Stochastic Scheduling of Chemotherapy Appointments Considering Patient Acuity Levels

Sırma Karakaya, Serhat Gul[1]

Department of Industrial Engineering, TED University, Ankara, Turkey

Melih Çelik

School of Management, University of Bath, Bath, United Kingdom

Abstract

The uncertainty in infusion durations and non-homogeneous care level needs of patients are the critical factors that lead to difficulties in chemotherapy scheduling. We study the problem of scheduling patient appointments and assigning patients to nurses under uncertainty in infusion durations for a given day. We consider instantaneous nurse workload, represented in terms of total patient acuity levels, and chair availability while scheduling patients. We formulate a two-stage stochastic mixed-integer programming model with the objective of minimizing expected weighted sum of excess patient acuity, waiting time and nurse overtime. We propose a scenario bundling-based decomposition algorithm to find near-optimal schedules. We use data of a major university hospital to generate managerial insights related to the impact of acuity consideration, and number of nurses and chairs on the performance measures. We compare the algorithm with several relevant scheduling heuristics and assess the value of stochastic solution.

Keywords: OR in health services, chemotherapy scheduling, patient acuity, stochastic programming

1. Introduction

Recent progresses in cancer treatment have resulted in an increase in patient survival rates but escalated the demand for outpatient chemotherapy (Katayama et al., 2021). The rising demand has made the task of managing outpatient chemotherapy clinics (OCCs) challenging for the clinic directors.

Premedication and infusion activities, which are performed consecutively, represent the two main steps of an OCC. During the premedication phase, the patient is set up for infusion by a nurse on a chemotherapy chair, and given some drugs for protecting them against the side effects of chemotherapy agents. In the infusion phase, patient is administered the chemotherapy agents through an intravenous bag and a catheter and monitored by a nurse.

Outpatient chemotherapy appointments are generally created by following a two-step approach. In the first step, called *chemotherapy planning*, daily plans are obtained by assigning patient treatments to specific

[1]Corresponding author, e-mail: serhat.gul@tedu.edu.tr

days. Then, in the *chemotherapy scheduling* step, appointment times are set for the patients in a daily plan to generate daily schedules. In this article, we focus on the second step by assuming that the chemotherapy planning decisions were already made.

Scheduling outpatient chemotherapy appointments is difficult due to the uncertainty in infusion durations and the simultaneous need for multiple resources (in particular, nurses and chairs) during each treatment phase. Furthermore, balancing the trade-off between conflicting performance measures is necessary for the OCC managers to address patient and provider expectations at the same time. Combined with these issues, the variation in acuity levels among different patients makes scheduling appointments for OCCs challenging.

Patient acuity is a type of patient classification system and represents the intensity of nursing care needed by a patient (Jennings, 2008). Using acuity levels while managing operations leads to a balanced nurse workload, improved nurse and patient satisfaction, and hence higher quality of care. Besides, acuity-based scheduling enhances the safety of the providers, as the chance of unexpected exposure to hazardous drugs would be decreased (Rodriguez et al., 2020).

OCC managers must ensure that nurses are not overburdened at any point in time during chemotherapy treatment. The level of burden can be measured by the total acuity levels of the patients treated by a nurse at a particular time. Hence, minimizing the excess acuity above the target load of a nurse is critical for the managers to ensure high-quality care. Moreover, the managers also have to carefully balance the trade-off between patient waiting time and nurse overtime. Nurse overtime is highly undesired by the providers because it inflates the operating costs. It also has negative impacts on the job performances and satisfaction levels of nurses. Experiencing excessive waiting creates stress and frustration for chemotherapy patients. It must be noted that patient waiting time is found to be the primary reason for patient dissatisfaction with appointments in oncology services (Kallen et al., 2012).

Data obtained from OCCs exhibits significant uncertainty in infusion durations (Castaing et al., 2016; Demir et al., 2021). The divergence in infusion durations from anticipated amounts may arise due to several reasons, such as the premature termination of patient treatments as a result of adverse reaction to chemotherapy agents, prolonged treatments when complications occur, and compulsory changes in medication plans (Castaing et al., 2016). Uncertainty in infusion durations is the principal reason behind the difficulty of managing instantaneous nurse workload. It also makes the allocation of time to patient appointments critical. Overestimation of the treatment durations may increase nurse overtime while it reduces patient waiting time. On the other hand, underestimated durations may worsen patient waiting time while positively affecting nurse overtime.

Having a limited number of nurses and chairs in the OCCs has significant impact into the critical performance measures for the clinics. Hence, resource assignment decisions must be carefully coordinated by considering the process flow characteristics specific to the OCCs. In the clinics, a patient seizes both resources simultaneously during the whole treatment. At a given time, a nurse can apply at most one premedication activity, but can monitor the infusion activities of multiple patients. However, the total workload of a nurse

at any given time should be limited. Therefore, the nurse assignment decisions must be based on patient acuity levels, nurse skill levels, and instantaneous nurse workload capacity levels.

In this article, we study the problem of sequencing patient appointments, setting appointment times, and assigning patients to nurses for an OCC with a limited number of chairs while controlling the instantaneous workload of nurses under infusion duration uncertainty. We assume that the daily list of patient appointments is predetermined. We formulate a two-stage stochastic mixed integer programming (TSMIP) model for the daily scheduling problem. We minimize the expected weighted sum of excess acuity level over the nurse capacity, nurse overtime and patient waiting time in the objective function. The expected value of the weighted sum of performance measures is calculated based on the scenarios that we generate by sampling from infusion duration distributions. Through exploiting the special structure of the TSMIP model, we propose a scenario bundling-based decomposition algorithm to solve the model. We determine the appropriate size of scenario bundles which help find near-optimal chemotherapy schedules within a limited time by conducting a comprehensive set of computational experiments. The algorithm is tested on several problem instances that are created based on data from a large university hospital. We compare our algorithm with a commercial solver and several practically relevant heuristics from the literature. We estimate the value of generating a stochastic solution rather than a deterministic one for our problem. We analyze the sensitivity of the near-optimal solutions to the weights of excess acuity, waiting time and overtime. We assess the impact of acuity consideration, and the number of chairs and nurses to the performance measure values.

We organize the remaining parts of the article as follows: In Section 2, we review the literature on outpatient chemotherapy scheduling studies, and position our study within the literature. In Section 3, we provide the detailed problem description and formulate the TSMIP model. In Section 4, we discuss the scenario bundling-based decomposition algorithm we propose. In Section 5, we demonstrate a comprehensive computational study. Finally, in Section 6, we share concluding remarks and point to further research directions.

2. Literature Review

We review the outpatient chemotherapy scheduling literature under two categories. The first category includes the studies on deterministic scheduling, while the latter includes the articles that consider uncertainty in infusion durations. The related stream of research is comprehensively reviewed in Lamè et al. (2016). Furthermore, scheduling of operations in other types of outpatient clinics is reviewed in Cayirli & Veral (2003), Gupta & Denton (2008), and Ahmadi-Javid et al. (2017). In the literature on general outpatient clinics, the articles that study a multi-resource setting are the most similar ones to the studies on outpatient chemotherapy scheduling. The differences between these two streams of research are discussed in detail in Demir et al. (2021).

We start our review with deterministic outpatient chemotherapy scheduling studies. As opposed to the study in this article, these ignore the uncertainty in infusion durations. Among the articles in the first

category, Turkcan et al. (2012), Benzaid et al. (2020), and Hooshangi-Tabrizi et al. (2020) studied both planning and scheduling problems. On the other hand, Heshmat et al. (2018), Hesaraki et al. (2019, 2020), and Liang & Turkcan (2016) focused only on the scheduling phase, as is the case also in our article.

Turkcan et al. (2012) solved planning and scheduling problems consecutively using separate integer programming (IP) models. In the planning phase, they assigned patients only to days based on measures including nurse overtime, idle time, and delays in treatment starts. In the scheduling phase, they assigned patients to time slots, nurses and chairs while minimizing the completion time of the last treatment of each day. Patient acuity levels were taken into account while modeling nurse capacity restrictions in both phases. Benzaid et al. (2020) proposed three IP models which must be solved consecutively to determine the day, starting slot and nurse for a patient appointment. They also considered acuity levels in their models. Hooshangi-Tabrizi et al. (2020) solved planning and scheduling problems using a single IP model. They assigned patient treatments to days, chairs and time slots over a planning horizon by assuming that nurses are preassigned to chairs. In the objective function, they considered the number of patients assigned to floating nurses, unfavored patient-to-slot assignments, non-scheduled appointment requests, and nurse overtime. Using another IP model, they solved a rescheduling problem at each day by updating the start times of appointments scheduled to that day. They did not consider acuity levels in their models. In other words, they assumed that each patient requires the same level of attention from a nurse.

Liang & Turkcan (2016) proposed different MIP models to be used for different types of care delivery. The first type assumes patients can be assigned to different nurses at different treatment visits. Patients are assigned to both nurses and slots by minimizing patient waiting time and nurse overtime in the first model. One the other hand, the second delivery type emphasizes continuity of care and assumes that the same nurse provides care for a patient at each visit day. Hence, patients are assigned to only slots in the second model. They considered patient acuity levels while representing nurse workload restrictions. They minimized overtime and excess nurse workload in the objective function. Heshmat et al. (2018) first clustered patients based on treatment durations, cancer type and acuity levels. Then, they assigned each patient cluster to a nurse, slot and group of chairs using an IP formulation. They minimized the completion time of the last treatment of a daily schedule. Hesaraki et al. (2019) generated daily templates that consist of appointment slots by proposing an IP model. Although they considered chair and nurse capacity limitations while generating an appointment template, they ignored patient acuity issues. They minimized flowtime and makespan in the objective function. Hesaraki et al. (2020) proposed a mixed integer program (MIP) to assign patients to time slots and nurses. They allowed multiple nurses to monitor a single patient consecutively throughout the treatment, but they favored fewer nurse changes to provide a safer process and reduce confusion of patients and nurses. Furthermore, they balanced workload among nurses, minimized nursing full-time equivalent and the time between patient ready time and appointment time. They did not consider acuity levels in the model, but discussed how their model can be generalized by modeling revised nursing capacity and workload constraints that include acuity levels.

Our study belongs to the second category of articles, as the uncertainty is explicitly considered in our model. In this category, Castaing et al. (2016) set the appointment times of patients by preserving the original appointment sequence. They solved the problem for a single nurse, and therefore assigned patients only to chairs. They minimized the expected patient waiting time and total length of operations. As opposed to Castaing et al. (2016), we sequence patients and formulate the model for multiple nurse case. Furthermore, we assign patients to nurses while considering the acuity levels of patients and skill levels of nurses. The objective function of our model includes excess workload in addition to waiting time and nurse overtime.

Gul (2021) extended the model in Castaing et al. (2016) by considering patient-to-nurse assignment decisions. Furthermore, they grouped patients according to their treatment durations and restricted the number of patients that can be assigned from each patient group to each nurse on a given day. Hence, they balanced total daily workload among nurses. In this study, different from Gul (2021), we consider prevention of excess nurse workload at any moment in time during the day. Also, we take into account patient acuity levels and differentiate nurses from each other based on their skill levels. Finally, we allow changes in the sequence of patients.

Mandelbaum et al. (2020) used a data-driven approach to schedule appointments of an infusion unit under uncertainty in both infusion durations and patient punctuality. However, their model is more appropriate for general outpatient clinics, as nurses were not considered. The servers in their infinite-servers relaxation-based approach represent only chairs. On the other hand, we thoroughly include complexities related to patient-to-nurse assignment decisions in our model. Slocum et al. (2020) tested simple deterministic scheduling heuristics using a discrete-event simulation model. The heuristics group patients according to the expected infusion durations and assign patients to fixed time blocks in a daily schedule. The number of time blocks in a daily schedule can be only 2 or 3 according to their approach. Our study is quite different from this study, as they did not propose any stochastic optimization approach to set appointment times. Alvarado & Ntaimo (2018) proposed three mean-risk stochastic programming (SP) models to assign patient treatments to days, slots, chairs and nurses. However, the model can be used to schedule only a single patient. Therefore no patient sequencing or simultaneous assignment decisions are given in the model. On the other hand, our model considers all those complexities and can provide a full daily schedule once it is solved.

Finally, Demir et al. (2021) proposed a two-stage stochastic programming SP model in which patients were sequenced and appointment times were set in the first stage, while patients were assigned to nurses and chairs in the second stage. However, they assumed that nurses have identical skill levels and hence considered nurse assignment decisions in the second stage of the formulation. They also did not consider acuity levels of the patients and did not prevent excessive instantaneous workload of nurses in their model. They assumed that patient-to-nurse assignments would be balanced without imposing explicit restrictions. On the other hand, we model varying skill levels among nurses and penalize excess instantaneous nurse workload. Furthermore, we make nurse assignment decisions in the first stage. Besides, we assign patients to time slots while determining appointment times, which could be set to any integer values in Demir et al.

(2021).

3. Problem Description

We study the problem of sequencing patient appointments, setting appointment times, and assigning patients to nurses under uncertainty in infusion durations for a given day. We consider instantaneous nurse workload, which is represented in terms of total patient acuity levels at any given time, and chair availability while scheduling patients. We formulate a two-stage stochastic mixed-integer programming (TSMIP) model to represent the problem.

At the first stage of the model, three main decisions are given: (i) patient appointment sequencing, (ii) appointment time setting by assigning each patient to a time slot, and (iii) patient-to-nurse assignment. Each time slot in the model represents a small and fixed time interval (e.g., a 15-minute block). Assigning a patient to a single slot at the first stage does not imply that the treatment of the patient will require only one slot. The slot assignment only determines the appointment time of the patient. As we assume that patients arrive on time, the assigned slot also represents the time of patient arrival to the clinic. Some nurses cannot be assigned to some patients, because the skill level of a nurse may be insufficient for successful treatment of the patient, due to a high acuity level. A nurse is considered to be capable of treating a patient only if the acuity level of the patient is not larger than the highest level of acuity that the nurse can handle.

Supported with data from a major oncology hospital (as will be discussed in Section 5), we assume that the pre-medication activity duration can be known in advance and requires constant amount of time for each patient. On the other hand, the same data set exhibits significant uncertainty in infusion durations. Therefore, infusion durations are assumed to be unknown before making first-stage decisions. After the duration uncertainty is resolved, the second-stage decisions are given to determine the actual treatment start time for each patient and calculate performance measure values for each scenario realization. The performance measures of the study are all related to the second-stage formulation, and they include patient waiting time, nurse overtime, and excess of total acuity levels over the nurse capacity. We assume that patient waiting time mainly measures patient satisfaction, nurse overtime measures provider related costs and satisfaction, and excess acuity measures quality of patient care. Patient waiting time measures the difference between the actual treatment start time and patient arrival time. Delays in patient treatments may be observed due to longer than expected durations of preceding treatments. Such deviations in the schedules may lead to unavailability of nurses and chairs, which in turn result in patient waiting. Nurse overtime refers to the time spent by a nurse beyond the shift finish time. Overtime may be observed due two main reasons. First, the daily patient mix might be overloaded due to the decisions made during the chemotherapy planning phase, which is beyond the scope of this study. In that case, an overtime would be observed no matter how the manager schedules the patients in the daily list. Second, unnecessary idleness of chairs and nurses may lead to nurse overtime. It is evident that managing the simultaneous usage of chairs and nurses is critical for improving both patient waiting time and nurse overtime.

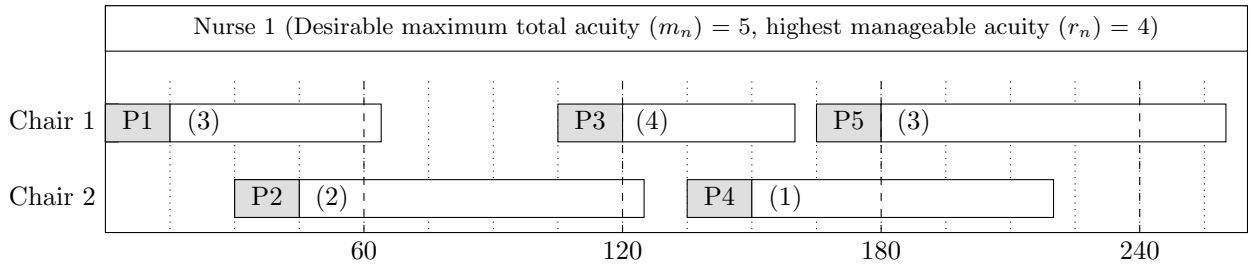

Figure 1: A sample half-shift realization for a single nurse and two chairs, respectively. The darker bars represent premedication, whereas numbers in brackets reflect the acuity level for each patient.

Simultaneous treatment of excessive number of patients by a single nurse is also penalized in the model. Each nurse has a capacity limit which determines the desired level of the sum of acuity values of the patients that could be treated by this nurse at any point in time. Excess of acuity levels over the nurse capacity is allowed at any time period, but incurs a penalty in the model. We also note that the violation level of the nurse capacity is checked independently at each scenario.

At the beginning of the treatment (i.e. during premedication), patients need special care by their assigned nurses. Therefore, a nurse cannot start the treatment of multiple patients at a given time slot. However, proctoring the infusion of other patients while performing the premedication of a patient is allowed. Besides the nurse capacity, the number of chairs is also a limiting factor on the number of patients that can be served simultaneously. The treatment of a patient starts immediately when a chair is assigned to the patient. However, to reduce model complexity, we do not represent chair assignment decisions explicitly in our formulation as chairs are identical. Chair assignments are implicitly considered in capacity restrictions and while making patient treatment start decisions.

To exemplify the operations in the OCC and understand the performance measures involved, a simplified half-shift (240-minute) realization focusing on a single nurse, two chairs, and five patients is provided in Figure 1. The arrival times of the five patients P1 through P5 at 0, 30, 105, 120, and 150 minutes, respectively. Each premedication takes 15 minutes and the infusion times are 47, 80, 40, 70, and 80 minutes. As can be observed, patient P1 arrives immediately, sits on Chair 1, and begins premedication. While the treatment of Patient P1 is ongoing, patient P2 arrives at the 30^{th} minute, finds the nurse and Chair 2 available, and starts premedication immediately. The process continues until patient P5 vacates Chair 1 in the 260^{th} minute once the infusion ends.

There are two cases of patients waiting in Figure 1. Patient P4 arrives at the 120^{th} minute and finds no available chair. Patient P2 leaves Chair 2 at the 125^{th} minute, but P4 needs to wait until the beginning of the next slot at the 135^{th} minute to start premedication. Similarly, P5 arrives at the 150^{th} minute, but needs to wait for 15 minutes due to unavailability of chairs. Hence, the total waiting time is 30 minutes. As the last patient leaves the OCC at the 260^{th} minute, total nurse overtime is $260 - 240 = 20$ minutes. It can also be inferred that the desirable acuity level of 5 units is only exceeded between the start of premedication for P3 and the departure of P2, which yields an excess acuity of 1 unit for a total of 20 minutes.

We assume that a particular *chair assignment policy* is imposed by the clinic for arriving patients, which may sometimes result in a *patient treatment sequence* different from the *patient appointment sequence*. The policy, which consists of two rules to be checked consecutively, is practically relevant as it is based on our observations in practice. We next discuss each rule associated with the policy along with the motivation behind rules.

- **Rule 1** - *Assign the first available chair to a patient whose nurse is available.* Suppose that there are two patients waiting for treatment, and the nurse of only one patient is ready when a chair becomes available. If the chair is assigned to the patient whose nurse is busy with another patient, then both chair and nurse idleness would be induced. Even though we do not explicitly consider it in the objective function, idle time is an important measure for clinic managers. Furthermore, reducing idle time would lead to reduction in overtime if overtime is observed in a clinic. Therefore, when the patients arrive at the the clinic for treatment, the clinics give priority to the ones whose nurses are available.

- **Rule 2** - *Assign the first available chair to the patient who arrived the earliest, if nurses of multiple patients are ready for treatment.* This rule implies that the appointment sequence would not change in the treatment phase, as all patients are assumed to arrive on time. Allowing a patient to move ahead of a patient who arrived earlier while their nurses are both available would not be possible in practice, as the urgency levels of all patients are the same in outpatient clinics. Such reordering practice would be be a source of grievance to the patients who arrived earlier.

In chemotherapy clinics, treatment start time decisions for patients are given dynamically before the uncertainties in all infusion durations are resolved. In the corresponding multi-stage SP model, the time points at which a chair becomes available would represent decision stages after the first stage. However, the *chair assignment policy* we enforce allows one to avoid formulating the problem as a multi-stage SP model. All treatment start time decisions can be made in the second stage after observing each scenario realization, as the policy is myopic, in the sense that it depends only on the patient appointment sequence and availability of nurses at the time that a chair becomes available. Therefore, under the conditions dictated by the *chair assignment policy*, the optimal appointment schedule found by the two-stage SP model would be the same as that of the multi-stage SP formulation version of our problem.

Assuming that a finite set of scenarios would sufficiently capture the uncertainty in infusion durations, we formulate the following TSMIP model based on the notation explained in Table 1.

$$\min \quad \mathcal{Q}(\boldsymbol{u}, \boldsymbol{x}, \boldsymbol{t}) \tag{1}$$

$$\text{s.t.} \quad u_{ij} + u_{ji} = 1 \qquad \forall i, j \in P : j > i \tag{2}$$

$$\sum_{n \in N_i} x_{in} = 1 \qquad \forall i \in P \tag{3}$$

Table 1: Notation for the TSMIP model

Index sets

P	Patients
P_d	Patients including the dummy patient, i.e., $P_d = P \cup \{0\}$
N	Nurses
N_i	Nurses that can treat patient $i \in P$
C	Chairs
S	Time slots
Ω	Scenarios

Parameters

σ	Duration of a time slot
q_s	$= (s-1)\,\sigma$, beginning time of slot $s \in S$
h	Shift length of the chemotherapy clinic
a_i	Acuity level of patient $i \in P$
r_n	Highest level of patient acuity that nurse $n \in N$ can manage
m_n	Desirable upper limit on the sum of acuity levels of the patients who are simultaneously treated by nurse $n \in N$
b	Premedication duration
$c_i(\omega)$	Infusion duration for patient $i \in P$ under scenario $\omega \in \Omega$
$p_i(\omega)$	$= \left\lceil \frac{b+c_i(\omega)}{\sigma} \right\rceil$, number of time slots required to complete the treatment of patient $i \in P$ under scenario $\omega \in \Omega$
$f_{is}(\omega)$	$= \max\{1, s - p_i(\omega) + 1\}$, earliest slot that the treatment of patient $i \in P$ can start if treatment proceeds at slot $s \in S$ under scenario $\omega \in \Omega$
$g_{is}(\omega)$	$= \max\left\{1, s - \left\lceil \frac{b}{\sigma} \right\rceil + 1\right\}$, earliest slot that the treatment of patient $i \in P$ can start if premedication proceeds at slot $s \in S$ under scenario $\omega \in \Omega$
λ	Penalty parameter for waiting time (Sum of penalty parameters for waiting time and overtime is 1)
β	Penalty parameter for the excess of total acuity levels

First-stage decision variables

u_{ij}	$= \begin{cases} 1 & \text{if patient } i \in P \text{ precedes patient } j \in P \text{ (general precedence) in appointment sequence} \\ 0 & \text{otherwise} \end{cases}$
x_{in}	$= \begin{cases} 1 & \text{if patient } i \in P \text{ is assigned to nurse } n \in N_i \\ 0 & \text{otherwise} \end{cases}$
t_{is}	$= \begin{cases} 1 & \text{if patient } i \in P \text{ is assigned to time slot } s \in S \\ 0 & \text{otherwise} \end{cases}$

Second-stage decision variables

$z^s_{in}(\omega)$	$= \begin{cases} 1 & \text{if the treatment of patient } i \in P, \text{ carried out by nurse } n \in N_i, \text{ starts in time slot } s \in S \\ & \text{under scenario } \omega \in \Omega \\ 0 & \text{otherwise} \end{cases}$
$w_i(\omega)$	Waiting time of patient $i \in P$ under scenario $\omega \in \Omega$
$e_n(\omega)$	Excess of total acuity level over the desirable upper limit of nurse $n \in N$ under scenario $\omega \in \Omega$
$o_n(\omega)$	Overtime for nurse $n \in N$ under scenario $\omega \in \Omega$

$$\sum_{s\in S} t_{is} = 1 \qquad \forall i \in P \tag{4}$$

$$\sum_{s\in S} q_s t_{js} \geq \sum_{s\in S} q_s t_{is} - M(1-u_{ij}) \qquad \forall i,j \in P : j \neq i \tag{5}$$

$$u_{ij} \in \{0,1\} \qquad \forall i,j \in P : j \neq i \tag{6}$$

$$x_{in} \in \{0,1\} \qquad \forall i \in P,\, n \in N_i \tag{7}$$

$$t_{is} \in \{0,1\} \qquad \forall i \in P,\, s \in S \tag{8}$$

where $\mathcal{Q}(\boldsymbol{u},\boldsymbol{x},\boldsymbol{t}) = E_\xi\left[Q\left(\boldsymbol{u},\boldsymbol{x},\boldsymbol{t},\boldsymbol{\xi}\left(\omega\right)\right)\right]$, and $Q(\boldsymbol{u},\boldsymbol{x},\boldsymbol{t},\boldsymbol{\xi}(\omega)))$ is given by:

$$\min \quad \lambda \sum_{i\in P} w_i(\omega) + (1-\lambda)\sum_{n\in N} o_n(\omega) + \beta \sum_{n\in N} e_n(\omega) \tag{9}$$

$$\text{s.t.} \quad \sum_{n\in N_i}\sum_{s\in S} z_{in}^s(\omega) = 1 \qquad \forall i \in P \tag{10}$$

$$z_{in}^s(\omega) \leq x_{in} \qquad \forall i \in P, n \in N_i,\, s \in S \tag{11}$$

$$\sum_{i\in P} z_{in}^s(\omega) \leq 1 \qquad \forall s \in S,\, n \in N_i \tag{12}$$

$$w_i(\omega) = \sum_{n\in N_i}\sum_{s\in S} q_s z_{in}^s(\omega) - \sum_{s\in S} q_s t_{is} \qquad \forall i \in P \tag{13}$$

$$\sum_{i\in P}\sum_{n\in N_i}\sum_{u=f_{is}(\omega)}^{s} z_{in}^u(\omega) \leq |C| \qquad \forall s \in S \tag{14}$$

$$\sum_{s\in S} q_s z_{jn}^s(\omega) + M(3-u_{ij}-x_{in}-x_{jn}) \geq \sum_{s\in S} q_s z_{in}^s(\omega) + b \qquad \forall i,j \in P : j \neq i,\, n \in N_i \tag{15}$$

$$\sum_{u=1}^{s} t_{iu} + u_{ij} + x_{in} - \sum_{u=1}^{max\{1,s-1\}} z_{in}^u(\omega) - \sum_{\substack{k\in P\\ k\neq i,j}}\sum_{u=g_{is}(\omega)}^{s} z_{kn}^u(\omega)$$

$$- \frac{1}{|C|}\sum_{\substack{k\in P\\ k\neq i,j}}\sum_{v\in N_k}\sum_{u=f_{is}(\omega)}^{s} z_{kv}^u(\omega) \leq z_{in}^s(\omega) + 2 \qquad \forall i \in P, j \in P_d : j \neq i, n \in N_i, s \in S \tag{16}$$

$$\sum_{i\in P}\sum_{u=f_{is}(\omega)}^{s} a_i z_{in}^u(\omega) \leq m_n + e_n(\omega) \qquad \forall n \in N_i,\, s \in S \tag{17}$$

$$o_n(\omega) \geq \sum_{s\in S} q_s z_{in}^s(\omega) + b + c_i(\omega) - h - M(1-x_{in}) \qquad \forall i \in P,\, n \in N_i \tag{18}$$

$$z_{in}^s(\omega) \in \{0,1\} \qquad \forall i \in P, n \in N_i,\, s \in S \tag{19}$$

$$w_i(\omega) \geq 0 \qquad \forall i \in P \tag{20}$$

$$o_n(\omega), e_n(\omega) \geq 0 \qquad \forall n \in N \tag{21}$$

The objective function (1) minimizes only the expected second-stage function, as no penalty is incurred related to the first-stage decisions. In particular, the function minimizes expected weighted sum of patient

waiting time, nurse overtime and excess of total acuity levels over the nurse capacity.

First-stage constraints are formulated by (2)-(8). Constraints (2) ensure that either patient $i \in P$ precedes patient $j \in P$ in the appointment sequence, or vice versa. Constraints (3) enforce each patient to be assigned to a nurse who has appropriate skills to treat the patient. More formally, nurse $n \in N$ can treat patient $i \in P$ only if the acuity level of patient $i \in P$ is within the limit that nurse $n \in N$ can manage. Therefore, the set N_i for patient $i \in P$ includes all nurses for whom the condition of $a_i \leq r_n$ is satisfied. Constraints (4) impose each patient to be assigned to a time slot, which represents the appointment time of patient. Constraints (5) stipulates that if patient $i \in P$ precedes patient $j \in P$ in the appointment sequence, then the appointment time of patient $i \in P$ cannot be later than that of patient $j \in P$. Constraints (6)-(8) represent the binary restrictions on the first-stage variables.

For a given set of first-stage decision variable values and a particular scenario realization, the expression in (9) determines the second-stage objective function value. Note that λ parameter controls the trade-off between waiting time and overtime, while β penalizes the excess of total acuity levels over the nurse capacity.

The constraint set for the second-stage scenario subproblem is represented by (10)-(20). Constraints (10) ensure that the treatment of a patient can be started by a single nurse and at a single slot. Constraints (11) guarantee that the nurse assigned to a patient at the first stage performs the treatment of the patient. Constraints (12) impose a nurse to start the treatment of at most one patient in any slot. Constraints (13) calculate the waiting time of a patient by finding out the difference between treatment start time and appointment time. Through identifying the patients whose treatment proceeds at slot $s \in S$, constraints (14) ensure that the number of simultaneously treated patients in the slot cannot exceed the number of chairs. Constraints (15) enforce a restriction valid for only the patients $\{i, j\} \in P$ who are assigned to the same nurse by maintaining that the treatment of patient $j \in P$ can start only after the premedication of patient $i \in P$ ends, if patient $i \in P$ precedes patient $j \in P$ in the appointment sequence.

Constraints (16) represent the particular chair assignment policy employed in the study while determining the treatment start time of arriving patients. Suppose that the following three conditions hold for patient $i \in P$: (a) the patient arrives before or at slot $s \in S$, (b) the treatment of this patient has not started before slot $s \in S$, and (c) the nurse assigned to the patient is not busy with premedication at slot $s \in S$. Assume that these three conditions also hold for patient $j \in P$, where patient $i \in P$ precedes $j \in P$ in the appointment sequence. If there is a single available chair at slot $s \in S$, then the treatment priority is given to patient $i \in P$. Under such circumstances, due to constraints (14) and (16), only the treatment for patient $i \in P$ must start at slot $s \in S$. This represents Rule 1 of the chair assignment policy. On the other hand, if (a) and (b) hold for both patients, but (c) holds only for patient $j \in P$, and there is an available chair, then the treatment for patient $j \in P$ must start at slot $s \in S$. This is enforced due to Rule 2 of the chair assignment policy.

Note that if constraints (16) are defined for $\{i, j\} \in P$, then u_{ij} never becomes 1 for patient $i \in P$ when i represents the last patient in the appointment sequence. As a result, the last patient cannot be enforced to

start at slot $s \in S$ even when the sufficient conditions are maintained. Therefore, we make use of a dummy patient $j \in P_d$ (i.e., $j = 0$), and preset $u_{ij} = 1$ for all patients $i \in P$ when $j = 0$ in constraints (16).

Constraints (17) calculate the sum of acuity levels of the patients treated by a nurse at any time slot. The resulting value is compared with the nurse capacity to determine the excess of total acuity levels for each slot. Constraints (18) compute overtime for each nurse by finding the discharge time of the last patient treated by the nurse. If the discharge time is smaller than the shift length, nurse overtime is set to zero. Constraints (19)-(21) are the binary and sign restrictions associated with the second-stage variables.

4. Solution Methodology

As will become evident in Section 5, the TSMIP model defined by (1)-(21) can be solved to optimality for only small instances with a limited number of patients, nurses, chairs, and scenarios. On the other hand, our preliminary experiments showed that when the patient sequence is fixed, the remaining part of the model can be solved in reasonable time for much larger instances. Motivated by this observation, we propose a heuristic approach that decomposes the original problem into scenario bundles, solves these bundles as subproblems, and finds a feasible patient sequence based on the bundle subproblem that yields the best solution. This sequence is then improved by means of a search procedure in the improvement phase. Our computational experiments show that while the scenario bundle decomposition approach requires the bulk of computational time for the heuristic, it generates high-quality initial solutions. The subsequent improvement step requires much shorter computational times, as the patient sequence is fixed in this step.

Scenario bundling approach is generally used to solve SP models by integrating it within the classical decomposition algorithms. For example, it is integrated within the progressive hedging algorithm (Escudero et al., 2013; Crainic et al., 2014; Gade et al., 2016; Jiang et al., 2021), alternative Lagrangian decomposition algorithms (Escudero et al., 2013), and L-shaped algorithm (Oliveira et al., 2011). To the best of our knowledge, this is the first study using scenario bundling to obtain promising initial solutions and integrating it with an improvement heuristic that consists of local search and diversification steps.

In what follows, we discuss the details of the scenario bundling-based decomposition (SBBD) algorithm and the improvements used to increase the efficiency of the algorithm, as well as that of the model given by (1)-(21).

4.1. Scenario Bundling-Based Decomposition Algorithm

The SBBD algorithm is a two-phase approach. An initial solution is found during the *construction* phase, followed by the improvement of this solution by means of a search algorithm in the *improvement* phase.

The construction phase starts by dividing the scenario set into nb scenario *bundles*, each of which except the last one consists of $|\Omega_b|$ scenarios. The methodology to determine nb, and thus $|\Omega_b|$, is discussed in Section 5.2. The TSMIP model defined by (1)-(21) is then solved for each of the nb bundle subproblems, yielding the optimal first-stage decisions of patient sequences, patient appointment times, and patient-nurse assignments

for each subproblem. Each of these nb first-stage variables are then used in the original model defined by (1)-(21) with the complete set of $|\Omega|$ scenarios (called the *master problem*), resulting in an objective value of z_b. The initial solution returned by the construction phase is the patient sequences of bundle b that yields the minimum objective value, i.e., $\arg\min_{b=1,2,\ldots,nb}\{z_b\}$.

The construction phase provides a feasible initial patient sequence, and hence an upper bound on the optimal objective value of the original TSMIP model. The details of the construction phase are given in Algorithm 1.

Algorithm 1 Scenario Bundling-Based Decomposition Algorithm: Construction Phase

Step 1: Initialization
Set nb and $|\Omega_b|$
Set the initial best objective $best\_obj = \infty$ and the initial best solution $best\_solution = \emptyset$
Step 2: Obtain the first-stage decisions of bundle subproblems
for $iter = 1, 2, \ldots, nb$ **do**
    Solve bundle sub-problems, obtain the first-stage variables: $\mathbf{u}_{iter}, \mathbf{t}_{iter}, \mathbf{x}_{iter}$
end for
Step 3 Obtain the initial solution for the master problem
for $iter = 1, 2, \ldots, nb$ **do**
    Solve the master problem with fixed $\mathbf{u}_{iter}, \mathbf{t}_{iter}, \mathbf{x}_{iter}$ and obtain z_{iter}
    if $z_{iter} < best\_obj$ **then**
        $best\_obj \leftarrow z_{iter}$, $best\_solution \leftarrow \mathbf{u}_{iter}$
    end if
end for

The improvement phase mainly uses two operators. The first of these is a local search operator, denoted by p_i, which swaps the i^{th} and $(i-1)^{st}$ patients in the sequence. The second is a diversification operator, denoted as $d_k p$, where the randomly chosen p^{th} patient is swapped with the $(p-k)^{th}$ patient in the sequence.

Following the initialization of the upper bound, best sequence, and iteration counters, the improvement phase solves the master problem by fixing the best sequence obtained from the construction phase, possibly yielding a better upper bound, as the nurse-patient assignments and/or patient appointment times may be different from those in the resulting solution of the construction phase.

The next step is to apply the local search operator, starting from the $(|N|+1)^{st}$ patient. We note here that for the first $|N|$ patients, swapping appointment times does not lead to any change in the sequence, as all of these patients are assigned to $t = 0$. The main reason behind enforcing the appointments of the first $|N|$ patients at $t = 0$ is that this helps with eliminating part of the variables and speeds up the solution process. Furthermore, this procedure complies with the actual practice in the OCC. In case the swap operator leads to a sequence that has been encountered before or the first $|N|$ patients cannot be assigned to $t = 0$ simultaneously (the latter occurs when the simultaneous assignment of these patients to nurses is infeasible), the solution is repaired or adjusted by an appropriate procedure explained in Section 4.2. If no repair or adjustment is needed, the master problem is solved with the new sequence, and the upper bound is updated whenever an improvement is encountered. Once the swaps of all adjacent patient pairs are considered in order, the algorithm checks whether there has been an improvement in the upper

bound throughout the iteration. If an improvement has been made during local search, the best sequence is updated and the local search restarts with the updated solution. If no improvement has been found, then the diversification procedure is initiated.

The diversification step starts by determining the k and p parameters of the $d_k p$ operator. If the current diversification iteration counter exceeds a threshold value inversely proportional to the value of k, we reset this counter and increase the value of k by one. Otherwise, we continue with the current value of k. This is to encourage exploitation of the search space in the earlier diversification iterations, and to avoid excessive iterations in the later stages. The p parameter is generated randomly, depending on the current value of k, the number of patients, and the number of nurses. Once these two parameters are determined, the diversification operator is applied on the best sequence. If the new sequence has been encountered before, the algorithm moves back to the start of local search with the current k. Otherwise, the master problem is solved with the current sequence and if any improvement is found, the objective value is updated, followed by a move back to the local search process with the current solution. Once local search is completed, the algorithm moves on to the next iteration of the diversification step.

A summary of the improvement phase is presented in Algorithm 2. The improvement algorithm may terminate when (i) the iteration number limit is reached, (ii) the time limit is reached, or (iii) if $k = |P|$ during the diversification step and the local search operator operates until the last patient. In the latter case, since $k = p = |P|$, there is no possibility of diversifying the sequence any more.

4.2. Performance Improvements for the SBBD Algorithm

The running time of the algorithm can be improved by (i) *repairing* the sequence of patients when a previously tested sequence has been obtained by the local search or diversification operator, and (ii) *adjusting* the sequence when the appointment time of at least one of the first $|N|$ patients cannot be set at $t = 0$.

To avoid running the master problem with a previously encountered sequence, the sequence is repaired. If the sequence is obtained after the local search operator p_i, then the value of i is incremented by 1 until a new sequence is found. When $i = |P|$ and no new sequence is found, then the algorithm proceeds to the diversification step. If the repeat sequence is obtained after the diversification operator $d_k p$, the local search operator p_i is called to find a new sequence. If no new sequence can be obtained by p_i until $i = |P|$, then in the diversification a different random patient p is selected. If all possible patients are attempted for swap under the fixed k, then the value of k is increased by 1 and the diversification step is repeated. A summary of the repair procedure is presented in Algorithm 3 in Appendix A of the Supplementary Material.

As we assume that the appointment times of the first $|N|$ patients in the sequence should be at $t = 0$, we adjust the sequence if, at any part of the algorithm, this condition cannot be maintained. To do so, the diversification operator $d_k p$ is called with $k = 1$ and $p = |N| + 1$, so that patients $|N|$ and $|N| + 1$ are swapped in the sequence. If the new sequence also does not satisfy the condition, we repeatedly call the

Algorithm 2 Scenario Bundling-Based Decomposition Algorithm-Improvement Phase

Step 1: Initialization
$UB = best\_obj$, $best\_seq = \{sequence : best\_obj\}$, $div\_seq = best\_seq$
Set the tested sequence set: $TS = \emptyset$
$iter = 1$, $diver\_iter = 1$, $k = 1$, $improved = false$
Set improvement iteration limit $AlgIterNum$
while (iter $\leq$AlgIterNum) **do**
    Step 2: Start improvement phase
    Solve the master problem with best_seq
    $TS \leftarrow TS \cup \{best\_seq\}$
    if $obj(iter) < best\_obj$ **then**
        $best\_obj \leftarrow obj(iter)$, $UB \leftarrow best\_obj$
    end if
    Step 3: Start local search
    for $i = |N| + 1$ to $|P|$ **do**
        $iter \leftarrow iter + 1$
        Apply swap operator p_i on div_seq to obtain new_seq
        if $new\_seq \in TS$ **then**
            Repair the sequence (See Algorithm 3)
        else if the first $|N|$ patients of new_seq cannot be assigned to $t = 0$ simultaneously **then**
            Adjust the sequence (See Algorithm 4)
        else
            $TS \leftarrow TS \cup \{new\_seq\}$
            Solve the master problem to obtain $obj(iter)$
            if $obj(iter) < best\_obj$ **then**
                $improved \leftarrow true$
                $best\_obj \leftarrow obj(iter)$, $UB \leftarrow best\_obj$
            end if
        end if
    end for
    if $improved = true$ **then**
        $best\_seq \leftarrow \{sequence : best\_obj\}$
        $div\_seq \leftarrow best\_seq$
        if $k \neq |P|$ **then**
            $k \leftarrow 1$, $diver\_iter \leftarrow 1$, $improved \leftarrow false$
            Restart the local search process
        else
            Terminate with $best\_seq$ as the sequence and $best\_obj$ as the objective value
        end if
    else if $k = |P|$ **then**
        Terminate with $best\_seq$ as the sequence and $best\_obj$ as the objective value
    else
        Step 4: Diversify the best sequence
        $iter \leftarrow iter + 1$
        if $diver\_iter > \left\lfloor \frac{|P|}{max\{2,k\}} \right\rfloor$ **then**
            $diver\_iter \leftarrow 1$
            $k \leftarrow k + 1$
        end if
        $p = U(\max\{k, |N| + 1\}, |P|)$
        Apply diversification operator $d_k p$ on $best\_seq$ to obtain div_seq
        $diver\_iter \leftarrow diver\_iter + 1$
        if $div\_seq \in TS$ **then**
            Restart the local search process
        else if the first $|N|$ patients of div_seq cannot be assigned to $t = 0$ simultaneously **then**
            Adjust the sequence (See Algorithm 4)
        else
            $TS \leftarrow TS \cup \{div\_seq\}$
            Solve the master problem to obtain $obj(iter)$
            if (obj(iter) < best_obj) **then**
                $best\_obj \leftarrow obj(iter)$, $UB = best\_obj$
                $improved \leftarrow true$
            end if
            Restart the local search process
        end if
    end if
end while

diversification operator by increasing p and k by 1 and performing the swap accordingly until the condition is satisfied. By doing so, the adjustment process attempts to change the order of the $|N|^{th}$ patient in the sequence with the nearest one with whom the swap would lead to the condition being satisfied. When such a sequence is found, k is reset to its original value before the adjustment step. A summary of this process is provided in Algorithm 4 in Appendix A of the Supplementary Material.

4.3. Improvements on the TSMIP Model

In addition to improvements in the algorithm, we also discuss the improvements we implement on the TSMIP model to decrease its running time, by means of setting tight big-M values and elimination of constraints and variables.

4.3.1. Setting the M values

Constraints (5), (15) and (18) in the TSMIP model involve big-M values, which can be tightened using problem-specific information.

Constraints (5) only apply if $u_{ij} = 1$, i.e., patient $i \in P$ precedes patient $j \in P$. Otherwise, the difference between the appointment times of the two patients can be at most $h - b$. Consequently, the M value for this constraint is set at $h - b$.

Constraints (15) become redundant if either patient $i \in P$ does not precede patient $j \in P$ ($u_{ij} = 0$) and/or patients $i \in P$ and $j \in P$ are not assigned to the same nurse $n \in N$ ($x_{in} + x_{jn} < 2$). When $M = h$, the constraint is still valid, as in the tightest case the treatment of patient $j \in P$ starts in the first slot and that of patient $i \in P$ starts at the last slot. This leads to the right-hand side value of h, which is still exceeded by the left-hand side.

Constraints (18) are redundant when patient $i \in P$ is not assigned to nurse $n \in N$, and the tightest case is when the pre-medication of $i \in P$ starts in the last slot, leading to a right-hand side value of $c_i(\omega) - M$. Applying a scenario-dependent M (denoted by $M(\omega)$) and setting it as $M(\omega) = \max_{k \in P}\{c_k(\omega)\}$ ensures the validity of the constraints.

4.3.2. Elimination of Constraints and Variables

The performance of the TSMIP model, when called from the decomposition algorithm, can be further improved by elimination of variables and constraints.

Each iteration of the heuristic algorithm uses a fixed patient sequence. The information gathered from the sequence may be used to eliminate some of the variables from the formulation. This in turn decreases the computation time.

As discussed in Section 4.2, the appointments of the first $|N|$ patients in the sequence are assigned to $t = 0$, and therefore $s = 1$. Since the sequence is fixed and the first $|N|$ patients in the sequence are known, we set $t_{i1} = 1$ for patients i in this sequence and $t_{i1} = 0$ for all other patients. Similarly, we also set the

$z^s_{in}(\omega) = 0$ for these patients for all $s > 1$. As the waiting time for these patients is zero, we eliminate the $w_i(\omega)$ variables.

For the $(|N| + 1)^{st}$ patient in the sequence, we use the information gathered from the acuity levels of the first $|N| + 1$ patients and the maximum nurse acuity levels r_n. As the $(|N| + 1)^{st}$ is the second patient of a nurse, we check whether this nurse can treat the two patients simultaneously or not. As the nurse-patient assignments are not known a priori, we perform this check for each possible nurse-patient assignment. If the nurse is able to treat the two patients simultaneously, the pre-medication of the $(|N|+1)^{st}$ patient can start immediately after the pre-medication of the previous patient. Since all pre-medication durations are constant and equal to a single slot, we set the t_{is} and $z^s_{in}(\omega)$ to zero for this patient-nurse pair and $s > 2$. If, on the other hand, the r_n level for this nurse is no sufficient to handle both patients simultaneously, we set the t_{is} and $z^s_{in}(\omega)$ variables for this patient-nurse pair to zero for $s = 1$ through $s = \min\{|S|, max\{\lceil (c_i(\omega) + b)/\sigma \rceil : i = 1, 2, \ldots, N\}\}$. In cases where nurses are more likely able to treat two patients simultaneously, as is the case in our experiments, this procedure can eliminate a significant number of variables.

The sequence information can also be used to eliminate the redundant constraints arising from the fixed sequencing (u_{ij}) variables. When $u_{ij} = 0$, constraints (5),(15) and (16) become redundant, and thus are eliminated from the model. The knowledge of the u_{ij} variables also eliminates the need to write constraints (5) for each patient pair. Instead, it is sufficient to consider two consecutive patients in the sequence for these constraints. Constraints (16) do not need to be written for the first patient in the sequence, as this patient will definitely start the treatment at the first slot and there will always be an available chair for this patient. These constraints also need not be written for the cases dummy patient is part of the pair (except when paired with patient $|P|$). Lastly, we do not need to write constraints (16) for patient pairs that are among the first $|C|$ patients in the sequence, as these patients will always find an available chair upon arrival. Furthermore, if their assigned nurse is available, these patients can also start their treatment directly upon their arrival.

5. Experimental Study

We conduct numerical experiments using a data set of planned and actual treatment durations collected from an OCC at Hacettepe University Oncology Hospital in Ankara, Turkey, during the years 2017 and 2018 (Demir et al., 2021). Based on extensive experiments, we first determine the appropriate size of scenario bundles for the implementation of our algorithm. After fixing the size of scenario bundles, we assess the average optimality gap considering the solutions of our algorithm and those found by CPLEX. Next, we compare the algorithm with commonly used scheduling heuristics in the literature. We then assess the benefit of considering uncertainty in infusion durations in the model. Finally, we perform sensitivity analysis on various model parameters to generate managerial insights related to chemotherapy scheduling.

Our algorithm is implemented by Microsoft Visual C++ 2019 using CPLEX 12.9 Concert Technology.

Table 2: Actual infusion duration intervals observed for each patient group and group occurrence probabilities

Patient group	**Probability of occurrence**	**Infusion duration (mins)**
1	26.96%	[16,44]
2	7.85%	[29,80]
3	33.33%	[74,132]
4	31.86%	[125,217]

Table 3: Characteristics of the patients in the instance set selected

Patient Index	1	2	3	4	5	6	7
Patient Group	4	1	4	3	1	3	3
Acuity Level	2	3	1	1	2	1	1

The experiments are performed on an AMD Ryzen 7 Pro 2700X computer with eight-core processor running at 3.60 GHz and 64GB RAM. We set the upper limit on the run time as 3 hours and number of SBBD algorithm iterations as 100 in each experiment.

5.1. Generating Problem Instances

For each computational experiment, we create an instance set consisting of 10 instances. We generate different instances for each instance set by sampling different infusion duration values from our data set. Unless otherwise stated, the number of scenarios, $|\Omega|$, is equal to 100. In other words, we sample 100 infusion duration values for a patient from our data set. Patients are categorized into four groups according to the length of treatment durations estimated by the head nurse of the OCC. The probability that a patient belongs to each particular group and the actual infusion duration ranges for each group are shown in Table 2. Further details of the data set are explained in Demir et al. (2021).

We sample an infusion duration for a patient for a given scenario realization based on the following approach: We first generate the patient group based on the group occurrence probabilities given in Table 2. Then, we randomly generate an infusion duration from the interval associated with the selected patient group. The use of uniform distribution to generate the infusion durations is justified in Demir et al. (2021). Infusion duration sampling procedure is applied for all patients independently in each scenario realization. Then, the steps for scenario generation are conducted repeatedly to produce all scenarios in a problem instance. The instances in an instance set differ from each other based on the set of infusion duration scenarios. The remaining model parameter values are kept the same in all instances in a given instance set.

The relevant literature shows that the acuity scales developed for infusion clinics differ significantly among each other (Fesler & Toms, 2020).The number of levels used in an acuity scale is generally set to either 3 or 5 (Liang & Turkcan, 2016; Alvarado & Ntaimo, 2018). Higher values imply greater level of nurse attention needed. Following the approach by Alvarado & Ntaimo (2018), we choose an acuity scale from three possible levels. The acuity level of a patient (a_i, $i \in P$) can take a value of 1, 2, or 3 based on probabilities of 0.7, 0.2, and 0.1, respectively. A higher acuity level does not always imply longer treatment durations. The treatment

of a patient may require close attention of a nurse, but may also take a small amount of time. Therefore, as also is the case in Liang & Turkcan (2016) and Alvarado & Ntaimo (2018), our infusion duration sampling procedure is independent from the acuity levels of patients. We set the highest level of patient acuity that a nurse can manage (r_n, where $n \in N$) as 2 or 3 with equal probabilities. Furthermore, we ensure that the desirable upper limit on the total acuity levels of the patients who are simultaneously treated by a nurse (m_n, where $n \in N$) is related to the skill levels of nurses. In particular, the following relationship is maintained in our experiments: $m_n = r_n + 2$, where $n \in N$ (i.e., m_n can be 4 or 5).

We create half-day schedules in line with the practice in the OCC we study. This means the morning and afternoon shift schedules are created independently. We set the planned shift length (h) as 240 minutes, as the OCC is open for 8 hours a day. Each slot length (σ) is equal to 15 minutes, which implies that there are in total $|S| = 16$ slots in a shift schedule.

The data collected from the OCC shows that a 95% confidence interval for premedication duration for all patients is $[14.61, 16.09]$ (Demir et al., 2021). Due to the narrow width of this interval, we assume that premedication duration for any patient is constant and equal to 1 slot (i.e. $b = 15$).

We consider the setting where sets of chairs are created in the clinic and each set is located at a separate room. Since a nurse can monitor multiple patients simultaneously only if they are treated in the same room, nurses are pre-assigned to the rooms by the management. We focus on the scheduling for a specific room. In particular, we create instances for the case where there are $|P|$=7 patients, $|C|$=4 chairs, and $|N|$=2 nurses.

Table 3 shows the acuity and group values of the patients in the instance set. We consider this set as a typical set (except in Section 5.6), as the distribution of patient group and acuity level values are consistent with the given frequencies. We set r_n as 2 and 3, and m_n as 4 and 5 for the first and second nurse, respectively. In the OCC considered, nurse overtime was given higher importance, so we set $\lambda = 0.4$. Finally, we set $\beta = 0.5$ based on our preliminary experiments.

5.2. Determining the Bundle Size

In this section, we explain the procedure we follow to find the best bundle size used in the first phase of our algorithm while searching for the initial solution. We fix the bundle size at a level that yields a good compromise for the trade-off between the optimality gap and computational time. In the subsequent sections, we use this bundle size in further experiments.

Bundle size is an important parameter, as it directly affects both the initial solution quality and run time spent to obtain an initial solution. Letting $|\Omega_b|$ represent the size of a bundle, we first test $|\Omega_b|$={2,3,4,5,6} values in our experiments related to the construction phase of the algorithm. In an experiment, the number of bundles created, which is denoted by nb, is equal to the following expression:

$$nb = \left\lceil \frac{|\Omega|}{|\Omega_b|} \right\rceil \tag{22}$$

For example, when $|\Omega_b| = 4$, there are $nb = 25$ bundles each including 4 scenarios, as the total number of

Table 4: The run time (in seconds) spent to find the initial solution for each value of $|\Omega_b|$

	$\|\Omega_b\|$				
Instance #	2	3	4	5	6
1	937.04	1209.44	2303.97	5876.17	10977.00
2	833.80	1188.91	1866.92	4922.86	7703.63
3	844.64	1068.68	2005.43	4619.55	8178.25
4	859.73	1241.53	2805.10	5677.50	10544.10
5	870.26	1203.10	2514.50	5370.69	8413.99
6	905.50	1337.48	3125.66	6834.53	10959.20
7	920.83	1165.22	2135.74	4909.34	7419.57
8	802.75	1214.87	2145.57	4411.89	6488.70
9	950.84	1077.02	1979.01	4458.34	8167.68
10	1462.47	1101.70	2166.41	5673.44	8893.79
Average	938.79	1180.80	2304.83	5275.43	8774.59

scenarios is equal to 100 in an instance. However, when $|\Omega|/|\Omega_b|$ is not an integer, the size of the last bundle is set a value less than $|\Omega_b|$. For instance, when $|\Omega_b| = 6$, $nb = 17$ bundles are created, where 16 of them include 6 scenarios, while the remaining one includes 4 scenarios.

Table 4 shows the run time spent to find the initial solution of our algorithm through solving nb subproblems associated with each $|\Omega_b|$ value. The average over 10 instances indicates that the run time significantly increases as $|\Omega_b|$ increases. While the average run time is 15.65 minutes when $|\Omega_b| = 2$, it reaches 146.24 minutes for $|\Omega_b| = 6$. The algorithm may find an initial solution after spending more than 3 hours when $|\Omega_b| = 6$. Since we set the total time limit as 3 hours for generating the final solution of the algorithm, we eliminate $|\Omega_b| = 6$ from further consideration. The average objective values over ten instances for the initial solutions associated with $|\Omega_b|$=$\{2, 3, 4, 5\}$ are found as $\{10.74, 10.09, 9.77, 9.69\}$, respectively. The results show that the average objective value improves as the value of $|\Omega_b|$ increases.

Since solving bundle subproblems with smaller $|\Omega_b|$ values requires less time during the construction phase of the algorithm, a larger amount of time remains for the implementation of the improvement phase. On the other hand, even though solving subproblems with larger $|\Omega_b|$ values is more challenging, it is more likely to obtain promising initial solutions in such cases. In other words, there is a trade-off between (i) spending extensive amount of time to find good initial solutions followed by a limited effort for improvement, and (ii) finding a quick initial solution and improving it by spending a large number of improvement iterations.

To determine the best value of bundle size among the remaining $|\Omega_b|$ values, we solve the instances by running our algorithm to the end. Table 5 shows the computational time needed to obtain the solutions and objective values associated with them. When $|\Omega_b|$ increases from 2 to 3, the objective value improves by more than 2%. The best result is obtained when $|\Omega_b| = 3$, while similar objective values are found when $|\Omega_b|$ is larger. However, the computational time needed to find the solution increases significantly when $|\Omega_b| > 3$. Therefore, we set $|\Omega_b| = 3$ in our remaining experiments.

Table 5: Run time of the algorithm (in seconds) and the objective values for different values of $|\Omega_b|$

	$\|\Omega_b\| = 2$		$\|\Omega_b\| = 3$		$\|\Omega_b\| = 4$		$\|\Omega_b\| = 5$	
Instance #	**Time**	**Objective**	**Time**	**Objective**	**Time**	**Objective**	**Time**	**Objective**
1	10800.00	11.52	10800.00	10.96	10800.00	10.96	10800.00	11.19
2	9533.16	8.62	9587.92	8.62	10800.00	8.62	10800.00	8.62
3	10605.30	8.53	7750.63	8.53	10800.00	8.93	10800.00	8.53
4	9535.87	10.50	10214.30	10.50	10800.00	10.52	10800.00	10.50
5	10800.00	9.48	10800.00	9.75	10800.00	9.51	10800.00	9.51
6	10529.70	11.38	10089.20	11.38	10800.00	11.38	10800.00	11.38
7	9794.73	9.74	9125.04	9.74	9643.46	9.74	10800.00	9.74
8	10800.00	10.59	8737.97	8.79	8782.05	8.79	10800.00	8.79
9	9067.15	9.07	9563.38	9.07	9814.51	9.06	10800.00	9.07
10	7241.02	8.93	8968.94	8.93	10310.30	9.09	10800.00	9.09
Average	9870.69	9.84	9563.74	9.63	10335.03	9.66	10800.00	9.64

Table 6: Percentage improvement over the best solution without elimination, number of improvement iterations, and computation time for the algorithm with variable and constraint elimination

Instance #	**Gap (%)**	**Improvement iterations**	**Run time (sec)**
1	0.00	68	10800.00
2	1.40	100	9587.92
3	4.07	79	7750.63
4	13.18	79	10214.30
5	0.20	67	10800.00
6	2.38	100	10089.20
7	12.55	100	9125.04
8	3.94	79	8737.97
9	1.18	84	9563.38
10	3.51	100	8968.94
Average	4.24	85.60	9563.74

5.3. Effect of Variable Elimination and Constraint Reduction on Algorithm Performance

As mentioned in Section 4.3.2, knowing the patient sequences enables the elimination of some variables and constraints; we use this advantage to expedite the algorithm. Table 6 shows the objective function improvement, number of improvement iterations, and the computational time with variable and constraint elimination. Our findings show that reducing the size of the model by constraint and variable elimination significantly enhances the running time of the algorithm. Without elimination, the algorithm is only able to find an initial solution (which corresponds to 34 construction iterations) and is not able to compute any improvement iterations within 3 hours. With elimination, the algorithm is able to run until the iteration count limit (instances 2, 6, 7, and 10) or until all the required number of diversification schemes are performed (instances 3, 4, and 8). On average, variable and constraint elimination improves the solution quality by 4.24%, and the improvement rate exceeds 10% in two instances.

5.4. Comparison with Optimal Solutions

To assess the solution quality of our algorithm with regard to optimal solutions, we compare our results with the optimal solutions for instances with the largest number of scenarios that can be solved to optimality by CPLEX. Keeping all settings the same as in Section 5.1, CPLEX can solve instances with up to 13 scenarios. Consequently, the instances in this section consist of 13 scenarios.

Table 7: Average gap between the optimal solutions and the solutions found by the SBBD algorithm, and the run times (in seconds) for CPLEX and SBBD algorithm

	Objective Value			Run Time	
Instance #	**CPLEX**	**SBBD**	**Gap (%)**	**CPLEX**	**SBBD**
1	8.34	8.34	0.00	9347.47	258.84
2	7.88	7.88	0.00	4312.00	375.48
3	5.90	5.97	1.14	4069.00	319.10
4	8.16	8.92	8.54	9087.83	499.10
5	9.10	9.10	0.00	10619.48	379.08
6	4.40	4.40	0.00	8974.74	296.97
7	7.15	7.15	0.00	8801.66	277.72
8	8.97	9.93	9.62	10524.32	628.73
9	5.27	5.27	0.00	4619.06	269.13
10	7.74	8.37	7.53	9701.17	282.84
Average	7.29	7.53	2.68	8005.67	358.70

Table 8: Objective values for the initial solutions of the SBBD, and the percentage gap between these solutions and those of the sequencing heuristics

	Objective Value	Gap (%)				
Instance #	**SBBD**	**LPT**	**SPT**	**VAR**	**CoVar**	**Random**
1	10.96	33.45	80.18	78.91	23.40	57.21
2	8.75	36.90	83.31	81.93	30.26	77.68
3	8.89	37.17	82.34	82.09	37.15	38.26
4	12.10	22.80	76.22	76.72	6.74	65.43
5	9.77	35.04	82.20	82.05	23.22	61.02
6	11.66	32.32	77.79	78.26	2.52	75.07
7	11.14	25.44	78.80	77.09	28.04	74.37
8	9.15	29.23	81.45	81.36	76.91	80.14
9	9.18	34.60	81.03	80.78	25.96	82.28
10	9.26	34.21	81.92	81.33	22.92	79.45
Average	10.09	32.12	80.53	80.05	27.72	69.09

Based on the average gap values and computation times in Table 7, CPLEX spends an average time of more than two hours to find the optimal solutions, whereas the proposed algorithm spends approximately six minutes. The average optimality gap for the solutions found by the algorithm is 2.68%, while for six out of ten instances the optimal solution is found. The results show that the proposed algorithm finds good quality solutions in a reasonable amount of time.

5.5. Comparison with Sequencing Heuristics

To justify our claim in Section 4.1 that the construction phase of the SBBD approach is an appropriate method to identify promising initial solutions, we compare our initial patient sequences with those found by the sequencing heuristics that are commonly applied in the literature and can be easily used in practice. As benchmarks, we form the initial sequence using four different sequencing heuristics based on the infusion durations: (i) decreasing mean (LPT), (ii) increasing mean (SPT), (iii) increasing variance of (VAR), and (iv) increasing coefficient of variation (CoVar). As the fifth benchmark, we also set the initial sequence randomly.

Table 9: Objective values for the final solutions of the SBBD, and the percentage gap between these solutions and those of the sequencing heuristics

	Objective Value	**Gap (%)**				
Instance #	**SBBD**	**LPT**	**SPT**	**VAR**	**CoVar**	**Random**
1	10.96	17.54	77.93	76.82	0.00	23.55
2	8.62	1.40	81.91	80.44	23.57	58.72
3	8.53	6.55	81.63	81.05	18.75	24.43
4	10.50	0.19	77.19	77.28	0.00	69.98
5	9.75	12.65	80.16	80.41	0.00	32.22
6	11.38	19.13	75.85	75.69	4.84	31.82
7	9.74	16.87	79.84	77.94	22.83	30.18
8	8.79	13.76	80.04	79.90	1.96	79.09
9	9.07	1.99	79.47	79.35	3.27	0.00
10	8.93	0.00	80.41	79.86	1.74	29.25
Average	9.63	9.01	79.44	78.87	7.70	37.92

In Table 8, we show the objective values associated with the initial solutions of the SBBD algorithm for each instance, and the percentage gap between the initial solutions of the SBBD and heuristics. The results reveal that the SBBD always provides the best initial solutions. Among the five benchmark approaches, CoVar performs the best. However, even with this heuristic, the average gap from the SBBD solutions is almost 28%. These findings suggest that the simple heuristics cannot provide good initial solutions, and the construction phase of the SBBD perform quite well in comparison to other heuristics.

We next investigate how essential it is to start with a good initial solution while implementing the SBBD algorithm. For this purpose, we test the improvement phase of the algorithm to reach the final solution by first starting with the initial sequence obtained by the construction phase of the SBBD, and then by the simple heuristics. In Table 9, we provide the objective values associated with the final solutions of the SBBD algorithm for each instance, and the percentage gaps between the final solutions of the SBBD and sequencing heuristics. Based on these results, the SBBD always yields the best solutions compared to the benchmark heuristics. Among the simple heuristics, CoVar performs the best, yielding a 7.7% gap with respect to the SBBD. LPT, Random, VAR and SPT sequences follow CoVar sequence, respectively, in terms of final solution quality. For SPT and VAR, we observe that the average gaps are very large, which are around 79%. These results show that it is important to start with the initial sequence finding mechanism of the SBBD algorithm.

5.6. Estimating the Value of Stochastic Solution

The value of stochastic solution (VSS) measures the added value of incorporating stochasticity of the problem environment in the modeling approach. We estimate the VSS using five different instance sets by varying the patient compositions across sets. In particular, we create new instance sets by sampling new values for acuity levels, patient groups, and the highest acuity level that a nurse can manage (provided in Tables 10 and 11). We first solve each instance using the SBBD algorithm to find the TSMIP solution. We then obtain the solutions of the mean value problem, where the random infusion durations in TSMIP are

Table 10: Characteristics of the patients considered in the experiments for the VSS calculations in each instance set

	Patient Index	1	2	3	4	5	6	7
Set 1	**Patient Group**	1	2	3	4	1	4	4
	Acuity Level	1	1	2	1	1	2	3
Set 2	**Patient Group**	4	3	3	1	3	1	3
	Acuity Level	2	3	1	1	3	1	1
Set 3	**Patient Group**	4	1	4	1	3	1	4
	Acuity Level	3	1	1	3	1	2	2
Set 4	**Patient Group**	4	1	4	3	1	3	3
	Acuity Level	2	3	1	1	2	1	1
Set 5	**Patient Group**	4	3	1	3	2	1	4
	Acuity Level	2	2	1	2	1	1	1

Table 11: r_n values used in the experiments for the VSS calculations in each instance set

Set #	$n = 1$	$n = 2$
1	2	3
2	2	3
3	3	3
4	2	3
5	2	2

replaced by the average values. Let z^{TSMIP} represent the evaluation of the objective function (1) for the TSMIP solution, and z^{MV} be the objective function (1) value calculated based on the mean value problem solution. Then, we estimate the VSS by calculating z^{MV}-z^{TSMIP}.

In Table 12, we report the averages of z^{TSMIP} and z^{MV} over 10 instances for each instance set, as well as the relative VSS, which shows the percentage improvement in the objective values. The table shows that the average relative VSS values range between 41.61% and 68.72%, which points to substantial gains by taking into account the inherent stochasticity of the infusion durations.

5.7. Sensitivity Analysis

In this section, we conduct sensitivity analyses to examine the impact of changes in some critical parameters to the performance measures. In particular, we test various combinations of weight values in the objective function and change the number of nurses and chairs in our experiments. Furthermore, we investigate how waiting time and nurse overtime are affected due to the consideration of instantaneous nurse workload limits while creating schedules.

Table 12: Relative percentage VSS for different instance sets

Instance Set #	**Average z^{TSMIP}**	**Average z^{MV}**	**Relative VSS %**
1	12.20	21.05	41.61
2	5.97	14.62	57.40
3	6.64	14.79	52.35
4	9.63	18.80	47.99
5	2.80	11.24	68.72

Table 13: Characteristics of the patients considered while analyzing sensitivity to penalty coefficients

Patient Index	1	2	3	4	5	6	7
Patient Group	1	4	3	3	2	3	4
Acuity Level	3	3	3	3	3	2	3

5.7.1. Sensitivity to penalty coefficients

We analyze the sensitivity of model solutions to the changes in the penalty coefficients in the objective function. To generate valuable insights, we methodically determine the combinations of penalty values. Through the parameter that controls the trade-off between waiting time and overtime, we consider the following cases: (i) waiting time is less important than overtime ($\lambda = 0.2$), (ii) waiting time and overtime are equally important ($\lambda = 0.5$), (iii) waiting time is more important than overtime ($\lambda = 0.8$). We also focus on the cases where the sensitivity to excess acuity is low ($\beta = 0.5$), medium ($\beta = 5$), or high ($\beta = 15$). Due to the direct relationship between nurse workload and the quality of care, we can assume that higher β values represent the setting where care quality level is also higher. With three potential values of each of λ and β, we test 9 different combinations of (λ, β) values in total. While creating an instance set, the remaining model parameter values are kept at their original levels as in Section 5.1, except the following ones: (i) $r_n = 3$ for both nurses, (ii) patient group and acuity levels are given in Table 13. The new instance set is selected to clearly illustrate the impact of penalty coefficients.

Table 14 shows the average objective, patient waiting time, nurse overtime and excess acuity values over 10 instances for each combination of (λ, β). As λ increases while β is kept constant, the waiting time decreases and nurse overtime increases significantly, as expected. Waiting time drops faster as λ increases independent of the value of β. The sensitivity of overtime to λ is particularly high when β value is high. For example, average overtime value increases from 28 minutes to 72 minutes when β equals 15 and λ is increased from 0.5 to 0.8. This implies that it becomes very difficult for the OCC manager to limit overtime when both patient satisfaction and the quality of care are given high importance in the clinic.

As expected, increasing β while keeping λ constant yields lower levels of excess acuity. However, higher values of overtime are observed in such a case. This result clearly shows the trade-off between excess acuity and overtime. In other words, this observation suggests that greater care quality can be achieved at the expense of overtime. This is reasonable because some appointments may need to be delayed to prevent excessive instantaneous nurse workload. Starting treatments later in the day may then lead to increased overtime.

5.7.2. Impact of the Number of Nurses and Chairs

We evaluate the sensitivity of model solutions to the number of chairs and nurses. Since we consider a chair set that can be simultaneously monitored by a nurse in the same room, we test only $|C| = \{3, 4, 5, 6\}$. Accordingly, the appropriate number of nurses responsible for a chair set could be $|N| = \{1, 2, 3\}$. We test all combinations of $|C|$ and $|N|$, but exclude unrealistic ones. In particular, we ignore the cases with: (i)

Table 14: Sensitivity of average objective value, patient waiting time (in minutes), nurse overtime (in minutes) and excess acuity value to λ and β

	Objective Value	**Waiting Time**	**Overtime**	**Excess Acuity**
λ=0.2, β=0.5	15.73	15.56	13.57	3.53
λ=0.2, β=5	27.51	20.85	15.97	2.11
λ=0.2, β=15	48.32	18.75	17.33	2.05
λ=0.5, β=0.5	14.77	7.47	18.40	3.67
λ=0.5, β=5	25.96	5.43	26.46	2.00
λ=0.5, β=15	46.30	6.24	28.16	1.94
λ=0.8, β=0.5	8.08	1.10	28.76	2.91
λ=0.8, β=5	17.40	0.74	36.69	1.89
λ=0.8, β=15	30.01	0.89	71.50	1.00

Table 15: Characteristics of the patients considered while analyzing sensitivity to the number of chairs and nurses, and nurse workload limits

Patient Index	1	2	3	4	5	6	7
Patient Group	1	4	1	4	3	3	3
Acuity Level	3	3	3	3	3	3	1

one nurse with six chairs, (ii) two nurses with three chairs, (iii) three nurses with three or four chairs. We set $\lambda = 0.2$ and $\beta = 5$ in our experiments. Other than the following ones, the remaining model parameter values are the same as the original levels given in Section 5.1: (i) $r_n = 3$ for each nurse, (ii) patient group and acuity levels provided in Table 15 are used.

Table 16 reports average patient waiting time, nurse overtime, nurse excess acuity and computation times with respect to different combinations of nurse and chair numbers. The average is calculated over all instances in the instance set. The results show that increasing the number of nurses leads to improvement in waiting time (before it is equal to 0) and excess acuity. The positive impact into waiting time is a natural consequence of the increase in capacity. Greater number of nurses also makes balancing the nurse workload easier and hence excess acuity decreases. Similarly, nurse overtime can be better controlled when there are higher number of nurses. However, we observe an exceptional case with five chairs; increasing number of nurses from one to two causes an increase in nurse overtime. This result is also intuitive due to the trade-off between excess acuity level and nurse overtime (see Section 5.7.1). Both measures may not improve at the same time especially when overtime values are very small.

When only one nurse is responsible for all patients (i.e. nurse to patient ratio is 1/7), then either overtime or excess acuity value may become substantial. For example, the excess acuity is equal to 7 for the case with four chairs. Since $m_n = 5$ for each nurse, excess acuity value of 7 implies that the instantaneous workload of a nurse is 140% greater than his/her regular workload. The lower limit on the nurse to patient ratio is generally recommended as 1:4 (i.e., one nurse can monitor up to 4 patients at the same time) in the literature (Castaing et al., 2016; Demir et al., 2021). In our experiments, when $|N| = 2$ (i.e nurse to patient ratio is 2:7), waiting time and overtime amount appear to be at reasonable values. However, the smallest average excess acuity is found as 1.41, which means that the instantaneous workload of a nurse is 28% greater

Table 16: Average patient waiting time (in minutes), nurse overtime (in minutes), nurse excess acuity levels, and computation times (in seconds) for various nurse-chair combinations. NS represents the unrealistic cases that are not solved

	Waiting Time			**Overtime**			**Excess Acuity**			**Run Time**		
	$\|N\|=1$	$\|N\|=2$	$\|N\|=3$	$\|N\|=1$	$\|N\|=2$	$\|N\|=3$	$\|N\|=1$	$\|N\|=2$	$\|N\|=3$	$\|N\|=1$	$\|N\|=2$	$\|N\|=3$
$\|C\|=3$	17.68	NS	NS	90.95	NS	NS	4.00	NS	NS	6541.90	NS	NS
$\|C\|=4$	17.49	14.52	NS	18.45	14.92	NS	7.00	1.41	NS	5140.66	10800.00	NS
$\|C\|=5$	0.71	0.00	0.00	0.39	0.95	0.00	8.04	2.20	1.00	3575.83	10800.00	10800.00
$\|C\|=6$	NS	0.00	0.00	NS	1.74	0.00	NS	1.77	1.00	NS	10800.00	10800.00

than his/her regular workload. This shows that 1:4 nurse to patient ratio may lead to low quality of care. Furthermore, specifying a patient to nurse ratio for OCCs is not a proper approach; acuity-based patient to nurse assignments must be made instead.

Table 16 also shows that increasing the number of chairs positively affects waiting time and overtime. On the other hand, nurse excess acuity is not directly related to the number of chairs in the OCC. Its value varies mainly due to the trade-off between overtime and excess acuity. This result also signifies the importance of the number of nurses with regard to the provision of high-quality care. However, since all three measures must be considered while providing services to patients, it is critically important to control the value of both resources simultaneously.

Note that increasing the number of nurses significantly worsens the computational performance. This is reasonable because it results in larger number of binary variables in both the first and second stage of the model. The time limit of the algorithm is reached even with two nurses. On the other hand, an increase in the number of chairs has positive impact into the computational performance, as solving the problem becomes easier.

5.7.3. Impact of Acuity Consideration

We show how the consideration of acuity while scheduling appointments impact patient waiting time and nurse overtime. To this end, we set m_n to a large enough value (i.e., $m_n = 10$) such that the constraints related to excess acuity (i.e., constraints (17)) become redundant. Since the excess acuity would never be observed in the resulting setting, the decision is given only with regard to waiting time and overtime. We compare those solutions with the solutions obtained for the setting where excess acuity must be considered (i.e., $m_n = 5$). We set $\lambda = 0.2$, $\beta = 5$, $r_n = 3$ for each nurse in our experiments. The groups and acuity levels of the patients are provided in Table 15. The remaining parameter values in the instance set are kept at their original levels that are given in Section 5.1.

Table 17 compares the solutions of the cases where $m_n = 5$ and $m_n = 10$ with respect to average patient waiting time and nurse overtime for each instance. The results show that if patient acuity levels are not considered while scheduling appointments, nurse overtime would be significantly underestimated, whereas patient waiting time estimation does not significantly change. Consequently, patient acuity levels must be taken into account to better estimate overtime costs and extra work time amount of nurses while generating schedules.

Table 17: Average patient waiting time and nurse overtime values for each instance when $m_n = 5$ and $m_n = 10$

	Waiting time		**Overtime**	
Instance #	$m_n = 5$	$m_n = 10$	$m_n = 5$	$m_n = 10$
1	23.10	12.75	11.02	8.01
2	16.50	22.65	14.07	7.63
3	16.50	21.60	11.99	6.74
4	12.15	12.30	15.83	9.54
5	5.70	5.85	13.18	11.12
6	11.70	16.05	15.57	10.08
7	20.10	9.00	16.05	10.67
8	11.70	13.50	13.53	8.79
9	13.35	11.55	17.78	12.82
10	14.40	15.60	20.19	10.34
Average	14.52	14.09	14.92	9.57

6. Conclusions

In this article, we study the problem of scheduling chemotherapy appointments and assignment of patients to nurses by considering patient acuity levels and chair availability under uncertain infusion durations. We formulate a TSMIP model which minimizes the expected weighted sum of excess acuity over the target nurse capacity, patient waiting time and nurse overtime. We propose a novel algorithm called SBBD, which integrate scenario bundling approach with an improvement heuristic in a two-phase approach. We compare SBBD with CPLEX and practically relevant heuristics from the scheduling literature. We assess the sensitivity of the schedules to the weights of excess acuity, waiting time and overtime. We analyze the impact of acuity consideration and the number of patients and nurses to the model solutions. We also estimate the value of stochastic solution. The most essential findings of the study is summarized below.

The SBBD algorithm, with the proposed constraint and variable elimination strategies, provides solutions close to the optimal. The scenario bundling approach used in the construction phase of the SBBD algorithm significantly outperforms well-known and practically relevant sequencing heuristics in terms of the quality of initial solutions. The experiments also show that it is critical to start with a good initial solution before the improvement phase is implemented.

In the literature, it is generally suggested that nurses can be responsible for up to four patients at a given time during treatment. However, our results show that this ratio may lead to a low quality of care due to excess acuity. Consequently, the OCC managers should determine the number of patients assigned to each nurse based on acuity levels rather than general nurse-to-patient ratios.

The estimated VSS shows a significant benefit in the consideration of uncertainty in infusion durations while creating chemotherapy appointment schedules. The results of sensitivity analysis point out that controlling overtime becomes difficult when both patient satisfaction (measured by waiting time) and quality of care (measured by excess acuity) are given high importance. Furthermore, there exists a clear trade-off between nurse overtime and excess acuity. To improve any of the three measures, the number of nurses can be increased. On the other hand, the excess acuity cannot be improved by increasing the number of chairs.

In this article, we randomly select and group scenarios in a scenario bundle in the construction phase of the SBBD algorithm. A useful extension of this study is to test different scenario bundling strategies based on different scenario similarity measures, and analyze their impact into the SBBD algorithm performance. Besides, in this study, we assume that a practical chair assignment policy is in effect in an OCC, and hence propose a two-stage SP model. As a future work, this assumption can be relaxed to formulate a multi-stage SP model that would be appropriate for the scheduling of appointments in any OCC.

References

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

Alvarado, M., & Ntaimo, L. (2018). Chemotherapy appointment scheduling under uncertainty using mean-risk stochastic integer programming. *Health Care Management Science*, *21*, 87–104.

Benzaid, M., Lahrichi, N., & Rousseau, L.-M. (2020). Chemotherapy appointment scheduling and daily outpatient–nurse assignment. *Health Care Management Science*, *23*, 34–50.

Castaing, J., Cohn, A., Denton, B. T., & Weizer, A. (2016). A stochastic programming approach to reduce patient wait times and overtime in an outpatient infusion center. *IIE Transactions on Healthcare Systems Engineering*, *6*, 111–125.

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

Crainic, T. G., Hewitt, M., & Rei, W. (2014). Scenario grouping in a progressive hedging-based meta-heuristic for stochastic network design. *Computers & Operations Research*, *43*, 90–99.

Demir, N. B., Gul, S., & Çelik, M. (2021). A stochastic programming approach for chemotherapy appointment scheduling. *Naval Research Logistics (NRL)*, *68*, 112–133.

Escudero, L. F., Garin, M. A., Perez, G., & Unzueta, A. (2013). Scenario cluster decomposition of the lagrangian dual in two-stage stochastic mixed 0–1 optimization. *Computers & Operations Research*, *40*, 362–377.

Fesler, S. M., & Toms, R. (2020). Infusion center outpatient acuity: an integrative review of the literature. *Journal of Pediatric Nursing*, *55*, 184–191.

Gade, D., Hackebeil, G., Ryan, S. M., Watson, J.-P., Wets, R. J., & Woodruff, D. L. (2016). Obtaining lower bounds from the progressive hedging algorithm for stochastic mixed-integer programs. *Mathematical Programming Series B*, *157*, 47–67.

Gul, S. (2021). Chemotherapy appointment scheduling under uncertainty by considering workload balance among nurses. *Pamukkale University Journal of Engineering Sciences*, *27*, 570–578.

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

Hesaraki, A. F., Dellaert, N. P., & de Kok, T. (2019). Generating outpatient chemotherapy appointment templates with balanced flowtime and makespan. *European Journal of Operational Research*, *275*, 304–318.

Hesaraki, A. F., Dellaert, N. P., & de Kok, T. (2020). Integrating nurse assignment in outpatient chemotherapy appointment scheduling. *OR Spectrum*, *42*, 935–963.

Heshmat, M., Nakata, K., & Eltawil, A. (2018). Solving the patient appointment scheduling problem in outpatient chemotherapy clinics using clustering and mathematical programming. *Computers & Industrial Engineering*, *124*, 347–358.

Hooshangi-Tabrizi, P., Contreras, I., Bhuiyan, N., & Batist, G. (2020). Improving patient-care services at an oncology clinic using a flexible and adaptive scheduling procedure. *Expert Systems with Applications*, *150*, 113267.

Jennings, B. M. (2008). Patient acuity. In R. G. Hughes (Ed.), *Patient Safety and Quality: An Evidence-Based Handbook for Nurses* chapter 23. Rockville (MD): Agency for Healthcare Research and Quality (US).

Jiang, X., Bai, R., Wallace, S. W., Kendall, G., & Landa-Silva, D. (2021). Soft clustering-based scenario bundling for a progressive hedging heuristic in stochastic service network design. *Computers & Operations Research*, *128*, 105182.

Kallen, M. A., Terrell, J. A., Lewis-Patterson, P., & Hwang, J. P. (2012). Improving wait time for chemotherapy in an outpatient clinic at a comprehensive cancer center. *Journal of Oncology Practice*, *8*, e1–e7.

Katayama, H., Tabata, M., Kubo, T., Kiura, K., Matsuoka, J., & Maeda, Y. (2021). Demand for weekend oupatient chemotherapy among patients with cancer in japan. *Supportive Care in Cancer*, *29*, 1287–1291.

Lamè, G., Jouini, O., & Cardinal, J. S.-L. (2016). Outpatient chemotherapy planning: A literature review with insights from a case study. *IIE Transactions on Healthcare Systems Engineering*, *6*, 127–139.

Liang, B., & Turkcan, A. (2016). Acuity-based nurse assignment and patient scheduling in oncology clinics. *Health Care Management Science*, *19*, 207–226.

Mandelbaum, A., Momcilovic, P., Trichakis, N., Kadish, S., Leib, R., & Bunnell, C. A. (2020). Data-driven appointment-scheduling under uncertainty: The case of an infusion unit in a cancer center. *Management Science*, *66*, 243–270.

Oliveira, W., Sagastizabal, C., & Scheimberg, S. (2011). Inexact bundle methods for two-stage stochastic programming. *SIAM Journal on Optimization*, *21*, 517–544.

Rodriguez, A. L., Jackson, H. J., Cloud, R., Morris, K., & Stansel, C. C. (2020). Oncology nursing considerations when developing outpatient staffing and acuity models. *Seminars in Oncology Nursing*, *36*, 151018.

Slocum, R. F., Jones, H. L., Fletcher, M. T., McConnell, B. M., Hodgson, T. J., Taheri, J., & Wilson, J. R. (2020). Improving chemotherapy infusion operations through the simulation of scheduling heuristics: A case study. *Health Systems*, *10*, 163–178.

Turkcan, A., Zeng, B., & Lawley, M. (2012). Chemotherapy operations planning and scheduling. *IIE Transactions on Healthcare Systems Engineering*, *2*, 31–49.