does not include any time spent to serve patients spilled over from the previous slots. One way of overbooking is to assume that the total time available for scheduling in a slot is $L + b$ mins, where b is the allowed or targeted overbooking. Assuming every slot can be overbooked differently, let b_s be the targetted overbooking time in slot s. We consider overbooking only the initial k slots, i.e., the overbooking buffer is $b_s = \beta$, $s \le k$, and 0 otherwise in some policies proposed next. This overbooking buffer can implicitly account for variance in scan times. Further, the amount of buffer in each slot should be decided such that it does not exceed the day's limit of T mins, i.e., $k(L+\beta) + (S-k)L \le T$.

3.1. Existing Policy

In the currently used policy, patients are assumed to be identical when allocating slots. Each time slot is of length $L = 120$ minutes. The secretary tries to schedule an equal number of patients across each slot, regardless of the scan type requested. The current practice is to schedule four patients in each of the initial six slots, starting from 8:00 AM until 6:00 PM (Algorithm 1). About $6-7$ patients are allocated to the seventh slot, 8:00-10:00 PM. The scanning facility operates until all scheduled patients are scanned, which takes upto 6AM the next morning. This policy is observed to result in high waiting times for almost all patients. We next describe three proposed policies for decreasing the waiting times while still avoiding the idle time of the facility.

Algorithm 1: Existing Policy

Input: Patient i
Number of patients scheduled in all slots num_s, $s \in \{1, 2, \ldots, 7\}$
Output: Slot s for patient i
Parameter: L
Initialize: $s \leftarrow 1$, $allot \leftarrow$ **False**
for $s \in \{1, 2, \ldots, 6\}$ **do**
  if $num_s < 4$ **then**
    Add patient i to slot s
    $allot \leftarrow$ **True**
    $num_s \leftarrow num_s + 1$
    break
  end
end
if $allot =$ ***False*** **then**
  Add patient i to slot 7
  $num_s \leftarrow num_s + 1$
end

3.2. Fill Slotwise with Overbooking (FAS with OB)

This policy allocates the earliest available time slot to a patient. Patients are assigned the earliest slot s for which expected utilization u_s is less than target $L+b_s$. The policy follows a first-come, first-available slot for each patient as described in Algorithm 2. Note that, this policy allows for some overbooking in all slots. For example, if two patients have been scheduled in slot s and u_s is only slightly less than L, the third patient will be added to slot s, irrespective of their scan type.

Consider an example of scheduling patients to five slots, each of length 120 mins ($S = 5, L = 120$). Suppose overbooking

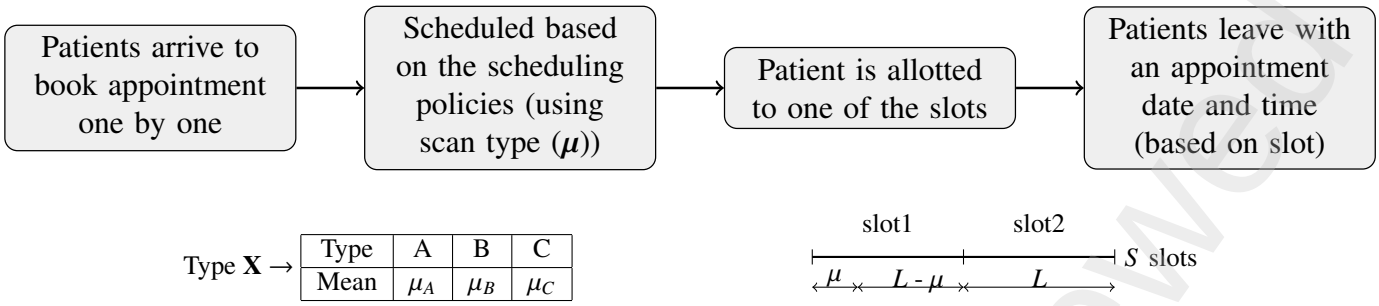

Figure 1: Flow of patients for booking appointments

Algorithm 2: FAS with OB

Input: Patient i of type p with average scan time μ_p, $p \in P$
Current utilization of all slots u_s, $s \in \{1, 2, \ldots, S\}$
Buffer for all slots b_s, $s \in \{1, 2, \ldots, S\}$
Output: Slot s for patient i
Parameter: L, S, and T
Initialize: $s \leftarrow 1$
while $s \leq S$ *and* $\sum_{s \in S} u_s < T$ **do**
  if $u_s < L + b_s$ **then**
    Add patient i to slot s
    $u_s \leftarrow u_s + \mu_p$
  end
  s = s+1
end

buffer β is 15 and k is 2. The total available time for scheduling is $T = 630$ mins. Suppose the patients arrive one-by-one in an online fashion as shown in Table 2. Assume the three categories have mean times, $\mu_A = 45$, $\mu_B = 40$, and $\mu_C = 35$. The schedule according to this policy is shown in Table 3. Note that the last slot is underutilized, and the next patient in the arrival stream cannot be scheduled as the limit of 630 minutes is reached.

3.3. *Bin Packing all slots and then Overbooking (BinPack then OB)*

This policy adapts the concept of the online bin packing problem to schedule patients. We treat appointment slots as bins with fixed capacities, and patients as items to be packed into these bins. Heuristics and approximate algorithms have been proposed in the literature for solving the online bin packing problem (Gupta and Ho, 1999). In our policy, a patient is scheduled in a slot if the patient's average service time can be completely accommodated in the slot, akin to packing items in bins. However, if the total average service time of all scheduled patients is less than the target, then some slots may remain underutilized. To address this issue, we overbook the initial k slots if the expected utilization u_s of the slot is less than target $L + b_s$ as described in Algorithm 3. The first overbooking is done in the slot that has the minimum expected utilization, and the process continues until either all k slots are overbooked or the available time limit of T mins is reached.

We can also consider a variant of Algorithm 3 that involves packing slots without any buffer time added to initial slots, such that the length of each slot is L. The procedure of scheduling remains the same as described above except that, $b_s = 0, \forall s \in S$. Considering the similar arrival stream of patients as in Table 2, the schedule according to BinPack then OB policy and its variant is shown in Table 4 and Table 5 respectively. Note that the overbooking is done after all the bins are packed. The coloured numbers represent the patients that are overbooked in the respective slots and updated utilization of those slots. Further patients cannot be scheduled as there is no slot left for booking as well as the total scheduling time is exceeded. Note that fewer patients are booked in the variant policy. This may lead to underutilization of the facility. This variant could be preferable in systems where patient waiting time is more valuable. However, we do not consider this variant ($b_s = 0, \forall s$) for further experiments as the facility under consideration is resource-constrained.

Algorithm 3: BinPack then OB

Input: Patient i of type p with average scan time μ_p, $p \in P$
Current utilization of all slots u_s, $s \in \{1, 2, \ldots, S\}$
Buffer for all slots b_s, $s \in \{1, 2, \ldots, S\}$
Output: Slot s for patient i
Parameter: L, S, and$T$
Initialize: *allot* $\leftarrow$ **False**
for $s \in \{1, 2, \ldots, S\}$ **do**
  if $u_s + \mu_p \leq L + b_s$ **then**
    Add patient i to slot s
    allot $\leftarrow$ **True**
    $u_s \leftarrow u_s + \mu_p$
    break
  end
end
if *allot* = ***False*** *and* $\sum_{s \in S} u_s < T$ **then**
  $a = \arg\min_{s \in 1,2,\ldots,S}(u_s)$
  if $u_a < L + b_a$ **then**
    Add patient i to slot a
    $u_a \leftarrow u_a + \mu_p$
  end
end

Patient ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Patient type	C	A	B	A	A	A	A	A	B	C	B	C	A	A	A	A	C	B	B	A
Average scan time(mins)	45	35	40	35	35	35	35	35	40	45	40	45	35	35	35	35	45	40	40	35
Std dev of scan time(mins)	15	12	8	12	12	12	12	12	8	15	8	15	12	12	12	12	15	8	8	15

Table 2: An example of an arrival stream of patients for booking one of the five slots.

Slots	Patients scheduled	Exp utilization of the slots (u_s mins)
Slot 1	1, 2, 3, 4	155
Slot 2	5, 6, 7, 8	140
Slot 3	9, 10, 11	125
Slot 4	12, 13, 14, 15	150
Slot 5	16, 17	80

Table 3: Schedule of patients from FAS with OB for the example in Table 2.

3.4. Variance Based Slot Allocation

This policy explicitly incorporates the variance in service times while scheduling appointments. Patients characterized by high variance (Type C) in service times are scheduled into later slots, while other patients (A and B) are assigned the earliest available slots. Further, bin packing is not used for the remaining patients. Rather, a patient is assigned a slot s if the probability of serving all the patients scheduled up to slot s before the end of slot s is more than a predefined threshold ζ.

$$P\left(\sum_{i=1}^{s} U_i \leq s \times L\right) > \zeta \tag{1}$$

The above probability can be approximated easily because the term $\sum_i U_i$ is approximately normal with mean and standard deviation derived from patients scheduled so far. Higher variance patients are scheduled from the v*th* slot onwards, where the parameter v is carefully selected in advance. Additionally, only one higher variance patient is scheduled per slot from the v*th* slot onwards (Algorithm 4). In cases where the number of such patients exceeds available slots, some slots are allotted to two such patients (starting from the last slot). Note that during the backfilling process, the above probability condition is not checked. Considering the similar arrival stream of patients as in Table 2, the schedule according to this policy is shown in Table 6. Type C patients are scheduled from the third slot onwards, and the threshold probability is taken to be 0.1 ($v = 3, \zeta = 0.1$).

Algorithm 4: Variance Based Slot Allocation

Input: Patient i of type p with average scan time μ_p and variance σ_p^2, $p \in P$
Current utilization of all slots U_s, $s \in \{1, 2, \ldots, S\}$
$V = \{v, v+1, \ldots, S\}$
$p_{max} = \arg\max_{p' \in P} \sigma_{p'}^2$
Output: Slot s for patient i
Parameter: v, ζ, L, S, and T
Initialize: *allot* ← **False**
Number of high var patients in slot s, $N^s_{p_{max}} = 0$
for $s \in \{1, 2, \ldots, S\}$ **do**
  if $\mathbb{P}(\sum_{i=1}^{s} U_i \leq s \times L) > \zeta$ **then**
    if $p = p_{max}$ **then**
      if $s \in V$ **then**
        Add patient i to slot s
        allot ← **True**
        $N^s_{p_{max}} \leftarrow N^s_{p_{max}} + 1$
        $V \leftarrow V \setminus \{s\}$
      end
    end
    else
      Add patient i to slot s
    end
  end
end
if *allot* = ***False*** **then**
  for $s \in \{S, S-1, \ldots, v\}$ **do**
    if $N^s_{p_{max}} < 2$ **then**
      Add patient i to slot s
      $N^s_{p_{max}} \leftarrow N^s_{p_{max}} + 1$
    end
  end
end

4. A simulation study: Evaluation of policies

We now describe a simulation-based study of the above four policies. Each working day is divided into eleven time slots, each of length two hours, i.e., $S = 11$ and $L = 120$ mins. This setup reflects the existing slot pattern.

4.1. Instances and scenarios generation

Random input instances are generated based on data obtained from the MRI facility operations to simulate the impact of the

Slots	Patients scheduled	Exp utilization of the slots (u_s mins)
Slot 1	1, 2, 3, 17	120 → 165
Slot 2	4, 5, 6, 16	105 → 140
Slot 3	7, 8, 9	110
Slot 4	10, 11, 13	120
Slot 5	12, 14, 15	115

Table 4: Schedule of patients from BinPack then OB for the example in Table 2.

Slots	Patients scheduled	Exp cumulative u_s (mins)	$P(\sum_{i=1}^{s} U_i \leq s \times L)$
Slot 1	2, 3, 4, 5, 6	180	0.008
Slot 2	7, 8, 9	290	0.05
Slot 3	1, 11, 13	410	0.09
Slot 4	10, 14, 15, 16	560	0.03
Slot 5	12, 17, 18	690	0.005

Table 6: Schedule of patients from Variance-based policy for the example in Table 2.

above policies. The data obtained from the facility included the number of patients with their scan types and service times collected over a period of seven days. There are three quantities required for simulating a day of operations: (i) The number of patients of each of the three categories that are to be scheduled on a day, (ii) the sequence in which these patients arrive at the appointment desk and (iii) the time taken to serve each of these patients.

In our study we first randomly generate a patient-mix, that is the number of patients N_A, N_B, and N_C of categories A, B, and C respectively that will be scanned in a single day. We call such a patient- mix an "instance" and denote it as $N_A - N_B - N_C$. We generate 10 random instances for our study based on the observed frequency of patient categories (Table 1). The total expected time to serve all patients in all these instances is close to the total daily operational time (22 hours).

For each instance generated above, we next generate 30 random permutations of the sequence in which patients arrive at the appointment desk. Each of these permutations is called "scenarios". Lastly, we execute the four policies on each scenario 1000 times. In each execution, the service time of each patient is randomly drawn from the normal distribution $\mathcal{N}(\mu_p, \sigma_p)$.

A Discrete Event Simulation (DES) model (Figure 2) is designed to simulate the schedules generated by the policies to compare key performance metrics. The parameters for scheduling and simulation are summarized in Table 7 and Table 1 respectively. To evaluate the impact of the policies, we consider the total idle time of the machine over the day, the average waiting time of patients, and the overtime required to complete the service of all scheduled patients. We assume that all the patients scheduled in their respective slots arrive at the beginning of the slot (Robinson and Chen, 2003). If the scheduled patients are not served within the 22 hours time frame, the staff (and machine) will continue to complete all the scheduled scans, and

Slots	Patients scheduled	Exp utilization of the slots (u_s mins)
Slot 1	1, 2, 3	120
Slot 2	4, 5, 6, 16	105 → 140
Slot 3	7, 8, 9	110
Slot 4	10, 11, 13	120
Slot 5	12, 14, 15	115

Table 5: Schedule of patients from Variant of BinPack then OB for the example in Table 2.

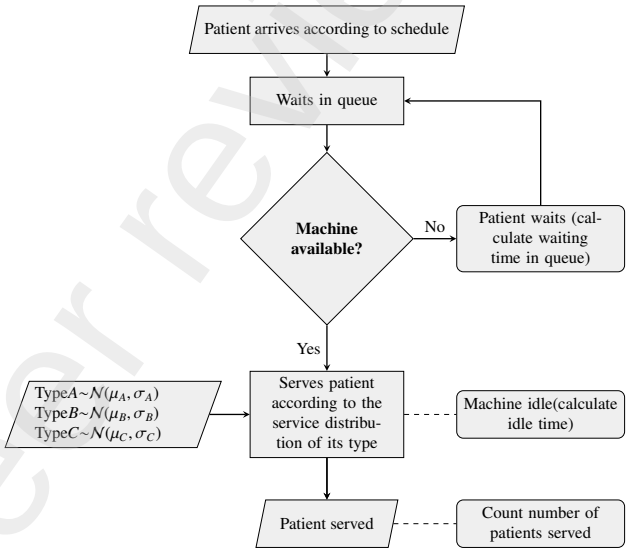

Figure 2: Flowchart of events in simulation model

Parameter	Value
Machine available time T	1320 mins
Total number of slots S	10
Slot length L	120 mins
Amount of buffer β	15 mins
No. of slots overbooked k	5
Threshold ϵ	0.1

Table 7: Parameters for online scheduling policies

the working time beyond the 22 hours is regarded as overtime.

4.2. Comparison of policies

The policies are tested on instances and scenarios described above. All the algorithms listed in Section 3 are implemented in Python. Table 8 demonstrates average key performance metrics across 1000 random realizations for each scenario. We set $\epsilon = 0.1$ for variance-based slot allocation policy (Algorithm 4), as with the increase in ϵ, idle time and overtime increase. For FAS with OB and Variance Based policies, all patients are automatically assigned to slots 1-10 only. This is natural since we do not want idleness in any of the initial slots. For BinPack

then OB, we explicitly do bin-packing only until slot 10 and then overbook k initial slots, leaving no patient for slot 11. The results indicate that all proposed online policies outperform the existing one. This improvement is primarily due to the scheduling strategy, wherein the proposed policies distribute patient appointments more evenly over all slots, in contrast to the current practice of alloting patients to limited time slots (Algorithm 1).

As evident from Table 8, both the FAS and BinPack policies are effective in ensuring low machine idle times. The overbooking of slots ensures high machine utilization. While the FAS with OB policy exhibits improvement over the existing policy, it still results in significant patient waiting times, due to overbooking across all appointment slots throughout the day. In contrast, the BinPack then OB policy limits overbooking to the initial slots. BinPack policy dominates the FAS policy concerning the average waiting time metric, reducing it by 17%. However, the BinPack policy also has a significant waiting time of about 3 hours on average. This can be attributed to the significant variance in scan times in the initial slots. Nevertheless, both policies improve upon the current waiting times of 4.5 hours on average. Among the three proposed policies, the variance-based slot allocation policy consistently outperforms others across all metrics, particularly in reducing patient waiting times.

The variance-based policy improves patient waiting times by scheduling high-variance patients in later slots. This shift helps mitigate the overall variance in waiting times experienced by patients. The wait times are nearly half compared to the existing policy. While this policy may result in slightly higher idle times compared to other policies, the maximum idle time remains at an acceptable value of 5 mins across all instances. Moreover, the consistently low standard deviation across all metrics validates the effectiveness of scheduling high-variance patients in later slots. Note that, in line with the policy descriptions, there is a decrease in waiting times as one moves towards the right in Table 8, and a slight increase in machine idle times. The overtime from all policies remains proportional to the number of patients to be served in a day. Although variance-based policy eliminates the decisions regarding the number of slots to overbook and the extent of overbooking in each slot, it requires an additional computation of probabilities before slot allocation (Algorithm 4). However, this can be approximately done using a normal distribution table and is computationally inexpensive.

Figures 3, 4, 5, and 6 show the average performance metrics for 30 scenarios for the 13-11-6 instance. Figures 3, 5 showcase clear differentiation among policies, highlighting the impact of each policy on waiting times. Specifically, even the highest waiting time under the variance-based policy is approximately 38% lower than the lowest waiting time observed under the existing policy (Figure 5), indicating significant improvement. Moreover, the existing policy demonstrates a wider spread of waiting times compared to the more consistent and narrower range observed under the variance-based policy. Additionally, a simple improvement of evenly scheduling patients across all slots, as implemented in the FAS policy, outperforms the existing policy. The overlapping of idle time plots (Figure 4, 6) indicates that machine utilization is not compromised while improving patient waiting times. For this particular instance, the variance-based policy achieves a 2-hour reduction in wait times by a mere 2-minute increase in machine idle time compared to the existing policy.

5. Offline Scheduling Models

We now consider the problem of assigning patients to the slots in an offline setting, i.e. we assume that the number of patients of each category is known before slots are allotted to patients. In such an offline scheduling, the secretary assigns slots to patients after observing all the demands for that day. A chance-constrained mathematical optimization model is formulated below for minimizing overflows from the slots while putting constraints on the probability of the machine being idle. Implementing an offline policy where patients are only assigned a date of scan at first and the slots are announced a few days later is quite difficult to implement. We use the output of the offline model to merely benchmark the online policies proposed in Section 3. The objective is to minimize expected patient waiting times and ensure the probability of the machine's idleness remains below a specified threshold.

The slots are contiguous, and the overflow of time in serving patients in a particular slot implies a delay in starting the service of patients in the subsequent slots. To restrict machine idleness up to a threshold value in a day, we impose constraints on each slot individually i.e. the probability of the machine being idle in each slot is below a given threshold. However, this will result in scheduling more patients in a slot than its average capacity to serve (L minutes), resulting in waiting times for patients in subsequent slots. The objective function aims to minimize the overflow of time required to serve patients scheduled within a slot, beyond the slot length L thus indirectly preferring low waiting times. Note that the service time X_p for scan type $p \in P$ is assumed to follow a normal distribution with mean μ_p and standard deviation σ_p.

5.1. Model formulation

Let N_p denote the number of patients of type $p \in P$ to be scheduled across S slots within the planning horizon of T minutes. We formulate the offline model using integer decision variables $n_{ps} \in \mathbb{Z}$, representing the number of patients of type p scheduled in slot s. We persist with the definition of the utilization of a slot s, as the total time required to serve all the patients assigned to slot s, denoted by U_s (Section 3). We further define the overflow of slot s, denoted by O_s, as the spillover of time to complete the service of patients scheduled in slot s. Overflow O_s of a slot s is computed as the positive difference between utilization of a slot s along with any spillover from slot $s-1$ and slot length of slot s, as follows:

$$O_s := (U_s + O_{s-1} - L)^+ \quad \forall s \in \{1, \ldots, S\} \qquad (2)$$
$$\text{and} \quad O_0 := 0.$$

Instance	Performance Metric	Existing		FAS with OB		BinPack then OB		Variance-based	
		Average	Std dev	Average	Std dev	Average	Std dev	Average	Std dev
11-10-8	WT	260.4	10.53	186.44	20.2	158.23	10.58	126.93	3.57
	IT	2.98	0.48	4.08	1.15	3.98	1.69	4.67	0.76
	OT	0	0	0	0	0	0	0	0
11-14-5	WT	256.38	11.07	195.29	17.14	163.13	10.89	127.76	4.27
	IT	1.95	0.51	2.76	1.26	2.91	1.27	4.05	0.77
	OT	4.99	2.12	6.01	3.04	5.25	2.29	7.35	2.59
12-8-9	WT	262.50	12.91	192.17	20.144	159.31	10.67	130.61	3.06
	IT	2.78	0.57	4.07	1.27	4.16	1.52	4.16	0.54
	OT	1.25	1.62	1.86	1.99	1.65	1.55	2.91	1.96
12-13-5	WT	252.21	10.36	193.93	16.63	159.01	11.64	129.83	3.86
	IT	2.70	0.49	3.60	1.23	3.62	1.58	4.21	0.64
	OT	0.03	0.11	0.74	1.31	0.89	1.63	0.74	1.26
13-11-6	WT	251.60	10.98	197.05	16.76	157.99	10.01	132.19	3.31
	IT	2.66	0.63	3.60	1.14	3.78	1.71	3.97	0.59
	OT	2.49	1.76	3.07	2.51	3.63	2.62	4.76	2.16
14-9-7	WT	260.15	10.69	188.58	18.53	162.92	8.65	132.32	3.29
	IT	2.25	0.39	4.19	0.61	3.36	1.23	4.09	0.51
	OT	6.44	2.33	7.28	2.44	6.98	2.72	8.16	1.92
15-10-6	WT	258.51	11.49	196.51	17.92	172.94	12.91	134.20	3.22
	IT	1.43	0.7	2.17	1.22	3.13	2.54	3.53	0.56
	OT	31.47	2.24	31.71	2.09	26.2	13.52	33.63	2.66
15-12-4	WT	246.90	10.67	197.23	19.19	159.78	12.03	133.22	3.64
	IT	2.15	0.57	3.04	1.2	3.63	2.31	3.48	0.52
	OT	8.17	2.33	9.52	2.62	9	3.58	10.06	2.47
16-11-4	WT	241.98	10.31	198.17	13.66	156.21	10.54	133.39	3.59
	IT	2.86	0.61	3.27	1.01	4.15	2.02	3.82	0.64
	OT	1.98	1.68	1.5	1.5	2.53	1.93	3.3	2.07
17-9-5	WT	245.23	11.81	190.05	16.22	157.91	10.76	136.79	3.56
	IT	2.84	0.8	3.82	1.11	4.21	2.84	3.74	0.49
	OT	6.18	1.99	7.57	2.95	5.72	3.61	6.75	2.64

Table 8: Waiting time (WT (mins)), Idle time (IT (mins)), and Overtime (OT (mins)) from online policies across 30 scenarios for all instances

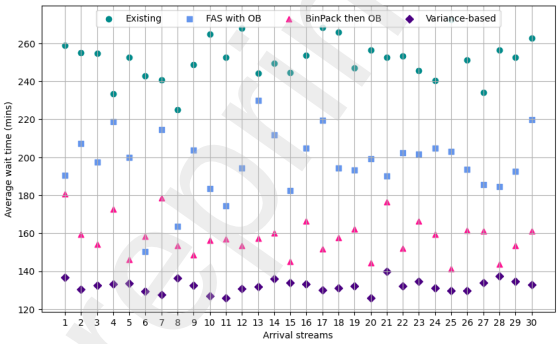

Figure 3: Average wait times (mins) across 30 scenarios for 13-11-6 instance

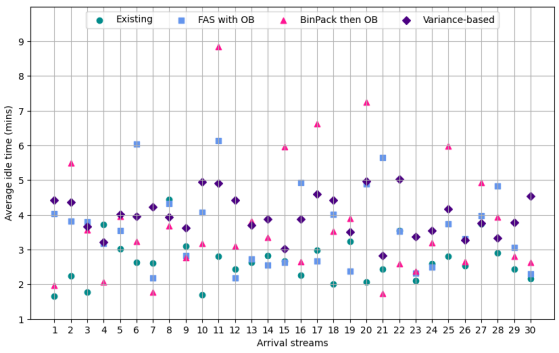

Figure 4: Average idle times (mins) across 30 scenarios for 13-11-6 instance

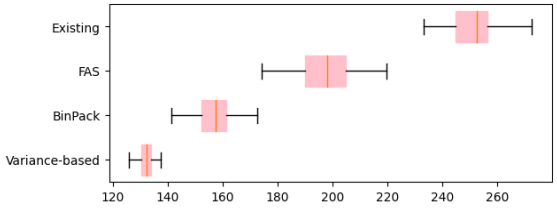

Figure 5: Average wait times (mins) across 30 scenarios for 13-11-6 instance

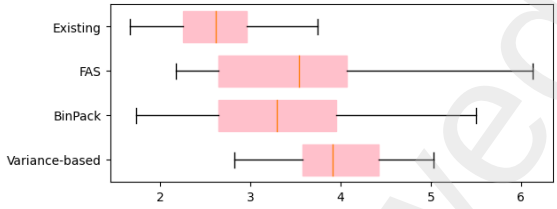

Figure 6: Average idle times (mins) across 30 scenarios for 13-11-6 instance

The other constraints of the model are:

$$\sum_{s=1}^{S} n_{ps} = N_p \qquad \forall p \in P \tag{3}$$

$$\mathbb{P}\left(U_s + O_{s-1} \leq L\right) \leq \epsilon \qquad \forall s \in \{1, \ldots, S\}. \tag{4}$$

Constraints (3) ensures that all patients are assigned a slot. Constraints (4) are the probabilistic constraints limiting the probability of the machine being idle in each slot. The objective is to minimize the sum of expected overflow in each slot to minimize patient wait in the subsequent slots, that is:

$$\min \quad \sum_{s=1}^{S} \mathbb{E}(O_s). \tag{5}$$

In order to write the constraints, in terms of decision variables n_{ps}, we consider the distribution of slot utilization U_s. For each slot s, U_s follows a normal distribution, as it represents the sum of independent, normally distributed service times of patients. Leveraging this property, we can reformulate Constraint (4) for Slot 1 in terms of the decision variables and statistical parameters (Schneider et al. (2020)) as:

$$\Phi\left(\frac{L - \mu_{U_1}}{\sigma_{U_1}}\right) \leq \epsilon, \tag{6}$$

where Φ is the CDF of standard normal distribution. The mean and variance of the utilization of the first slot U_1 are:

$$\mu_{U_1} = \sum_p \mu_p n_{p1} \quad \text{and} \quad \sigma^2_{U_1} = \sum_p \sigma_p^2 n_{p1}.$$

Substituting the latter two expressions in the first slot constraint (6) and taking cdf inverse gives:

$$\sum_{p \in P} \mu_p n_{p1} + \Omega \sqrt{\sum_{p \in P} \sigma_p^2 n_{p1}} \geq L, \tag{7}$$

where $\Omega = \Phi^{-1}(\epsilon)$. For subsequent slots, $s > 1$, the presence of the overflow variable O_s in the probabilistic utilization constraints (4) presents a challenge. O_s, defined using a max$(\cdot)$ operator, lacks a closed-form distribution. However, approximating O_s as $\bar{O}_s$, where

$$\bar{O}_s := U_s + \bar{O}_{s-1} - L \quad \forall s \in \{1, \ldots, S\} \tag{8}$$
$$\text{and} \quad \bar{O}_0 := 0,$$

we can write constraints (4) as:

$$\sum_{k=2}^{s} \sum_p \mu_p n_{ps} + \Omega \sqrt{\sum_{k=2}^{s} \sum_p \sigma_p^2 n_{ps}} \geq s \times L \quad \forall s \in \{2, \ldots, S\}. \tag{9}$$

The square root term in the above constraints (7) and (9) can be reformulated using auxilliary variables $\alpha_s, \forall s \in \{1, \ldots, S\}$ as follows:

$$\sum_{k=1}^{s} \sum_{p \in P} \mu_p n_{pk} + \Omega \alpha_s \geq s \times L \qquad \forall s \in \{1, \ldots, S\} \tag{10}$$

$$\sum_{k=1}^{s} \sum_p \sigma_p^2 n_{ps} \leq \alpha_s^2 \qquad \forall s \in \{1, \ldots, S\} \tag{11}$$

$$\alpha_s \geq 0 \qquad \forall s \in \{1, \ldots, S\}. \tag{12}$$

Constraints (10)-(12) are equivalent to Constraints (9) when Ω is negative, i.e., ϵ is less than 0.5. The objective function can also be rewritten using the definition of $\bar{O}_s$ (8). The constraints for all the slots (7)-(12) can be combined and the complete approximate model is as follows.

$$\begin{aligned}
\min \quad & \sum_s \left(\sum_{k=1}^{s} \sum_{p \in P} \mu_p n_{pk} - s \times L \right) \\
\text{s.t.} \quad & \sum_s n_{ps} = N_p && \forall p \in P \\
& \sum_{k=1}^{s} \sum_{p \in P} \mu_p n_{pk} + \Omega \alpha_s \geq s \times L && \forall s \in \{1, \ldots, S\} \\
& \sum_{k=1}^{s} \sum_{p \in P} \sigma_p^2 n_{ps} \leq \alpha_s^2 && \forall s \in \{1, \ldots, S\} \\
& n_{ps} \in \mathbb{Z}, \quad \alpha_s \geq 0 && \forall p \in P; s \in \{1, \ldots, S\}.
\end{aligned} \tag{MIQCP}$$

The above formulation is a non-convex Mixed-Integer Quadratically Constrained Program (MIQCP) due to the presence of nonconvex quadratic constraints (11).

5.2. Exact Vs Approximate model

We refer the model with O_s variables as an exact model and denote its feasible region by G. Similarly, the model with $\bar{O}_s$ variables is an approximate model, with feasible region R. The

two sets are defined as follows:

$$G = \begin{cases} n_{ps} \in \mathbb{Z}: & \sum_s n_{ps} = N_p, & \forall p \in P \\ & \mathbb{P}(U_s + O_{s-1} \le L) \le \epsilon & \forall s \in \{1, \dots, S\} \end{cases} \tag{13}$$

$$R = \begin{cases} n_{ps} \in \mathbb{Z}: & \sum_s n_{ps} = N_p, & \forall p \in P \\ & \mathbb{P}\left(U_s + \bar{O}_{s-1} \le L\right) \le \epsilon & \forall s \in \{1, \dots, S\} \end{cases} \tag{14}$$

Claim 1. *Feasible region of the approximate model is a subset of that of the exact model, i.e., $R \subseteq G$.*

Proof. The first set of constraints is the same in both sets. Since, $\mathbb{P}(U_s+O_{s-1} \le L) \ge \mathbb{P}(U_s+\bar{O}_{s-1} \le L)$, from the definition of O_s (2) and $\bar{O}_s$ (8), this implies that any point that satisfies R also satisfies G, therefore, $R \subseteq G$. □

Claim 2. *The objective function of the exact model evaluated at the feasible solution of the approximate model provides the upper bound to the optimal objective function value of the exact model.*

Proof. Consider the objective function for both the models (5),

$$f_G = \sum_s \mathbb{E}[U_s + O_{s-1} - L]^+ \tag{15}$$

$$f_R = \sum_s \mathbb{E}[U_s + \bar{O}_{s-1} - L] \tag{16}$$

Now, for any random variable Y,

$$\begin{aligned} \mathbb{E}[Y] &= \mathbb{E}[Y^+] - \mathbb{E}[Y^-] \\ &\le \mathbb{E}[Y^+] \end{aligned}$$

Therefore, it implies, $f_R \le f_G$. Now, let $\hat{n}$ be the optimal solution of the approximate model and n^* be the optimal solution of the exact model. Then from above, $f_R(\hat{n}) \le f_G(\hat{n})$, and for a minimization problem, $f_G(n^*) \le f_G(\hat{n})$. Unfortunately, evaluating $f_G(\hat{n})$ exactly is challenging, as there is no closed-form expression available for O_s, and consequently, for f_G, even under the assumption of a normal distribution. We, therefore, resort to numerically compute $f_G(\hat{n})$, and verify the claim using simulation (Section 4.1). Assuming our simulation experiments are accurate, the approximate model solution evaluated at the exact objective function $f_G(\hat{n})$ provides an upper bound to the objective function of the exact model $f_G(n^*)$. □

The approximation of O_s is likely to work well with the smaller value of ϵ. However, it might lead to overestimation in scenarios with higher idleness probabilities. This is when ϵ increases, the likelihood of O_s being zero (no overflow) also rises, resulting in a significant gap in O_s, and $\bar{O}_s$.

We present another formulation of scheduling patients-wise in Appendix A as opposed to the type-wise approach described in Section 5.1. It is formulated using the set of binary variables x_{is}, such that $x_{is} = 1$ if patient i is assigned to slot s, and zero otherwise. The obtained formulation is a Second-Order Conic Program (SOCP) as described in the Appendix A. Although the formulation being a SOCP ensures convexity, it encounters computational challenges arising from symmetry induced by binary variables. For example, if two patients are of the same type, scheduling the first in a specific slot is equivalent to scheduling the second in the same slot. This takes a prolonged time for solvers. On the other hand, (MIQCP) above uses a few general integers and is computationally solvable with a non-convex solver. We perform our experiments using (MIQCP) formulation for comparison with online policies.

6. Computational study of offline model

In this section, we compare the best online policy among the three proposed policies with the offline policy. The offline model is implemented in Pyomo (Bynum et al., 2021) and Gurobi 11.0. (Gurobi, 2023) is used as a non-convex MIQCP solver. We first present a few observations from offline policy and later compare it with variance-based online slot allocation policy.

6.1. Observations from offline policy

We consider the probability of machine idleness within a slot to vary from $0.05 - 0.45$ and solve (MIQCP) for a few instances described in Section 4.1. At $\epsilon = 0.5$, the model simplifies to a deterministic counterpart of (MIQCP) disregarding uncertainties in service times. Smaller values of ϵ signify a higher level of robustness in the model, reflecting a more conservative approach to handling uncertainty. After obtaining an optimal assignment, we simulated the patient-slot assignments obtained from the model solution to incorporate the randomness in service times as depicted in Figure 2. Table 9 reports the objective function value obtained from both the optimization model ('Opt') and the simulation model ('Sim'), corresponding to increasing levels of robustness. The blanks in the table indicate instances where the model is infeasible for a given ϵ.

Table 9 highlights the difference in simulated objective value and the optimal objective value for a given ϵ. The difference is because the simulation model captures the actual overflows O_s, unlike $\bar{O}_s$ considered in the approximated model (MIQCP), this verifies Claim 5.2. Further, the difference increases with the increase in value of ϵ, from 9% at $\epsilon = 0.1$ to 57% at $\epsilon = 0.45$. As the permissible probability of idleness in each slot increases, the model tends to schedule fewer patients per slot, consequently increasing the likelihood of serving all the patients scheduled within the slot. Hence, the chance of the quantity $U_s + O_{s-1}$ being less than 120 increases with an increase in ϵ. Therefore, approximation of the overflow $\bar{O}_s$ becomes less accurate as robustness decreases (higher values of ϵ).

The optimal values of slot overflows can be derived by evaluating constraints on the obtained solution. However, the other metrics, such as machine idle time and patient waiting time are not explicitly provided by the optimization model. Conversely, these metrics can be obtained from the simulation model. Table 10 and 11 present the waiting and idle time values obtained from the simulation model, respectively. As expected, the total idle time throughout the day decreases with increasing robustness, while the average waiting time per patient increases.

Table 12 reports the overtime values obtained from both the optimization and simulation model, where an overtime of 0 indicates all patients were served within the 22-hour timeframe. Note that, while optimal overtime is merely the difference in the total expected service time of all scheduled patients and the total machine available time, simulated overtime values are more realistic. Furthermore, while waiting and idle time values may not vary significantly across instances, overtime is directly proportional to the number of patients scheduled for the day. This relationship is intuitive, as the time required to serve patients increases with their number.

6.2. *Comparison of online variance-based policy against offline policy*

Table 13 shows the average performance metrics obtained from variance-based online policy and simulated offline policy for each instance. It is evident that the variance-based policy performs as well as the offline policy considering the three measures. The difference in values is due to the absence of prior information regarding the exact number of patients of all types. Consequently, the increased values of the measures reflect the price of information. The percentage difference in the objective function of variance-based policy and simulated offline policy is 16% (Figure 7). This indicates that the approximated model provides a decent quality upper bound to the exact model. Further, while the variance-based policy exhibits slightly higher waiting times, with an average percentage difference of 17%, it compensates with a reduction in both idle time and overtime. Specifically, the variance-based policy achieves a reduction of 32% in idle time and 6% in overtime values, indicating more efficient resource utilization and improved appointment scheduling management.

7. Conclusions

In this paper, we study the problem of MRI scan scheduling, aiming to find an optimal balance between machine idle time and patient waiting time while minimizing machine and staff overtime. We present three policies to assign patients to time slots in a day, as and when they arrive to book appointments with a requested scan type. Among the three policies, the Variance-based policy performs better in terms of all metrics considered. By considering the variance in service times and scheduling high-variance patients in later slots, this policy effectively reduces average patient waiting times throughout the day while maintaining efficient machine utilization. Although implementing a variance-based policy in practice is slightly more difficult as compared to others, it is possible to design an appointment system that schedules high-variance patients in later slots and does probability computations before every assignment.

The other two policies presented consider overbooking initial slots to ensure a buffer against variability in service times. The buffer time (b), and the number of slots to be overbooked (k) can be determined based on the specific requirements of the system. The results obtained from the BinPack then OB policy are intuitive, as it only allots a patient to a slot if it can be completely accommodated in it. In contrast, the FAS policy allows not only overbooking in initial slots but also inherent overbooking in later slots by allotting patients even if there is only one minute of time left in the slot. This results in either empty or underutilization of later slots with no patients scheduled in them. This leads to increased patient wait times as patients are called about two slots ahead of when they can be served. The implementation of these policies requires continued monitoring of the slot utilization. While the FAS policy may seem easier to implement, its long waiting times can lead to overcrowding and increased patient stress.

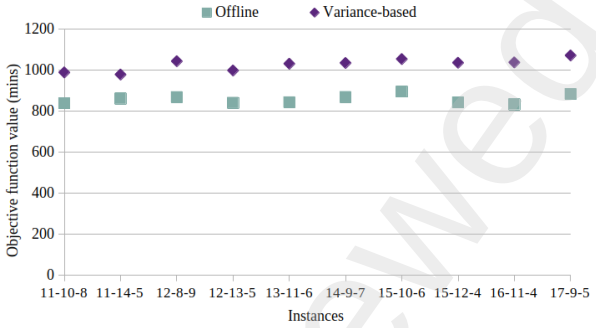

Figure 7: Objective function value from simulated offline and variance-based online policy for all instances

In addition to the aforementioned policies, we propose a chance-constrained model to determine the optimal number of each type of patient to be assigned to slots throughout the day, assuming arrival streams are known a priori. While we do not solve the exact model, the approximated model provides a good estimate for it. The variance-based policy comes quite close to the approximate offline model and hence is recommended. Solving the exact offline model or even developing a relaxation for the exact model to provide a good lower bound estimate seems to be interesting lines for future work. Practically, the question of deciding the day of the scan is also quite important, and a separate study on this question is also planned in the future.

Appendix A. Another formulation for offline scheduling

An alternative formulation for the problem described in Section 5 is assigning individual patients to time slots. In contrast to the previous type-wise approach, this necessitates information of type p of each patient $i \in I$. It can be formulated using binary variables x_{is}, such that $x_{is} = 1$ if patient i is assigned to slot s, and zero otherwise. Here, the utilization of slot can be expressed in terms of random service times of patients (X_i) scheduled in slot s and decision variable x_{is} unlike MIQCP. Further, the overflow of slot s can be written as follows:

$$O_s := \left(\sum_i X_i x_{is} + O_{s-1} - L \right)^+ \qquad \forall s \in \{1, \ldots, S\}$$

and $\quad O_0 := 0.$

Instance		0.45	0.4	0.35	0.3	0.25	0.2	0.15	0.1	0.05
11-10-8	Opt	198.22	259.43	322.67	399.22	451.19	547.06	636.85	773.65	-
	Sim	469.96	486.54	541.78	578.51	589.13	672.33	742.71	839.13	-
12-13-5	Opt	196.4	250.74	321.49	380.37	433.99	535.72	626.15	746.97	-
	Sim	463.69	492.96	527.36	566.42	601.25	658.72	692.18	841.09	-
13-11-6	Opt	202.63	260.72	327.07	391.64	449.02	539.48	633.67	769.52	-
	Sim	483.56	506.17	523.12	571.68	616.05	658.16	740.67	843.62	-
14-9-7	Opt	206.38	267.59	330.83	409.78	465.81	562.02	656.21	794.23	-
	Sim	480.72	509.01	527.38	577.05	624.30	661.36	730.01	869.36	-
15-10-6	Opt	231.00	292.20	355.44	428.03	484.97	579.83	675.03	797.82	988.153
	Sim	506.48	540.54	551.62	606.34	638.70	698.13	748.00	895.99	1071.149

Table 9: Optimal and simulated objective function value (in mins) from offline model for different values of ϵ

Instance	0.45	0.4	0.35	0.3	0.25	0.2	0.15	0.1	0.05
11-10-8	77.16	78.05	84.13	86.53	88.88	96.14	104.52	112.04	-
12-13-5	77.72	79.79	83.77	87.03	89.78	97.48	101.65	110.61	-
13-11-6	80.22	81.14	86.12	91.19	90.61	96.81	105.46	109.81	-
14-9-7	79.28	81.61	84.86	89.21	91.46	96.77	104.45	114.57	-
15-10-6	84.78	86.40	92.55	90.53	94.58	107.48	103.95	113.13	133.48

Table 10: Waiting time (mins) from offline model for different values of ϵ

Instance	0.45	0.4	0.35	0.3	0.25	0.2	0.15	0.1	0.05
11-10-8	33.672	29.14	25.33	17.9	17.57	13.3	9.73	6.45	-
12-13-5	29.78	26.68	22.87	17.59	16.19	12.25	9.68	5.93	-
13-11-6	31.08	26.30	24.47	19.28	14.32	13.57	9.45	6.07	-
14-9-7	31.98	28.69	25.21	21.27	16.17	13.75	9.41	6.74	-
15-10-6	32.76	32.17	25.09	19.18	16.53	11.72	9.30	5.43	2.58

Table 11: Idle time (mins) from offline model for different values of ϵ

Instance	Optimal	0.45	0.4	0.35	0.3	0.25	0.2	0.15	0.1	0.05
11-10-8	0	26.68	22.40	18.77	12.09	10.97	6.34	3.95	0	-
12-13-5	0	27.01	22.36	19.37	15.03	11.86	8.32	4.16	2.01	-
13-11-6	0	33.06	28.40	25.27	19.10	16.15	12.37	8.56	4.09	-
14-9-7	0	37.12	32.63	29.47	24.31	20.72	17.61	13.34	12.38	-
15-10-6	18.01	62.91	58.32	55.74	48.99	43.93	42.10	38.74	35.97	32.78

Table 12: Over time (mins) from offline model for different values of ϵ

Instance	Policies	Avg wait time	Avg idle time	Avg over time	Avg obj function
11-10-8	Sim offline	112.04	6.45	0	839.13
	Variance-based	126.93	4.67	0	989.35
11-14-5	Sim offline	111.97	5.78	6.75	861.79
	Variance-based	127.76	4.05	7.35	979.08
12-8-9	Sim offline	115.53	6.03	2.49	868.75
	Variance-based	130.61	4.16	2.91	1044.33
12-13-5	Sim offline	110.61	5.93	2.01	841.09
	Variance-based	129.83	4.21	0.74	998.96
13-11-6	Sim offline	109.81	6.07	4.09	843.62
	Variance-based	132.19	3.97	4.76	1031.93
14-9-7	Sim offline	114.57	6.74	12.38	869.36
	Variance-based	132.32	4.09	8.16	1035.50
15-10-6	Sim offline	113.13	5.43	35.97	895.99
	Variance-based	134.20	3.53	33.63	1054.65
15-12-4	Sim offline	111.42	5.03	10.46	842.76
	Variance-based	133.22	3.48	10.06	1036.64
16-11-4	Sim offline	110.89	5.41	0	834.07
	Variance-based	133.39	3.82	3.30	1038.29
17-9-5	Sim offline	115.66	5.67	8.50	883.96
	Variance-based	136.79	3.74	6.75	1072.27

Table 13: Average measures (in mins) from simulating offline and variance-based online policy for all instances

The other constraints of the model are:

$$\sum_{s\in S} x_{is} = 1 \qquad \forall i \in I \qquad \text{(A.1)}$$

$$\mathbb{P}\left(\sum_{i\in I} X_i x_{is} + O_{s-1} \le L\right) \le \epsilon \qquad \forall s \in \{1,\dots,S\}. \qquad \text{(A.2)}$$

The objective function remains the same as in MIQCP. Using the similar approximation of O_s (8), and standardizing the constraints, we get the following second-order conic program (Atamtürk and Bhardwaj (2018)):

$$\begin{aligned}
\min \quad & \sum_{s}\left(\sum_{k=1}^{s}\sum_{i\in I}\mu_i x_{ik} - s\times L\right) \\
\text{s.t.} \quad & \sum_{s} x_{is} = 1 && \forall i \in I \\
& \sum_{k=1}^{s}\sum_{i\in I}\mu_i x_{ik} + \Omega\alpha_s \ge s\times L && \forall s \in \{1,\dots,S\} \\
& \sum_{k=1}^{s}\sum_{i\in I}(\sigma_i x_{ik})^2 \le \alpha_s^2 && \forall s \in \{1,\dots,S\} \\
& x_{is} \in \{0,1\}, \quad \alpha_s \ge 0, && \forall i \in I; s \in \{1,\dots,S\},
\end{aligned} \qquad \text{(SOCP)}$$

where $\Omega = \phi^{-1}(\epsilon)$ and ϕ is the standard normal cdf. Constraints (A.1) ensures that each patient is scheduled in at most one slot. Constraints (A.2) are analogous to (4), expressed in terms of x_{is}. The formulations MIQCP and SOCP are equivalent as shown in the following. Let x^*_{is} be a feasible solution to SOCP, we can construct a feasible solution n^*_{ps} corresponding to MIQCP as follows:

$$n^*_{ps} = \sum_{i:i=p} x^*_{is}, \qquad \text{(A.3)}$$

as n_{ps} represents the total number of patients of type p in slot s. The first constraint in both models ensures that each patient is scheduled in at most one slot. The equivalence of these constraints is evident from A.3. The second constraints in both models are equivalent as the first term $\sum_{k=1}^{s}\sum_{i\in I}\mu_i x^*_{is}$ in SOCP transforms to $\sum_{k=1}^{s}\sum_{p\in P}\mu_p n^*_{ps}$ in MIQCP, reflecting the aggregation of patients by type. The third constraints are equivalent because $x^2_{ik} = x_{ik}, x \in \mathbb{B}$. Conversely, a solution n^*_{ps} feasible to MIQCP can be transformed to x^*_{is} feasible to SOCP. Since the objective functions of both models are identical, any optimal solution to one model will also be optimal for the other.

References

D. Gupta, B. Denton, Appointment scheduling in health care: Challenges and opportunities, IIE transactions 40 (2008) 800–819.

T. Cayirli, E. Veral, Outpatient scheduling in health care: a review of literature, Production and operations management 12 (2003) 519–549.

T. S. Pham, A. Legrain, P. De Causmaecker, L.-M. Rousseau, A prediction-based approach for online dynamic appointment scheduling: A case study in radiotherapy treatment, INFORMS Journal on Computing (2023).

D. Gupta, B. Denton, Appointment scheduling in health care: Challenges and opportunities, IIE transactions 40 (2008) 800–819.

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

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

M. B. White, M. Pike, Appointment systems in out-patients' clinics and the effect of patients' unpunctuality, Medical Care (1964) 133–145.

A. Soriano, Comparison of two scheduling systems, Operations Research 14 (1966) 388–397.
T. F. Cox, J. P. Birchall, H. Wong, Optimising the queuing system for an ear, nose and throat outpatient clinic, Journal of Applied Statistics 12 (1985) 113–126.
P. Bhattacharjee, P. K. Ray, Simulation modelling and analysis of appointment system performance for multiple classes of patients in a hospital: a case study, Operations Research for Health Care 8 (2016) 71–84.
L. V. Green, S. Savin, B. Wang, Managing patient service in a diagnostic medical facility, Operations Research 54 (2006) 11–25.
A. Sauré, M. L. Puterman, The appointment scheduling game, INFORMS Transactions on Education 14 (2014) 73–85.
P. Cappanera, F. Visintin, C. Banditori, D. Di Feo, Evaluating the long-term effects of appointment scheduling policies in a magnetic resonance imaging setting, Flexible Services and Manufacturing Journal 31 (2019) 212–254.
S. Benjaafar, D. Chen, R. Wang, Z. Yan, Appointment scheduling under a service-level constraint, Manufacturing & Service Operations Management 25 (2023) 70–87.
B. Addis, G. Carello, E. Tànfani, A robust optimization approach for the advanced scheduling problem with uncertain surgery duration in operating room planning-an extended analysis (2014).
D. Bertsimas, M. Sim, The price of robustness, Operations research 52 (2004) 35–53.
A. Ben-Tal, A. Nemirovski, Robust solutions of linear programming problems contaminated with uncertain data, Mathematical programming 88 (2000) 411–424.
J. Patrick, M. L. Puterman, M. Queyranne, Dynamic multipriority patient scheduling for a diagnostic resource, Operations research 56 (2008) 1507–1525.
D. Smedira, D. Shmoys, Scheduling appointments online: the power of deferred decision-making, in: International Workshop on Approximation and Online Algorithms, Springer, 2022, pp. 82–115.
Y. Chen, Y.-H. Kuo, P. Fan, H. Balasubramanian, Appointment overbooking with different time slot structures, Computers & Industrial Engineering 124 (2018) 237–248.
N. Alon, B. Awerbuch, Y. Azar, The online set cover problem, in: Proceedings of the thirty-fifth annual ACM symposium on Theory of computing, 2003, pp. 100–105.
J. N. Gupta, J. C. Ho, A new heuristic algorithm for the one-dimensional bin-packing problem, Production planning & control 10 (1999) 598–603.
L. W. Robinson, R. R. Chen, Scheduling doctors' appointments: optimal and empirically-based heuristic policies, Iie Transactions 35 (2003) 295–307.
A. T. Schneider, J. T. van Essen, M. Carlier, E. W. Hans, Scheduling surgery groups considering multiple downstream resources, European journal of operational research 282 (2020) 741–752.
M. L. Bynum, G. A. Hackebeil, W. E. Hart, C. D. Laird, B. L. Nicholson, J. D. Siirola, J.-P. Watson, D. L. Woodruff, Pyomo–optimization modeling in python, volume 67, third ed., Springer Science & Business Media, 2021.
Gurobi, Gurobi Optimizer Reference Manual, 2023. URL: https://www.gurobi.com.
A. Atamtürk, A. Bhardwaj, Network design with probabilistic capacities, Networks 71 (2018) 16–30.