Noname manuscript No.
(will be inserted by the editor)

Optimal Override Policy for Chemotherapy Scheduling Template via Mixed-Integer Linear Programming

Yu-Li Huang · Ishtiak Sikder · Guanglin Xu

Received: date / Accepted: date

Abstract Owing to treatment complexity in chemotherapy administration, nurses are usually required at the beginning, end, and at certain times during the treatment period to ensure high-quality infusion. It is, thus, critical for an outpatient chemotherapy unit to design a scheduling template that can effectively match nursing resources with treatment requirements. The template contains time slots of different lengths and thus allows schedulers to place patients into these slots according to the provider's order. As the template is often used over a period of several months, there usually exists a mismatch between the daily patient mix and the fixed structure of the given template. Hence, override policies must be employed to adjust to demand. However, these policies are often manually performed by schedulers. To this end, we propose a mixed-integer linear programming model to develop systematic optimal override policies in place of the manual process to improve template utilization while maintaining the same template design. Numerical experiments based on real-life data from a chemotherapy unit have been conducted to demonstrate the effectiveness of the proposed model.

Keywords Chemotherapy, scheduling template, override policy, mixed-integer linear programming

Yu-Li Huang
Robert D. and Patricia E. Kern Center for the Science of Health Care Delivery, Mayo Clinic, Rochester, MN, USA
E-mail: huang.yuli@mayo.edu

Ishtiak Sikder
Department of Systems Engineering and Engineering Management, University of North Carolina at Charlotte, Charlotte, NC, USA
E-mail: isikder@uncc.edu

Guanglin Xu, Corresponding Author
Department of Systems Engineering and Engineering Management, University of North Carolina at Charlotte, Charlotte, NC, USA
E-mail: guanglin.xu@uncc.edu

1 Introduction

The safe administration of chemotherapy is paramount and therefore the optimal management of workflows in the chemotherapy units has become a major focus for any practical settings [11]. In the past decades, chemotherapy administration has gradually shifted from the inpatient setting to the outpatient setting due to the advanced development in medical delivery methods, drug and prescription innovation, and side effect management [21]. In general, outpatient service involves patients coming into the hospital to obtain essential health care and leaving within the day after receiving the service. The standard method of scheduling outpatient appointments is to apply a policy that accounts for appointment block size and appointment intervals; see e.g., [3].

Appointment scheduling has been an extensively studied research area in operations research and management science [1], [8], [17], [18], since it has applications in a wide range of fields such as outpatient scheduling [7], [10], [14], [16], surgery planning [5], call-center staffing [9], and cloud computing server operations [20]. It often aims to efficiently and effectively allocate scarce resources to satisfy requirements against some physical and/or economic constraints. For instance, the primary operational objective of many appointment systems is to design scheduling templates for appointments to optimize the overall benefit or costs of the system. Generally, there are two typical types of appointment systems: single- and multi-server systems; see [13], [22].

One of the main challenges to develop a flexible appointment scheduling system regardless appointment system types is the process variability. There are several sources of variability. First, although the allotted time for a patient in the scheduling template is fixed, the actual duration of the appointment is typically unknown. For instance, the treatment duration of a patient may depend on his health state, which has to be observed on the fly. There is also uncertainty due to the requirement of urgent appointments, walk-ins, and the occurrence of patients not showing for appointments. Random no-shows cause poor resource utilization and unanticipated loss of revenue for health care providers, which may, in turn, compromise service quality [2]. Furthermore, the varying patient mix also causes dramatic modeling challenges for the scheduling systems especially in chemotherapy treatment facility. In fact, tackling this type of variability is the main focus of this paper. Stochastic programming [15] is generally utilized as a solution method to solve appointment scheduling problems with uncertainties; see e.g., [6], [19]. Moreover, simulation is another primary approach to tackle uncertainty in appointment scheduling problems; see e.g., [4].

In practice, treatment plan schedules for chemotherapy may vary depending on the type of cancer, the associated treatment regimen and its goal, as well as the patients' state of health. An oncologist decides the choice of a particular regimen, but modifications in drug dosage and schedules are often necessary due to the variability in the health status of a patient. Therefore, a scheduling template must be determined to consist of time slots that can accommodate a patient mix requiring various treatment duration. In other words, the time slots in the template should be of different lengths based on the distribution of the patient mix. However, a fixed scheduling template may not be able to accommodate varying patient mix on a daily basis. To this end, override policies should be employed to modify the existing time slots to accommodate patients, who cannot be served by time slots with exact lengths.

In this paper, we develop a mixed-integer linear programming (MILP) model to develop optimal override policies. In particular, assuming that a fixed scheduling template and a patient mix are given, the MILP model selects time slots to treat patients via pre-determined override rules. Our model can provide schedulers with flexibility in the preference of the override rules. We verify the MILP model over a real-life data set collected from a chemotherapy unit from The Mayo Clinic.

1.1 Our contributions

This study introduces a novel optimization scheme to generate optimal override policies that may be applied to assign appointment template slots to accommodate patients requiring different treatment lengths. Particularly, a mixed-integer linear programming model for determining the optimal override policies has been proposed and solved by using an off-the-shelf solver. Numerical experiments on real data from a chemotherapy unit are conducted to illustrate the effectiveness of the proposed approach. To the best of the authors' knowledge, this is the first study to utilize mixed-integer linear programming techniques to derive optimal policies in the context of chemotherapy appointment scheduling.

1.2 Organization of the paper

The remaining sections are organized as follows. In Section 2, we discuss the chemotherapy appointment scheduling template and the override rules, and propose a mixed-integer linear programming (MILP) model to optimally apply the override policies. In Section 3, numerical experiments demonstrating the effectiveness of the proposed approach on real-life data from a chemotherapy unit. Finally, concluding remarks and future extensions for optimization-based override policy problems are given in Section 4.

2 Problem Formulation

This section describes the override policies and then proposes a corresponding mixed-integer linear programming (MILP) model that can be solved using off-the-shelf solvers such as Gurobi or CPLEX. In what follows, it is assumed that a given scheduling template is deployed at the chemotherapy unit and that the patient volume of different types are available when the scheduler makes appointment scheduling.

2.1 Override policies

Chemotherapy units often utilize a template over a period of several months or even years. To ensure that a template accounts for nursing resource needs during patient treatments, a fixed template is often proposed. However, the distribution of the different types of incoming patients generally varies on a daily basis. This phenomenon leads to a mismatch between the time blocks designed by the template and patients who require different lengths of treatment duration. To handle this mismatch, the scheduler often manually follows override policies to modify the appointment template to accommodate patients in the current practice at Mayo Clinic [12]. By the clinic requirement, the override policy should incorporate the following three rules:

- *Policy 1*: Place a patient in a longer appointment slot when the designated slot is no longer available. For example, a patient requiring a 2-hour slot can be scheduled in a 3-hour slot or any slots longer than two hours.
- *Policy 2*: Combine two subsequent slots to create a slot that is equal to or greater than the amount of time required for the treatment. For example, if a chair/ bed has a 1-hour slot from 7:45 AM to 8:45 AM and a 2-hour slot from 8:45 AM to 10:45 AM, then these can be combined to serve a patient requiring a 3-hour slot from 7:45 AM to 10:45 AM.
- *Policy 3*: Break a longer appointment slot into two shorter slots. For example, a 6-hour slot can be used for two 3-hour slots or a 2-hour slot and a 4-hour slot. It can also be used for two 2-hour slots or a 2-hour slot and 3-hour slot. It is not preferred to break a longer slot into more than two shorter slots as the nursing resource is not fully considered. For a 6-hour slot to be used for assigning two 3-hour slots, the nursing resource is not planned at the end of the first 3-hour patient slot and the beginning of the second 3-hour patient slot. In this example, resource constraint violations are increased by two.

The current manual override process at Mayo Clinic is highly driven by patient preferences. Schedulers sometimes manipulate the template and deviate from the override policies by making allocations to fit as many patients as possible in a single template block; even if there are other times available in a day. Thus, policy 3 is often used to accommodate patients, especially those who require a shorter duration of treatment. For override simplicity, schedulers prefer policy 1 over policy 3 and policy 3 over policy 2 since policy 2 requires finding the right slots to combine, which could be time-consuming. In the current state, schedulers are not restricted to deviate from the override policies. For example, schedulers may break a template slot to treat three patients, which is a violation of policy 3. Schedulers may also combine three template slots to treat a patient, which is a violation of policy 2. Hence, a systematic approach is paramount.

2.2 Mixed-integer linear programming formulation

In this section, we aim to optimize the use of the override policy by developing a mixed-integer linear programming (MILP) model. More specifically, given a particular mix of patient arrival including the total number of patients and the distribution of over all patient types; the MILP model searches for the optimal patient assignment.

To simplify the formulation, we define two necessary index sets which are specified for ease of exposition. However, the sets can be specified by different requirements.

Sets:

- Set of time slots $\mathcal{T} := \{1, 2, \ldots, 44\}$.
- Set of patient types $\mathcal{I} := \{30, 60, 120, 180, 240, 300, 360\}$.

Parameters:

- p_i: Number of patients of type i for all $i \in \mathcal{I}$.
- $c_{i,t}$: Number of type i appointments scheduled to start at time t for all $i \in \mathcal{I}$ and for all $t \in \mathcal{T}$. Note that the parameters c_{it} are determined by the given scheduling template.
- d: The length of the time slot. In the paper, we consider the case $d = 15$.
- λ_i $i = 1, \ldots, 5$: Weight parameters that are used in the objective function.

Variables:

Now we define the following binary variables to represent an assignment where a type i' appointment slot is used to treat a patient of type i $(i \leq i')$.

- $x_{i,i',j',t}$: Binary variable, equals 1 if the j'th appointment slot for a patient of type i' at time t is used to accommodate a patient of type i $(i \leq i')$; and 0 otherwise.

We then define the following binary variables to represent an assignment where an appointment slot for type i' and an appointment slot for type i'' are combined to treat a patient of type i $(i \leq i' + i'')$.

- $y_{i,i',i'',j',j'',t}$: Binary variable, equals 1 if the j'th appointment slot for a patient of type i' at time t and the j''th appointment slot for a patient of type i'' at time $t+i'/d$ are combined to treat a patient of type i $(i \leq i'+i'')$; and 0 otherwise.
- $z_{i,i',j',t}$: Binary variable, equals 1 if the j'th appointment slot for a patient of type i' at time t is used to start the treatment of a patient of type i $(i > i')$; and 0 otherwise.
- $w_{i,i',j',t}$: Binary variable, equals 1 if the j'th appointment slot for a patient of type i' at time t is used to complete the treatment of a patient of type i $(i > i')$; and 0 otherwise.

Next, we define the following binary variables to represent an assignment where an appointment slot of type i' is split into two smaller slots of types i and i'' which can be used to treat two patients.

- $v_{i,i'',i',t,j'}$: Binary variable, equals 1 if the j'th appointment slot for a patient of type i' at time t is divided into two blocks to serve a patient of type i and a patient of type i'' where $(i + i'' \leq i')$; and 0 otherwise.

Finally, we define the following auxiliary variables to help model necessary equality constraints.

- $u_{i,t}$: Integer variable, represents the number of type i patients who can be treated at time t.
- q_i: Integer variable, represents the number of type i patients who have not been served.

With the sets, parameters, and variables defined above, we are ready to outline the following constraints of the MILP model.

Constraints:

The following constraints enforce that each time slot in the appointment template can be used for a maximum of one task (e.g., equal assignment, larger assignment, combination, and breaking). More specifically, for all $i' \in \mathcal{I}$, for all $t \in \mathcal{T}$, and for all $j \in [c_{i',t}] := \{1, \ldots, c_{i',t}\}$, we have

$$\sum_{i:i\leq i'} x_{i,i',j',t} + \sum_{i:i>i'} z_{i,i',j',t} + \sum_{i:i>i'} w_{i,i',j',t} + \sum_{(i,i'')\in\mathcal{E}} \frac{1}{2} v_{i,i'',i',t,j'} \leq 1 \qquad (1)$$

where $\sum_{i:i\leq i'}$ denotes the shorthand of notation $\sum_{i\in\mathcal{I}:i\leq i'}$, and $\mathcal{E}$ is defined as follows:

$$\mathcal{E} := \Big\{(i,i'') : i < i', i'' < i', i + i'' \geq i'\Big\}.$$

Note that the set $\mathcal{E}$ is defined to filter out unnecessary variables. For instance, breaking assignments $v_{i,i'',i',j,t'}$ can never happen for the corresponding indices in $\overline{\mathcal{E}}$, which is the complement set of $\mathcal{E}$.

The following constraints are used to ensure the policy to combine two appointment slots to treat a patient.

$$y_{i,i',i'',j',j'',t} \leq z_{i,i',j',t} \quad \forall\,(i,i',i'',j',j'',t) \in \mathcal{E}_y \qquad (2)$$

$$y_{i,i',i'',j',j'',t} \leq w_{i,i'',j'',t+i'} \quad \forall\,(i,i',i'',j',j'',t) \in \mathcal{E}_y \qquad (3)$$

$$y_{i,i',i'',j',j'',t} \geq z_{i,i',j',t} + w_{i,i'',j'',t+i'} - 1 \quad \forall\,(i,i',i'',j',j'',t) \in \mathcal{E}_y \qquad (4)$$

$$\sum_{i'\in\mathcal{I}} \sum_{j'\in[c_{i',t-i'/d}]} y_{i,i',i'',j',j'',t-i'/d} \leq 1 \quad \forall\,(i,i'',j'',t) \in \mathcal{E}_w \qquad (5)$$

$$\sum_{i''\in\mathcal{I}} \sum_{j''\in[c_{i'',t}]} y_{i,i',i'',j',j'',t} \leq 1 \quad \forall\,(i,i',j',t) \in \mathcal{E}_z \qquad (6)$$

where $\mathcal{E}_y, \mathcal{E}_w$, and $\mathcal{E}_z$ are respectively defined as

$$\mathcal{E}_y := \left\{ (i, i', i'', j', j'', t) : \begin{array}{c} i \in \mathcal{I},\, i' \in \mathcal{I},\, i'' \in \mathcal{I} \\ j' \in [c_{i',t}],\, j'' \in [c_{i'',t+i'/d}] \\ i' < i,\, i'' < i,\, i' + i'' \geq i \\ t + (i' + i'')/d \in \mathcal{T} \end{array} \right\},$$

$$\mathcal{E}_w := \left\{ (i, i'', j'', t) : \begin{array}{c} i \in \mathcal{I},\, i'' \in \mathcal{I},\, i > i'' \\ t \in \mathcal{T},\, j'' \in [c_{i'',t}] \end{array} \right\},$$

$$\mathcal{E}_z := \left\{ (i, i', t, j') : \begin{array}{c} i \in \mathcal{I},\, i' \in \mathcal{I},\, i > i' \\ t \in \mathcal{T},\, j' \in [c_{i',t}] \end{array} \right\}.$$

The following constraints ensure the implementation of the breaking policy where a larger appointment slot is used to treat two patients. For all $i \in \mathcal{I}$, for all $t \in T$, and for all $j' \in [c_{i',t}]$, we have

$$\sum_{i' \in \mathcal{I}} \sum_{i'' \in \mathcal{I}} v_{i,i'',i',t,j'} \leq 1. \tag{7}$$

Next, we define the following equations which will help establish the objective function. For all $i \in \mathcal{I}$ and for all $t \in \mathcal{T}$, we define

$$\begin{aligned} u_{it} = &\sum_{i':i' \geq i} \sum_{j' \in [c_{i,t}]} x_{i,i',j',t} + \sum_{(i',i'') \in \mathcal{E}'} \sum_{j' \in [c_{i',t}]} \sum_{j'' \in [c_{i'',t}]} y_{i,i',i'',j',j'',t} \\ &+ \sum_{(i,i'') \in \mathcal{E}} \sum_{j \in [c_{i,t}]} v_{i,i'',i',t,j'}. \end{aligned} \tag{8}$$

For all $i \in \mathcal{I}$, we define

$$q_i = (p_i - \sum_{t \in \mathcal{T}} u_{i,t})_+. \tag{9}$$

Finally, we formulate the mixed-integer linear programming model as follows:

$$\begin{aligned} \min\ & \lambda_1 \sum_{i \in \mathcal{I}} q_i + \lambda_2 \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}} u_{i,t} + \lambda_3 \sum_{i \in \mathcal{I}} \sum_{i':i'>i} \sum_{j' \in [c_{i,t}]} \sum_{t \in \mathcal{T}} x_{i,i',j',t} \\ & + \lambda_4 \sum_{(i,i',i'',j',j'',t) \in \mathcal{E}_y} y_{i,i',i'',j',j'',t} + \lambda_5 \sum_{(i,i',i'',t,j) \in \mathcal{E}_v} v_{i,i'',i',t,j'} \\ \text{s. t.}\ & \text{equations } (1) - (9). \end{aligned} \tag{10}$$

We make two remarks as follows. First, the objective function is used to comprehensively balance the number of unassigned patients and the usage of different policies. Second, the parameters λ_i are used as the priorities of these considerations.

3 Numerical Experiments

In this section, we validate the MILP model by using real-life data collected from a chemotherapy unit at the Mayo Clinic. All the experiments were implemented in Python 3.7 using Gurobi 8.0 as the MILP solver and were performed on a Macintosh OS X Yosemite system with a quad-core 3.20GHz Intel Core i5 CPU and 8 GB RAM.

3.1 Experimental setup

Our experiments were conducted on an actual patient scheduling data set collected from the Andreas Cancer Center, which is situated at Mankato, Minnesota and operated under the Mayo Clinic Health System. Particularly, the data set includes clinical visits from seven different types of patients, which are categorized by the treatment lengths: 30, 60, 120, 180, 240, 300, and 360 minutes. 22-working-day observations were selected in the data set; see Table 1. The chemotherapy unit utilizes a fixed template, which is visualized in the left panel in Figure 1. The template adopts 14 chairs/beds to accommodate chemotherapy patients on each working day between 7:00AM and 5:00PM. Particularly, the template has a total capacity of 61 slots; where 23, 8, 10, 11, 6, 2 and 1 slot(s) have lengths of 30, 60, 120, 180, 240, 300, and 360 minutes respectively.

As shown in Table 1, the daily patient mix significantly varies (see, specifically, columns corresponding to *patient number*). Therefore, the scheduler must adopt override policies to accommodate patients' requirements while ensuring their treatment quality. Although the schedulers at Andreas Cancer Center were given instructional policies on how to override template slots [12], it was fairly challenging to follow these policies manually to ensure optimal patient assignments so as to maximize the utilization of the template slots. Therefore, we will compare the proposed MILP model with the manual implementations at the Center. Five priority parameters of the model were defined as follows $\lambda_1 = 50, \lambda_2 = 4, \lambda_3 = 1, \lambda_4 = 2, \lambda_5 = 3$ to control optimal patient assignments over the fixed template. Under this setting, the model gives the highest priority ($\lambda_1 = 50$) to allocate template slots to treat patients. Moreover, when there is no equal assignment for a patient, the model would have the following preference orders: the larger assignment, the combining override policy, and the breaking override policy ($\lambda_3 = 1 < \lambda_4 = 2 < \lambda_5 = 3$). The experience of using these policies varies among schedulers. The consensus in terms of how long it takes to perform each policy is approximately one minute for policy 1, five minutes for policy 2, and three minutes for policy 3. In addition, to ensure the template's intention of conserving the nurses' availability for the treatment of incoming patients, the impact of breaking a slot (policy 3) on the nurse time is also measured for comparison. Each time a slot is broken to accommodate two appointments, increases two extra time slots (~ 30 minutes) which require nurse resource.

3.2 Solution analysis

In this subsection, we analyze the experimental results. First, Out of the 22 working days, the proposed optimal assignments from the MILP model can completely accommodate the patient influx for 20 working days except for day 13 and day 21. For both of these days, the total patient volume was 62 which translated to 6,840 and 6,780 in total minutes. The volumes exceed the maximum possible number of 61 available slots designed into the template. The total number of required minutes also exceed the maximum number of minutes available, which is 6,750 minutes. In summary, the average daily patient volume was 52.1 with a standard deviation of 6.1. The 22-day data and patient assignments under policy 1, 2, and 3 from current manual operation and MILP model are outlined in table 1.

Table 1 also summarizes the override policy usage for these days. On average, Policy 1 has been suggested 2.1 times under the optimal method compared to 3.1 times under the manual method; which shows a 35% reduction in corresponding usage. Policy 2 has been performed 1.2 times on average under the optimal method compared to 2.4 times on average under the manual method, which shows an average reduction of 48% in corresponding overrides. And lastly, policy 3 has been suggested 1.2 times on average under the optimal method compared to 7.9 times on average under the manual method. Policy 3 is the most undesirable override and the optimal method has been able to reduce it by 85%. In overall, the optimal method outperforms the current manual process by 67%, where the override usages have been reduced from 13.4 times to 4.5 times on average. Overall, the results indicate that the proposed MILP model can save 30-minute override time for the schedulers, and 3 work hours for nurses on a regular day.

Figure 1 demonstrate via day 16 on how manual (middle panel) and optimal (right panel) methods work. The manual method results in 3 overrides under policy 1, 1 under policy 2 and 12 under policy 3. For example, a 1-hour appointment is scheduled at 9:30 am in a 3-hour slot starting at 9:15 am on chair 14 ; which meets the criteria for policy 1. On chair 13, the 5-hour slot starting at 9:00 am is combined with a part of 2-hour slot starting at 2:00 pm to be a 6-hour appointment under policy 2. As for policy 3, the 3-hour slot starting at 8:30 am on chair 9 has been broken into a 2-hour (8:30 am) and a 1-hour (10:30 am) appointment. The most undesirable manual override case is found on chair 5 where a 3-hour slot starting at 10:00 am has been broken into three appointments (two 30-minute and one 1-hour). This case often occurs when schedulers attempt to fit patients according to the patients' preference. On the other hand, the optimal method results in 3 overrides under policy 1, 1 under policy 2, and 6 under policy 3. This optimal method systematically caps the freedom of breaking a slot while suggests other options for schedulers in order to honor the template design for protecting nurse's time.

	Number of patients								Total	Manual policy				Optimal policy			
Day	30	60	120	180	240	300	360	Total	minutes	1	2	3	Total	1	2	3	Total
1	15	9	8	5	8	2	0	47	5,370	1	2	7	10	2	1	0	3
2	21	8	5	10	7	2	0	53	5,790	4	1	8	13	1	0	0	1
3	17	10	7	7	9	2	0	52	5,970	3	5	8	16	3	2	0	5
4	15	10	9	5	5	0	2	46	4,950	1	3	6	10	2	1	0	3
5	14	8	10	8	5	3	0	48	5,640	6	3	8	17	1	0	0	1
6	15	13	8	8	6	4	1	55	6,630	4	3	8	15	0	4	4	8
7	20	11	8	7	3	4	0	53	5,400	4	3	8	15	4	1	0	5
8	14	7	8	6	2	3	1	41	4,620	3	1	5	9	0	1	0	1
9	17	13	6	5	3	1	1	46	4,290	3	1	10	14	5	0	0	5
10	18	11	7	6	3	3	2	50	5,460	3	2	7	12	3	2	0	5
11	23	12	9	4	5	0	2	55	5,130	6	3	7	16	4	1	0	5
12	12	8	7	7	6	1	2	43	5,400	5	2	4	11	0	1	0	1
13	24	10	10	7	7	1	3	62	6,840	0	3	13	16	2	1	2	5
14	24	10	12	10	4	1	2	63	6,540	1	1	11	13	0	1	6	7
15	20	9	12	6	6	2	1	56	6,060	4	2	7	13	3	0	0	3
16	21	16	11	4	6	1	2	61	6,090	3	1	12	16	3	1	6	10
17	17	5	7	8	7	1	2	47	5,790	2	1	5	8	1	1	0	2
18	14	9	8	7	5	2	1	46	5,340	2	2	7	11	1	0	0	1
19	14	9	13	3	7	5	1	52	6,600	3	5	7	15	0	8	8	16
20	20	11	6	8	9	1	0	55	5,880	4	4	8	16	5	1	0	6
21	24	8	11	10	5	3	1	62	6,780	3	1	12	16	2	0	0	2
22	20	8	11	5	8	0	1	53	5,580	4	3	6	13	3	0	0	3
Template	23	8	10	11	6	2	1	61	6,750								
Average number of overrides										3.1	2.4	7.9	13.4	2.1	1.2	1.2	4.5

Table 1: 22 days scheduling data from Andreas Cancer Centre at the Mayo Clinic and override results from manual operations and the MILP model.

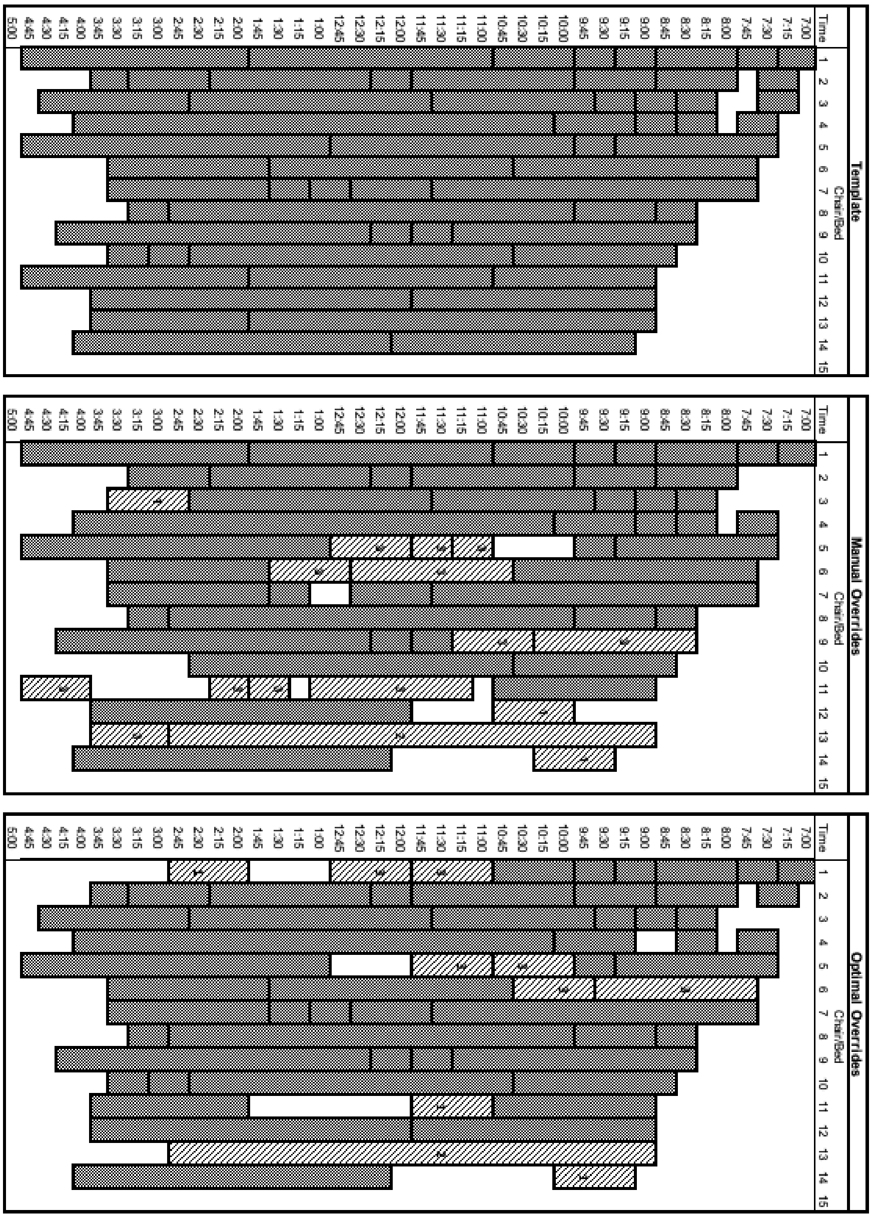

Fig. 1: The template which is deployed at the chemotherapy unit (left) manual override (middle) and optimal override (right) for day 16.

3.3 Sensitivity analysis on the parameters

In this subsection, we conduct sensitivity analysis on the parameters that are employed to prioritize the override policies in the objective function. In addition to the setting, denoted by (s1), $\lambda_3 = 1, \lambda_4 = 2$, and $\lambda_5 = 3$ in Section 3.2, we conduct two more settings. Thus, we investigate the sensitivity analysis of the following three settings.

$$(s1) : \lambda_3 = 1,\ \lambda_4 = 2,\ \lambda_5 = 3$$
$$(s2) : \lambda_3 = 1,\ \lambda_4 = 1,\ \lambda_5 = 1$$
$$(s3) : \lambda_3 = 1,\ \lambda_4 = 3,\ \lambda_5 = 2$$

Note that setting (s1) employed in Section 3.2 is preferred for the nursing resource (policy 1 > policy 2 > policy 3), that the setting (s2) gives equal preference over the three policies, and that the setting (s3) is preferred for schedulers (policy 1 > policy 3 > policy 2). Table 2 lists the results from the experiments. As illustrated, the average number of times that *policy 2* is used in setting (s1) is slightly greater than the one in setting (s3), while the average number of times that *policy 3* is used in setting (s1) is less than the one in setting (s3). This observation is consistent with what is expected.

	Average number of policies		
Setting	Policy 1	Policy 2	Policy 3
s1	2.1	1.2	1.2
s2	0.3	1.2	3.2
s3	1.4	1.0	1.7

Table 2: Sensitivity analysis of the parameters associated with the policies.

4 Concluding Remarks

In this paper, we introduced override policies for using a fixed chemotherapy scheduling template to accommodate the varying daily patient mix. In particular, we proposed a mixed-integer linear programming model to determine the optimal override policies, which may significantly mitigate schedulers' work by reducing operating time to find the best template slot assignments to patients. Numerical experiments were conducted to demonstrate the effectiveness of the proposed approach. Experimental results also indicated the efficiency of the computation. Furthermore, sensitivity analysis was conducted to demonstrate the flexibility of the proposed approach. Schedulers may tune the parameters to accommodate different preferences for using the model.

The proposed MILP model can be employed to evaluate the performance of a given scheduling template and thus can be incorporated into a high-level optimization problem to determine an optimal scheduling template. As the prerequisite of using the model is the availability of the patient mix, it is a natural extension to consider the development of an overriding schema where patients of different types are arriving in a dynamic setting.

References

1. Bailey, N.T.: A study of queues and appointment systems in hospital out-patient departments, with special reference to waiting-times. Journal of the Royal Statistical Society. Series B (Methodological) pp. 185–199 (1952)
2. Berg, B.P., Denton, B.T., Erdogan, S.A., Rohleder, T., Huschka, T.: Optimal booking and scheduling in outpatient procedure centers. Computers & Operations Research **50**, 24–37 (2014)
3. Cayirli, T., Veral, E.: Outpatient scheduling in health care: a review of literature. Production and operations management **12**(4), 519–549 (2003)
4. Cayirli, T., Veral, E., Rosen, H.: Designing appointment scheduling systems for ambulatory care services. Health care management science **9**(1), 47–58 (2006)
5. Denton, B., Gupta, D.: A sequential bounding approach for optimal appointment scheduling. IIE transactions **35**(11), 1003–1016 (2003)
6. Erdogan, S.A., Denton, B.: Dynamic appointment scheduling of a stochastic server with uncertain demand. INFORMS Journal on Computing **25**(1), 116–132 (2013)
7. Fetter, R.B., Thompson, J.D.: Patients' waiting time and doctors' idle time in the outpatient setting. Health services research **1**(1), 66 (1966)
8. Gupta, D., Denton, B.: Appointment scheduling in health care: Challenges and opportunities. IIE transactions **40**(9), 800–819 (2008)
9. Gurvich, I., Luedtke, J., Tezcan, T.: Staffing call centers with uncertain demand forecasts: A chance-constrained optimization approach. Management Science **56**(7), 1093–1115 (2010)
10. Ho, C.J., Lau, H.S.: Minimizing total cost in scheduling outpatient appointments. Management science **38**(12), 1750–1764 (1992)
11. Huang, Y.L., Bach, S.M., Looker, S.A.: Chemotherapy scheduling template development using an optimization approach. International journal of health care quality assurance (2019)
12. Huang, Y.L., Bryce, A.H., Culbertson, T., Connor, S.L., Looker, S.A., Altman, K.M., Collins, J.G., Stellner, W., McWilliams, R.R., Moreno-Aspitia, A., et al.: Alternative outpatient chemotherapy scheduling method to improve patient service quality and nurse satisfaction. Journal of oncology practice **14**(2), e82–e91 (2018)
13. Jiang, R., Ryu, M., Xu, G.: Data-driven distributionally robust appointment scheduling over wasserstein balls. arXiv preprint arXiv:1907.03219 (2019)
14. Kaandorp, G.C., Koole, G.: Optimal outpatient appointment scheduling. Health Care Management Science **10**(3), 217–229 (2007)
15. Kall, P., Wallace, S.W.: Stochastic programming. Springer (1994)
16. Kong, Q., Lee, C.Y., Teo, C.P., Zheng, Z.: Scheduling arrivals to a stochastic service delivery system using copositive cones. Operations research **61**(3), 711–726 (2013)
17. Mak, H.Y., Rong, Y., Zhang, J.: Sequencing appointments for service systems using inventory approximations. Manufacturing & Service Operations Management **16**(2), 251–262 (2014)
18. Mak, H.Y., Rong, Y., Zhang, J.: Appointment scheduling with limited distributional information. Management Science **61**(2), 316–334 (2015)
19. Muthuraman, K., Lawley, M.: A stochastic overbooking model for outpatient clinical scheduling with no-shows. Iie Transactions **40**(9), 820–837 (2008)
20. Shen, S., Wang, J.: Stochastic modeling and approaches for managing energy footprints in cloud computing service. Service Science **6**(1), 15–33 (2014)
21. Turkcan, A., Zeng, B., Lawley, M.: Chemotherapy operations planning and scheduling. IIE Transactions on Healthcare Systems Engineering **2**(1), 31–49 (2012)
22. Zacharias, C., Pinedo, M.: Managing customer arrivals in service systems with multiple identical servers. Manufacturing & Service Operations Management **19**(4), 639–656 (2017)