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

A two-phase approach for the Radiotherapy Scheduling Problem

Tu-San Pham · Louis-Martin Rousseau · Patrick De Causmaecker

the date of receipt and acceptance should be inserted later

Abstract The Radiotherapy Scheduling Problem (RTSP) focuses on optimizing the planning of radiotherapy treatment sessions for cancer patients. In this paper, we propose a two-phase approach for the RTSP. In the first phase, radiotherapy sessions are assigned to specific linear accelerators (linacs) and days. The second phase then decides the sequence of patients on each day/linac and the specific appointment times. For the first phase, an Integer Linear Programming (IP) model is proposed and solved using CPLEX. For the second phase, a Mixed Integer Linear Programming (MIP) and a Constraint Programming (CP) model are proposed. The test data is generated based on real data from CHUM, a large cancer center in Montréal, Canada, with an average of 3,500 new patients and 40,000 radiotherapy treatments per year. The results show that in the second phase, CP is better at finding good solutions quickly while MIP is better at closing optimality gaps with more run time. Lastly, a simulation is conducted to evaluate the impact of different scheduling strategies on the outcome of the scheduling. Preliminary results show that batch scheduling reduces patients' waiting time and overdue time.

Keywords Radiotherapy scheduling · Integer Programming · Constraint Programming · Simulation · Operations research

Highlights

- The paper proposes a two-phase approach for a Radiotherapy Scheduling Problem arising at a cancer center in Montréal, Canada. Solving the problem helps reduce waiting time to start treatments while considering technical constraints and patients' preferences. The approach successfully solves realistic-sized instances.
- The paper performs a simulation to evaluate different scheduling strategies (sequential, daily, weekly, etc.) and shows that batch scheduling improves waiting time as well as overdue time of patients. These analyses can assist hospital administrators by offering a novel strategy for optimizing radiotherapy scheduling.

Tu-San Pham · Louis-Martin Rousseau
Polytechnique Montréal
E-mail: tu-san.pham@polymtl.ca

Patrick De Causmaecke
KU Leuven

1 Introduction

According to the World Health Organization (WHO) [1], cancer is a leading leading cause of death worldwide. It accounts for nearly 10 million deaths globally in 2020. The incidence of cancer has been increasing sharply in the last few decades. The number of treatment facilities and personnel, however, have not grown proportionally. This not only puts strain on the treatment facilities and their staff but also results in long waiting times for patients. Numerous studies [5, 14, 4] have shown that increased waiting time for radiotherapy treatment has a negative impact on clinical outcomes. Therefore, better planning to reduce waiting time is crucial in improving cancer treatment results.

There exist many options for cancer treatment, including surgery, chemotherapy, and radiotherapy. Different treatment methods are often combined and can be repeated as needed. In this work, we aim to optimize the planning of radiotherapy treatments, an effective form of cancer treatment. Approximately 50% of all cancer patients require radiotherapy as a part of their treatment [1, 20]. In radiotherapy treatment (RT), a patient receives a daily dose of radiation to kill cancer cells. The treatment is most commonly

[1] https://www.who.int/news-room/fact-sheets/detail/cancer

delivered by a linear accelerator (*linac*). The *Radiotherapy Scheduling Problem* (RTSP) consists of deciding the radiotherapy treatment schedule for a set of patients, given a set of linacs, within a planning horizon. The problem is complicated due to several strategies and constraints that vary from hospital to hospital, resulting in numerous variants of the RTSP. In this paper, we consider a real-world variant arising at CHUM (Centre hospitalier de l'Université de Montréal), a large cancer center in Montréal, Canada.

Currently, at CHUM, scheduling is done manually by scheduling staff. RT scheduling can be divided into two levels: (1) assigning patients' treatments to days and linacs (*treatment scheduling*); and (2) assigning exact time slots for patients on each day/linac (*appointment scheduling*). Optimizing at the first level helps reduce patient waiting time while optimizing at the second level helps accommodate patients' preferences. Most existing work in the literature focuses solely on treatment scheduling [6,7,19,13]. Other work addresses both treatment scheduling and appointment scheduling opting for heuristics for large instances due to the intractability of exact models [16,15,22,21]. In this work, we propose a two-phase approach to target both levels of scheduling. Patients are assigned to specific linacs and days in the first phase, and the sequence of patients on each day/linac and their exact appointment times are decided in the second phase. An Integer Programming (IP) model is proposed for the first phase, whereas a Mixed Integer Linear Programming (MIP), and a Constraint Programming (CP) model are proposed for the second phase. Using these models, we are able to solve larger instances compared to existing approaches in the literature in terms of number of linacs, number of patients, granularity of the schedules and planning horizon. Notably, we demonstrate a successful implementation of CP in the second phase of the RTSP. With an expressive modelling language, CP is much more compact and flexible than heuristic methods. Empirical results show that CP is able to find high-quality solutions rapidly, which makes it suitable for real-world applications. MIP, on the other hand, is better at providing good lower bounds.

We evaluate the effect of the two-phase approach on patients' waiting time and overdue time in a long-term, real-world setting where scheduling decisions are made periodically. We do so by using the models within a simulation where daily patient arrival follows a Poisson distribution. We evaluate different scheduling policies: a greedy heuristic currently used at CHUM, where appointments are booked every time a patient is admitted; and batch scheduling policies where scheduling decisions are made periodically, i.e. daily, bi-weekly or weekly. We combine those policies with delaying patients' appointments until a later time point. The results show that batch scheduling improves patients' waiting time as well as overdue time. We note that batch scheduling and delaying treatment affect patient groups differently. These analyses can serve as guidelines to help healthcare administrators improve their scheduling policy.

The structure of the paper is as follows. In Section 2, we describe the problem and analyse related work. Section 3 describes our two-phase approach, while data generation is presented in Section 4. Section 5 provides numerical results for the two-phase approach. Section 6 presents the simulation to evaluate the effect of different scheduling policies on the waiting time and overdue time. Finally, Section 7 closes the paper with conclusions and topics for future work.

2 Problem statement and related work

We herein describe the problem considered in this paper and then discuss related work.

2.1 Problem statement

In this paper, we study a real variant of the RTSP in the context of CHUM. In a one-year period spanning 2017-2018, CHUM treated 3,500 patients and carried out 40,000 radiotherapy treatments. The center is equipped with 10 linacs, seven of which are often used at maximum capacity. The remaining three linacs are specialized linacs required for certain types of treatment. Cancer treatment is a complicated process with many procedures involved. The treatment workflow used at CHUM is illustrated in Figure 1. Once a new consultation request is made, a (potential) patient first undergoes a consultation session where the doctor explains their cancer status and options. If the patient agrees to proceed with the treatment, he or she will go through preparation steps such as external consultation and exams, approving the care plan and booking the scans. The next step is treatment planning, where a set of tests and imaging need to be done, which could include x-rays, CT scans, MRI scans, and PET scans. The treatment plan will need to be verified and approved by a physicist before the treatments are prepared. The radiotherapy treatments are then carried out, with review and possibly revision during the course of treatment. Finally, the treatment ends with post-treatments and follow-ups.

In radiotherapy treatment, a patient receives a high dose of radiation to kill cancer cells. The radiation is divided into small doses called *fractions*. A series of fractions is delivered using a linac during the course of several consecutive days, with breaks on the weekend. Currently, the scheduling is done manually. Once a treatment plan is approved, linacs and staff are booked. Patients are classified according to four different categories, each requiring different treatment deadlines. Palliative patients (categories P1 and P2) need urgent care to relieve intense pain, hence the treatment deadline is set to one and three days, respectively. Curative

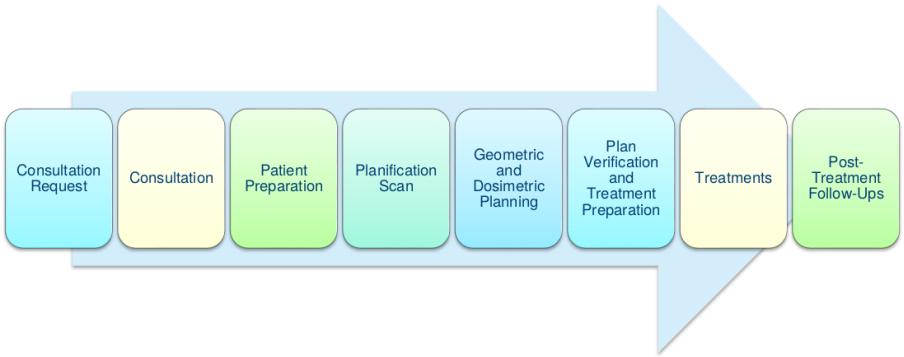

Fig. 1: Radiotherapy treatment workflow at CHUM[2].

Category	Proportion (%)	Treatment deadline (days)	Percentage of overdue treatment (%)	Average waiting time (days)
P1	0.44	1	14.29	1.09
P2	27.14	3	79.89	6.91
P3	41.36	14	74.55	18.11
P4	31.06	28	29.89	22.59

Table 1: Waiting time targets, percentage of overdue treatment and average waiting time of cancer patients at CHUM in 2017-2018 by patient category.

patients (categories P3 and P4) have their deadline set to 14 and 28 days, respectively. A patient's waiting time is calculated based upon the time elapsed between his or her admission date and the date of the first treatment. Currently, CHUM is under-capacity and hence faces difficulty meeting treatment deadlines for many patients. Table 1 shows patient categories along with their proportion, percentage of overdue treatments and average waiting time at CHUM in the period between November 2017 and July 2019. As can be seen from the table, more than 70% of patients in category $P2$ and $P3$ are treated after their due dates. The objective of our problem is hence minimizing waiting time, i.e. treatments should start as soon as possible once the patient is ready. The treatment deadline is treated as a soft constraint.

Based on a patient's diagnosis, doctors will propose a treatment plan, which determines how many fractions the patient will receive and the length of each fraction. At CHUM, fraction lengths range from 10 to 165 minutes. The majority of patients have fraction lengths of 25 to 30 minutes. Since fraction lengths at CHUM are always multiples of 5 minutes, we set the granularity of our schedules to 5-minute blocks. For example, a fraction length of 15 minutes is equivalent to 3 time blocks. A fine-grained schedule makes our problem substantially more difficult compared to other works in literature, which either assume a fixed duration of treatment for all patients or use larger time blocks, e.g. 10 or 20 minutes.

Once the treatment starts, it needs to be carried out daily, with breaks only on the weekend. Each patient has a ready date and a due date and the patient can only start treatment after the ready date. When a patient is scheduled after his or her due date, we call this overdue treatment, which is penalized heavily. Although undesirable, patients can switch between linacs during the course of treatment. At CHUM, some patients have to be treated on a specialized linac. To simplify the problem, we do not consider those special cases and assume that all patients can switch between all linacs. However, the total number of linacs assigned to a given patient should be minimized, for several reasons: (1) each linac is usually associated with a set of technicians who familiarize themselves with the patient's condition prior to treatment; and (2) there might be some patient-specific setup required before each treatment. Changing treatment rooms often will lengthen the process. We also note that some part of linac capacity is reserved for emergency patients. For patients' convenience, appointment times should be consistent throughout the course of treatment. We allow for movement of some fixed appointments to make room for or to better accommodate new patients. However, those changes should be minimized. Some patients, especially those living far from the treatment center, have a time window preference within which they wish to have their appointments.

The most important objectives of the problem are reducing waiting time and overdue time. Other objectives deal with appointment time consistency, patient-linac consistency, respecting time window preferences, and minimizing changes to fixed appointments. The following terms hence need to be minimized: (1) deviations in appointment times during treatment; (2) the number of linacs assigned to each patient; (3) the violation of time window preferences; and (4) changes to fixed appointments. Preliminary experiments using a MIP

[2] Figure provided by CHUM.

formulation show that using an exact model to solve all objectives directly is intractable. In addition, from a practical point of view, reducing waiting time and overdue time are the most important goals in RT scheduling and hence should not be compromised to suit other objectives. Therefore, we propose a two-phase approach for the problem. In the first phase, the starting dates of treatments are decided, along with the linac for each treatment. The most important objectives, i.e. reducing waiting time and overdue time, along with patient-linac consistency are handled in this phase. The second phase, which decides the sequence of patients and the exact appointment times on each day and each linac, resolves the remaining objectives.

2.2 Related work

Radiotherapy scheduling is a relatively young research field. In 1993, Larsson [12] introduces the first software for a radiotherapy patient scheduling system as a replacement for the conventional paper-based system, and predicts that it will support the use of sophisticated mathematical models for the problem in the future. However, it is not until 2008 that the first two mathematical models for the Radiotherapy Scheduling Problem are published by Conforti [6]. The models maximize the number of treated patients within a given horizon with identical treatment times and a single linac. The models are then extended to tackle more realistic instances where the treatment times of patients vary [7]. The first models are based on lots of assumptions and simplification. Later work focuses more on dynamic scheduling to tackle stochastic factors in radiotherapy treatments. Saure et al. [19] model the radiotherapy scheduling problem as a discounted Markov Decision Process (MDP), to provide a dynamic policy that takes into account future events. A similar approach is also utilized by Gocgun [9], which additionally allows for cancellation of treatments.

In [13], Legrain et al. propose a hybrid method combining stochastic optimization and online optimization to solve the problem in an online fashion, while taking into account information on the distribution of future arrivals of patients. All of the aforementioned work focuses solely on the first scheduling level, i.e. assigning treatments to dates, while neglecting the sequence of patients on each linac for each day. They focus on taking into account stochastic future arrivals of patients in a dynamic setting to reduce waiting time. Our work, in contrast, persues a different goal: building a complete scheduling system which takes into account patients' preferences. We therefore focus on both phases of scheduling. In addition, we target larger instances. The aforementioned models are very costly and hence are limited to relatively small instances. In [19], their practical example has an arrival rate of 8.25 requests per day, with 120 appointment slots which is equivalent to three linacs. Legrain et al. [13] consider a 20-minute time slot for each patient and tests the algorithm on instances with up to two linacs and arrival rate λ less than 3.5. In addition, they address the scheduling problem in an online setting where patients leave the center with their appointments scheduled, in constrast to our batch scheduling setting. Those models hence cannot be applied to the problem that we consider at CHUM.

In [8], the authors propose two CP models and one IP model to solve both treatment scheduling and appointment scheduling for RT treatment. They compare the performance of those models and come up with a conclusion similar to ours, namely that CP models find feasible solutions earlier. They minimize violation of treatment deadlines as the primary objective and deviation in appointment times as the secondary objective. However, they assign patients to *time windows* of 1.5 to 4 hours instead of deciding the exact appointment times, which reduces the complexity of the problem. Their test instances have three linacs and arrival rates ranging from four to eight patients, with four to six time windows per day.

Another form of radiotherapy treatment using a particle beam is studied by two research groups in Vienna [16, 15, 22]. In particle beam treatment, a single centralized beam serves different treatment rooms. Hence it is important to have a detailed schedule with exact appointment time of patients to utilize the beam most efficiently. Therefore, even though their setting is different, their problems share many characteristics with ours. Their problems consider both dates of treatments and sequence of treatments. They also aim to minimize inconsistencies in appointment time. As shown in [16], their exact model for the problem is highly intractable. All of the aforementioned papers propose metaheuristics for scheduling particle beam radiotherapy treatment. In phase 2 of our problem, we propose a CP approach to solve the radiotherapy appointment scheduling problem. Even though optimal solutions are not obtained for real-world size instances, we show that our approach provides high-quality solutions in a short run time. Our CP approach also has the advantages of a compact model which is easy to adapt to other variants and additional constraints.

In [22], Volg et al. describe how problem instances are generated. However, their instances are generated in such a way that optimal solutions can be calculated, i.e. there exist "beautiful" schedules with no idle times, and no violation of side constraints. In our experience, realistic instances have more irregularities in the schedule due to the stochastic nature of healthcare. Those irregularities cause numerical difficulty in algorithms, both exact and metaheuristic. Our paper focus on generating and solving such realistic instances.

In [18, 17] and [2], the authors consider a problem which arises at the Nottingham University Hospitals. The first two publications present several constructive approaches and a GRASP-based algorithm. The third one proposes an IP model

assigning patients to days and linacs, which corresponds to our first phase. Those papers investigate whether delaying the scheduling decision can lead to better schedules. This is similar to our simulation in Section 6 of the paper. The authors also use a subset of the data to partially fill the schedule to create test instances. However, the details on how it is done are not presented. Kapamara et al. [10] develop four heuristic methods to schedule patients in both pre-treatment and treatment stages. Pre-treatment for radiotherapy patients is also considered in [3]. The authors model the problem as an optimization problem with hierarchical multiple objectives and solve it as a series of single-objective optimization problems. Each subproblem is formulated as a mathematical programming model.

In yet another work, the authors consider a real problem taken from the Netherlands Cancer Institute (NKI) [21]. Similar to our problem, they consider two phases of the scheduling process and focus on satisfying patient preferences on appointment time. An exact model and a heuristic pre-assigning patients to linacs to break down the problem into many sub-problems are proposed. With the MIP model, they obtain optimal solutions for instances up to 66 patients and two linacs with a planning horizon of five days, i.e. one working week, which is significantly shorter than the planning horizon we consider in this paper (60 days). In addition, the authors do not describe how they generate problem instances, which has a significant effect on the difficulty of the instances, as proven by the numerical results in Section 5.2.

3 A two-phase approach

The RTSP consists of finding the best treatment schedule for a set of patients $\mathcal{P}$ over a planning horizon $\mathcal{T}$, given a set of linacs $\mathcal{K}$. The sets of palliative patients and curative patients are denoted as $\mathcal{P}^{\mathcal{P}}$ and $\mathcal{P}^{C}$ respectively, $\mathcal{P} = \mathcal{P}^{\mathcal{P}} \cup \mathcal{P}^{C}$. Each instance consists of a set of fixed patients with appointments made from the previous scheduling decisions. This set is denoted as $\bar{\mathcal{P}}$, while the set of new patients is denoted as $\hat{\mathcal{P}}$, $\mathcal{P} = \bar{\mathcal{P}} \cup \hat{\mathcal{P}}$. Each patient $i \in \mathcal{P}$ has a ready date r_i when the patient is ready for treatment, and a due date d_i, before which the treatment must start. Each patient i is associated with a treatment plan, which specifies the number of fractions I_i that the patient needs to receive and the duration of each fraction (p_i). Each linac k has a capacity (C_k^t) measured in blocks of 5 minutes. $\hat{C}_k^t$ represents the available capacity of linac k in day t after deducting the fixed appointments from the previous scheduling decisions. The set of parameters is presented in Table 2.

The approach consists of two phases. Phase 1 determines the assignment of fractions to days and linacs. Phase 2 decides the sequence of patients and the exact appointment times given the assignment fraction-day-linac. For phase 1, we propose an IP model. For phase 2, we propose a MIP and a CP model.

Parameter	Explanation
$\mathcal{P}$	set of patients
$\mathcal{P}^{\mathcal{P}}$	set of palliative patients
$\mathcal{P}^{C}$	set of curative patients
$\hat{\mathcal{P}}$	set of new patients
$\bar{\mathcal{P}}$	set of fixed patients from the previous scheduling decisions
$\mathcal{T}$	set of days in the horizon
$\mathcal{S}$	set of time blocks each day
$\mathcal{K}$	set of linacs
C_k^t	total capacity of linac k in day t
$\hat{C}_k^t$	available capacity of linac k in day t
r_i	ready date of patient i
d_i	due date of patient i
p_i	fraction duration for patient i
I_i	number of fractions of patient i

Table 2: Problem parameters

3.1 Phase 1: Assigning fractions to days and linacs

The following variables are defined on the set of new patients $\hat{\mathcal{P}}$.

- x_{ik}^t = 1 if the new patient $i \in \hat{\mathcal{P}}$ is assigned to day t on linac k, 0 otherwise.
- z_i^t = 1 if the new patient i receives his or her first treatment on day t, 0 otherwise.
- w_{ik} = 1 if the new patient i is assigned to linac k during his or her treatment.

Model:

$$\text{minimize } \omega_1 \sum_{i\in\hat{\mathcal{P}}}\sum_{t\in\mathcal{T}}(t-r_i)^2 z_i^t + \omega_2 \sum_{i\in\hat{\mathcal{P}}}\sum_{t\in\mathcal{T},t>d_i}(t-d_i)^2 z_i^t + \omega_3 \sum_{i\in\hat{\mathcal{P}}}\sum_{k\in\mathcal{K}} w_{ik} \tag{1}$$

subject to

$$\sum_{k\in\mathcal{K}} x_{ik}^t \le 1 \qquad \forall i \in \hat{\mathcal{P}}, t \in \mathcal{T} \tag{2}$$

$$\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} x_{ik}^t \le I_i \qquad \forall i \in \hat{\mathcal{P}} \tag{3}$$

$$\sum_{k\in\mathcal{K}}(x_{ik}^t - x_{ik}^{t-1}) \le \sum_{k\in\mathcal{K}} x_{ik}^n \qquad \forall i \in \hat{\mathcal{P}}, t \in \mathcal{T}, x_{ik}^{-1} = 0, \; n = t+1,\ldots,min\{|\mathcal{T}|-1, t+I_i-1\} \tag{4}$$

$$x_{ik}^t = 0 \qquad \forall i \in \hat{\mathcal{P}}, k \in \mathcal{K}, t \in \{0,\ldots,r_i-1\} \tag{5}$$

$$\sum_{i\in\hat{\mathcal{P}}} p_i x_{ik}^t \le \hat{C}_k^t \qquad \forall t \in \mathcal{T}, k \in \mathcal{K} \tag{6}$$

$$\sum_{i\in\mathcal{P}^C} p_i x_{ik}^t \le \hat{C}_k^t - \gamma C_k^t \qquad \forall t \in \mathcal{T}, k \in \mathcal{K} \tag{7}$$

$$\sum_{t\in\mathcal{T}} z_i^t = 1, \qquad \forall i \in \hat{\mathcal{P}} \tag{8}$$

$$z_i^t \geq \sum_{k\in\mathcal{K}} x_{ik}^t - \sum_{k\in\mathcal{K}} x_{ik}^{t-1}, \quad \forall i \in \hat{\mathcal{P}}, t \in \mathcal{T}, x_{ik}^{-1} = 0 \tag{9}$$

$$\sum_{k\in\mathcal{K}} x_{ik}^t \geq z_i^t, \qquad \forall i \in \hat{\mathcal{P}}, t \in \mathcal{T} \tag{10}$$

$$w_{ik} \geq x_{ik}^t \qquad \forall i \in \hat{\mathcal{P}}, t \in \mathcal{T}, k \in \mathcal{K} \tag{11}$$

$$x_{ik}^t, z_i^t, w_{ik} \in \{0,1\} \qquad \forall i \in \hat{\mathcal{P}}, t \in \mathcal{T}, k \in \mathcal{K} \tag{12}$$

Objective function (1) consists of three terms which minimize (1) the squared number of waiting days; (2) the squared number of overdue days; and (3) the total number of linacs assigned to a patient during his or her treatment. We minimize the squared value of waiting days and overdue days to make sure the violation is distributed evenly between patients, i.e. no patient has to wait much longer than the others. ω_1, ω_2 and ω_3 are the weights of the three terms, respectively. The values of the weights can be decided by practitioners, depending on how much they consider one objective more important than the others. To analyse the weight setting, we first look into the value ranges of the individual terms in the objective function. The first two terms in the objective function (1) are the squared values of the waiting time and overdue time respectively, which are typically less than 30 days. The value range of the first two terms for each individual patient is hence from 0 to 900 (approximately speaking). The value range of the third objective term is the total number of linacs, which is less than 7 in our test instances. We consequently set $\omega_1 = \omega_3 = 1$ and $\omega_2 = 1000$ to prioritize minimizing overdue time over minimizing waiting time and linac consistency. Constraints (2) ensure each patient is scheduled on at most one linac per day. Constraints (3) limit the number of sessions delivered to each patient. Constraints (4) ensure daily treatment. Constraints (5) ensure no patients are scheduled before their ready date. Constraints (6) ensure the capacity of each linac is not exceeded. Constraints (7) save γ percent of linac capacity for palliative patients. Constraints (8) make sure all new patients are scheduled. Constraints (8, 9, 10) define the first day of treatment. Constraints (11) keep track of which linac a patient is assigned to during his or her course of treatment. Constraints (12) are integrality constraints.

3.2 Phase 2: Order of patients on linacs - MIP model

In phase 2, three objectives are considered: (1) consistency in appointment times; (2) time window preferences; and (3) changes to fixed appointments. The first two objectives are applied for new, curative patients only. The last objective is applied for fixed appointments regardless of patient category.

The input for phase 2 consists of the assignments of patients to days and linacs. Let P_k^t be the set of patients scheduled on linac k, day t, $P_k^t = \bar{P}_k^t \cup \hat{P}_k^t$ where $\bar{P}_k^t$ is the set of fixed patients from the previous scheduling decisions and $\hat{P}_k^t$ is the set of new patients. Each fixed patient $i \in \bar{P}_k^t$ has an appointment time slot from the previous scheduling period, denoted as $\bar{s}_i^t$. The model allows for changes to the appointment times of fixed patients, but tries to minimize those changes. We define variables and constraints only on the set of linacs and days where there exist new assignments of treatments.

Variables:

- y_{is}^{tk}: binary variables indicating patient i is scheduled for time block s on day t, linac k.
- $\bar{\Delta}_i^t$ and $\underline{\Delta}_i^t$: representing the deviation (earlier or later) from the time window for new, curative patient $i \in \hat{\mathcal{P}}^{\mathcal{C}}$ on day t.
- $\bar{\theta}_i$ and $\underline{\theta}_i$: representing the earliest and latest appointment time slots for new, curative patient $i \in \hat{\mathcal{P}}^{\mathcal{C}}$.
- $\bar{\phi}_i^t$ and $\underline{\phi}_i^t$: representing changes in appointment times (to earlier or later) of fixed patient $i \in \bar{\mathcal{P}}$ on day t.

Model:

$$\begin{aligned}\text{minimize } \omega_4 \sum_{i\in\hat{\mathcal{P}}^{\mathcal{C}}, t\in\mathcal{T}} (\bar{\Delta}_{it} + \underline{\Delta}_{it}) + \omega_5 \sum_{i\in\hat{\mathcal{P}}^{\mathcal{C}}} (\bar{\theta}_i - \underline{\theta}_i) \\ + \omega_6 \sum_{i\in\bar{\mathcal{P}}, t\in\mathcal{T}} (\bar{\phi}_i^t + \underline{\phi}_i^t)\end{aligned} \tag{13}$$

subject to

$$\sum_{s\in\mathcal{S}} y_{is}^{tk} = 1 \qquad \forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \mathcal{P}_k^t \tag{14}$$

$$\sum_{i\in\mathcal{P}_k^t} y_{is}^{tk} \leq 1 \qquad \forall t \in \mathcal{T}, k \in \mathcal{K}, s \in \mathcal{S} \tag{15}$$

$$\begin{aligned} y_{is}^{tk} \leq 1 - y_{i's'}^{tk} \quad & \forall t \in \mathcal{T}, k \in \mathcal{K}, i, i' \in \mathcal{P}_k^t, i \neq i' \\ & s = 0, \ldots, |\mathcal{S}| - p_i \\ & s' = s+1, \ldots, s+p_i-1 \end{aligned} \tag{16}$$

$$\begin{aligned} y_{is}^{tk} = 0 \quad & \forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \mathcal{P}_k^t, \\ & s = |\mathcal{S}| - p_i + 2, ...|\mathcal{S}| - 1 \end{aligned} \tag{17}$$

$$\begin{aligned} \bar{\Delta}_i^t \geq y_{is}^{tk}(t_i^{min} - s) \quad & \forall t \in \mathcal{T}, k \in \mathcal{K}, \\ & \forall i \in \hat{\mathcal{P}}_k^{\mathcal{C}t}, s \in \mathcal{S} \end{aligned} \tag{18}$$

$$\begin{aligned} \underline{\Delta}_i^t \geq y_{is}^{tk}(s - t_i^{max}) \quad & \forall t \in \mathcal{T}, k \in \mathcal{K}, \\ & \forall i \in \hat{\mathcal{P}}_k^{\mathcal{C}t}, s \in \mathcal{S} \end{aligned} \tag{19}$$

$$\bar{\theta}_i \leq \sum_{s\in\mathcal{S}} s y_{is}^{tk} \qquad \forall t \in \mathcal{T}, i \in \hat{\mathcal{P}}^{\mathcal{C}t}, k = k_i^t \tag{20}$$

$$\underline{\theta}_i \geq \sum_{s\in\mathcal{S}} s y_{is}^{tk} \qquad \forall t \in \mathcal{T}, i \in \hat{\mathcal{P}}^{\mathcal{C}t}, k = k_i^t \tag{21}$$

$$\bar{\phi}_i^t \geq \bar{s}_i^t - \sum_{s \in \mathcal{S}} s y_{is}^{tk}, \quad \forall t \in \mathcal{T}, k \in \mathcal{K}, \forall i \in \bar{\mathcal{P}}_k^t \quad (22)$$

$$\underline{\phi}_i^t \geq \sum_{s \in \mathcal{S}} s y_{is}^{tk} - \bar{s}_i^t \quad \forall t \in \mathcal{T}, k \in \mathcal{K}, \forall i \in \bar{\mathcal{P}}_k^t \quad (23)$$

$$y_{is}^{tk} \in \{0,1\} \quad \forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \mathcal{P}_k^t \quad (24)$$

$$\bar{\Delta}_i^t, \underline{\Delta}_i^t, \bar{\theta}_i, \underline{\theta}_i \in [0, C_k^t] \quad \forall t \in \mathcal{T}, i \in \hat{\mathcal{P}}^{\mathcal{C}} \quad (25)$$

$$\bar{\phi}_i^t, \underline{\phi}_i^t \in [0, C_k^t] \quad \forall t \in \mathcal{T}, i \in \bar{\mathcal{P}} \quad (26)$$

Objective function (13) minimizes the deviation from time windows, inconsistency in appointment times, and changes to appointments of fixed patients with the respective weights ω_4, ω_5, and ω_6. Similarly to phase 1, those weights can be adjusted by practitioners to reflect the treatment center's priorities. The value ranges of the three terms in the objective function (13) for each patient is the deviation from time windows, inconsistency in appointment times, and change to appointment times, which are all less than or equal to the number of time slots per day, e.g. 120 in our problem setting. We therefore set $\omega_4 = \omega_5 = 1$ and $\omega_6 = 60$ to prioritize not changing fixed appointments over time preferences and consistency in appointment times. Constraints (14) ensure each patient is assigned to exactly one block on treatment day. Constraints (15) make sure each slot is assigned to at most one patient. Constraints (16) ensure the number of required blocks for each patient by blocking the next time blocks on the linac. Constraints (17) ensure no patients are scheduled on the last time block(s) of the day which would be insufficient for their sessions. Constraints (18) and (19) define any deviation from the time preferences. Constraints (20) and (21) define the earliest and latest appointment time of a patient during the course of treatment. Constraints (22) and (23) define changes in appointment times for fixed patients. The rest are domain constraints.

3.3 Phase 2: Order of patients on linacs - CP model

As an alternative to the MIP model, a Constraint Programming (CP) model is proposed for phase 2. The model uses variables and constraints supported by IBM CP Optimizer [11]. CP Optimizer supports two main types of decision variables, namely *integer* and *interval variables*. An interval variable consists of a start point and an end point, representing an interval of time. Another type of variable used in this model is a *sequence variable*. A sequence variable consists of a set of interval variables. CP Optimizer offers many built-in constraints which makes it convenient modelling scheduling constraints.

In this model, the parameters are similar to the ones in the MIP model in Section 3.2. We hereby introduce the variables and constraints.

- The interval variable y_{ki}^t represents the appointment duration of patient i on day t, linac k. y_{ki}^t has a start point $s(y_{ki}^t)$ and an end point $e(y_{ki}^t)$. The duration of y_{ki}^t ($d(y_{ki}^t) = e(y_{ki}^t) - s(y_{ki}^t)$) is equal to the fraction length p_i of the corresponding patient. Recall that C_k^t is the capacity of linac k on day t. The domain of y_{ki}^t is defined as follows:

$$\forall t \in \mathcal{T}, \forall k \in \mathcal{K}, i \in \mathcal{P}_k^t : \begin{cases} s(y_{ki}^t) \in [0, C_k^t - p_i] \\ e(y_{ki}^t) \in [C_k^t - p_i, C_k^t] \\ d(y_{ki}^t) = p_i \end{cases} \quad (27)$$

- The *sequence variable* seq_k^t contains all interval variables y_{ki}^t corresponding to all fractions performed on linac k, day t. The specialized constraint *noOverlap* is imposed on seq_k^t to ensure that a fraction must end before the next one starts.

$$\forall t \in \mathcal{T}, k \in \mathcal{K} : \begin{cases} seq_k^t = \{y_{ki}^t | i \in \mathcal{P}_k^t\} \\ \text{noOverlap}(seq_k^t) \end{cases} \quad (28)$$

- The interval variable x_i represents the time span of all appointments for patient i. x_i starts at the earliest appointment time and ends at the latest appointment end time of patient i. The longer the duration of x_i, the more inconsistency there is in appointment times for the corresponding patient during his or her course of treatment. This is imposed by the specialized constraint *span* of CPOptimizer. This constraint is only applied for new, curative patients. An illustration of the *span* constraint can be found in Figure 2.

$$i \in \hat{\mathcal{P}}_k^{\mathcal{C}t} : \begin{cases} s(x_i) \in [0, C_k^t] \\ e(x_i) \in [0, C_k^t] \\ span(x_i, [y_{ki}^t | \forall t \in \mathcal{T}, \forall k \in \mathcal{K}]) \end{cases} \quad (29)$$

- $\bar{\Delta}_i^t$ and $\underline{\Delta}_i^t$ represent the deviation (to earlier or later) from the preference time windows of new, curative patients. They are defined by constraints (31).

$$\forall t \in \mathcal{T}, i \in \hat{\mathcal{P}}^{\mathcal{C}} : \begin{cases} \bar{\Delta}_i^t \in [0, C_k^t] \\ \underline{\Delta}_i^t \in [0, C_k^t] \end{cases} \quad (30)$$

$$\forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \hat{\mathcal{P}}^{\mathcal{C}} : \begin{cases} \bar{\Delta}_i^t \geq t_i^{min} - StartOf(y_{ki}^t) \\ \underline{\Delta}_i^t \geq StartOf(y_{ki}^t) - t_i^{max} \end{cases} \quad (31)$$

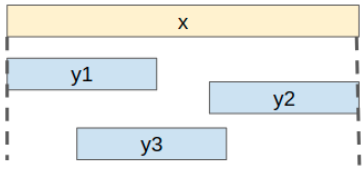

Fig. 2: An illustration of constraint $span(x, [y_1, y_2, y_3]))$, which implies $s(x) = min\{s(y_1), s(y_2), s(y_3)\}$ and $e(x) = max\{e(y_1), e(y_2), e(y_3)\}$
.

The objective function (32) consists of three parts which minimize the following objectives, respectively: (1) the violation of the time window preference of patients; (2) inconsistency in appointment times; and (3) changes in appointments for fixed patients. The corresponding weights ω_4, ω_5 and ω_6 take up the same values with those in the MIP model in Section 3.2.

$$\begin{aligned}\text{minimize } & \omega_4 \sum_{t\in\mathcal{T}} \sum_{i\in\hat{\mathcal{P}}^{ct}} (\bar{\Delta}_i^t + \underline{\Delta}_i^t) \\ & +\omega_5 \sum_{i\in\hat{\mathcal{P}}^c} LengthOf(x_i) \\ & +\omega_6 \sum_{t\in\mathcal{T}} \sum_{k\in\mathcal{K}} \sum_{i\in\bar{\mathcal{P}}_k^t} abs(StartOf(y_{ki}^t) - \bar{s}_i^t) \qquad (32)\end{aligned}$$

4 Data generation

Test data is generated based on the real data from CHUM. In general, each type of cancer is associated with a *generic treatment plan*, which defines the number and duration of fractions. However, those generic treatment plans are used as a guideline, based on which doctors decide a *personal treatment plan* specific to each patient's condition. In this paper, we use *treatment plan* to refer to a personal, individual treatment plan. A personal treatment plan sets patient's category (palliative or curative, which type), number of fractions, and duration of fractions. Each patient has an admission date (the date on which the patient decides to start treatment or their information becomes available in the system), a ready date and a due date, which are together referred as timeline information. The following are some key definitions for data generation.

- A **treatment plan pool** consists of a large number of personal treatment plans taken from real data from CHUM by omitting the information of patients and their timeline information. The pool consists of 5000 treatment plans, which corresponds to more than one year of data.
- A **patient flow** represents the flow of new patients admitted to the hospital daily. The number of patients arriving daily follows a Poisson distribution. The parameter of the Poisson distribution is λ (the *event rate*), which represents the average number of new patients per day (*arrival rate*). Each patient flow instance is hence parameterized by λ and the *number of days of simulation*, denoted by l. The expected total number of patients P is decided by λ and l. Given a λ, a set of l numbers is generated respecting the Poisson distribution. For each day of the simulation period (d_0 to d_{l-1}), a corresponding number of virtual patients are generated and the corresponding date is set as their admission date. Each patient is assigned to a treatment plan selected randomly from the treatment plan pool. The distance between the admission date and the ready date is then generated randomly in a range defined by the patient category as listed in table 3. The due date is calculated from the admission date and the patient's category as listed in table 1.
- An **instance scenario** is a partially-filled schedule. Given a number of linacs and a patient flow, an instance scenario is generated as follows. For each patient in the patient flow, a start date is generated randomly within the range $[-10, 24]$. The start date of a patient being negative means that the patient starts his/her treatment before the first day of the considered instance. The patient is then assigned to a **series** of free slots in the schedule, i.e. a set of appointments in several consecutive days with identical appointment times. The process is repeated until the desired occupancy rate of linacs is met. In our instances, the generating process is terminated when the occupancy of the first day reaches 90% of linacs' capacity. The layout of an instance scenario has strong impact on the difficulty of an instance. Therefore, we aim to generate instances as realistic as possible. To reflect a real scenario, stochastic factors are introduced by randomly swapping some appointments and moving some appointments to other slots on the same day.
- A complete test **instance** consists of an instance scenario and a patient flow. Each instance is hence parameterized by several parameters as listed in table 4.

Category	Duration (days)
P1	0
P2	$0 \sim 2$
P3	$5 \sim 7$
P4	$5 \sim 7$

Table 3: Duration between admission date and ready date by patient category.

Parameter	Meaning
λ	average number of new patients per day
l	number of days of simulation
$\lvert\mathcal{P}\rvert$	number of patients (decided by λ and l)
$\lvert\mathcal{K}\rvert$	number of linacs

Table 4: Instance parameters.

All datasets used in this paper can be found at http://hanalog.polymtl.ca/wp-content/uploads/2020/10/RTSP_dataset.zip.

5 Evaluation of the two-phase approach

To evaluate our models, we conduct two experiments in pursuit of two different goals. The first experiment aims at eval-

uating the performance of the two-phase approach, while the second experiment focuses on evaluating an instance's difficulty by its characteristics. All experiments are carried out on an Intel Core i7-7800 3.50GHz running Oracle Linux Server 7.9. IBM ILOG CPLEX and CP Optimizer version 20.1 are used as the solvers.

5.1 Evaluating the two-phase approach

To evaluate the two-phase approach, 20 instances are generated. In order to cover the target instance sizes, we first choose the number of linacs, which ranges from one to seven. Those linacs are filled with fixed patients as described in Section 4. For a given number of linacs, the number of fixed patients can vary slightly from instance to instance due to the stochastic nature of the procedure. The number of fixed patients in our instances ranges from 26 to 272 patients. To choose the number of new patients for an instance, we first carry out a preliminary analysis to choose a reasonable range of new patients that the given number of linacs can accommodate without resulting in extremely overdue treatments. We then choose several values within the resulting acceptable range to represent different crowding levels. In our instances, the number of new patients ($\hat{\mathcal{P}}$) ranges from 10 to 82 patients while the total number of patients ($\mathcal{P}$), including fixed patients from the previous scheduling periods, ranges from 36 to 347 patients. The planning horizon is set to 60 days. We test two versions of the two-phase approach: the MIP and the CP version, where the MIP and CP models are applied to phase 2, respectively. Both phases are given a time budget of one hour each.

We first analyze the results of phase 1, shown in Table 5. For each instance the number of linacs, number of patients and number of new patients by category are present. The average waiting time and overdue time overall and by patient category are reported. As one can observe, we are able to obtain solutions with an optimality gap of less than 5% for instances up to 7 linacs and 82 new patients, within 1 hour. The model therefore can be used for real-size instances at CHUM. The waiting time is distributed properly according to a patient's category, which is lowest at $P1$ and $P2$. Overdue time occurs mostly at category $P3$, which is in line with the real-world situation. We would like to highlight that the average waiting time and overdue time depend on the ratio of total number of patients and total linac capacity.

The results of phase 2 are presented in Table 6. To examine the quality of the solutions, for each instance, in addition to the objective value, optimality gap and lower bound, we also report the violation of soft constraints, including the average violation of time preferences; the average deviation of appointment time during the treatment of new curative patients; and the average change in appointment time of fixed patients, all measured in time blocks. As can be seen from the table, for small instances, both MIP and CP give good results. Out of the first ten instances, MIP closes the optimality gap in seven instances, and gives worse result than CP in only in one instance. CP does not fall far behind with the objective values not too far away from the optimal solution obtained by MIP. However, it appears to be incompetent at providing good lower bounds. In the remaining ten instances, CP gives better solutions in nine instances compared to those generated by MIP. When considering the violation of patients' preferences, one can see that the solutions generated by MIP in large instances violate patients' preferences to a large extent, especially in changing fixed appointments. CP, on the other hand, provides good solutions in term of patients' preferences. In the largest instances with seven linacs, the violation of time window is less than three time blocks (e.g., 15 minutes) per section while the average change to fixed appointments is less than two time blocks per fixed patient. The average deviation of appointment time is less than 43 time blocks, which appears to be high. However, this is inevitable since we favor not changing fixed appointments over having stable appointment times for patients. Comparing these results with optimal solutions for small instances, where the average deviation in appointment times ranges from 16 to 50 time blocks, we conclude that this violation is acceptable given the instances' layout. There are two conclusions from this experiment. First, MIP is better at closing the optimality gap in some instances but fails to provide good solutions in others, while CP is better at finding good solutions in all instances. Second, CP provides good solutions for instances with up to seven linacs, the same number of linacs at CHUM.

To take a closer look at how MIP and CP perform given a short run time, we examine the results of both solvers on the second phase after five minutes of run time, in Table 7. MIP finds optimal solutions in two out of 20 instances while CP obtains an optimal solution in only one instance. For the remaining instances, CP gives better solutions than MIP to a great extent. In large instances, the changing of fixed appointments is high (up to 20 time blocks or 100 minutes per patient). The remaining violation seems reasonable compared to solutions obtained after one hour of run time. This violation, however, is still acceptable. Meanwhile, MIP gives very bad solutions and fails to give any incumbent in three instances. These results suggest that if obtaining a good solution with a short run time is the priority, then CP is a better approach compared to MIP.

5.2 Evaluating instance difficulty by instance characteristics

To confirm our speculation that instance difficulty is highly affected by the way the partial schedules are filled, we conduct an experiment on two instance sets, namely *orderly* (O)

										Average waiting time (days)					Average overdue time (days)				
Instance	$\mathcal{K}$	$\mathcal{P}$	$\hat{\mathcal{P}}$	$\hat{\mathcal{P}}_1$	$\hat{\mathcal{P}}_2$	$\hat{\mathcal{P}}_3$	$\hat{\mathcal{P}}_4$	runtime	gap	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$	overall	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$	overall
ins01	1	36	10	0	2	4	4	0.09	0.00%	-	1	8.25	12.25	8.4	-	0.00	0	0	0
ins02	1	55	15	0	4	6	5	0.47	0.00%	-	1.25	12.83	19.00	11.8	-	0.00	1.83	0	0.73
ins03	1	54	17	1	6	6	4	0.62	0.00%	3	3	12.5	19.25	10.18	2	1.00	0.33	0	0.59
ins04	2	96	15	0	5	5	5	0.78	0.00%	-	1	9.8	15.80	8.87	-	0.00	0.2	0	0.07
ins05	2	94	16	0	4	11	1	0.99	0.00%	-	1.25	9.64	16.00	7.94	-	0.00	0.27	0	0.19
ins06	3	121	30	0	4	20	6	6.63	0.00%	-	1.75	8.5	9.50	7.8	-	0.00	0	0	0
ins07	3	137	33	0	8	13	12	81.00	0.00%	-	1	11	13.67	9.55	-	0.00	0.08	0	0.03
ins08	3	147	35	0	8	17	10	25.66	0.00%	-	1	10	13.00	8.8	-	0.00	0	0	0
ins09	4	172	42	0	13	19	10	57.21	0.00%	-	0.69	10	13.70	8	-	0.00	0.26	0	0.12
ins10	4	226	47	0	20	15	12	1157.21	0.01%	-	1.65	10.33	16.00	8.09	-	0.05	0.33	0	0.13
ins11	4	191	52	0	12	21	19	3601.29	0.20%	-	1.33	10.24	15.63	10.15	-	0.00	0.29	0	0.12
ins12	5	259	55	0	22	18	15	3601.15	0.06%	-	1	8.67	12.93	6.76	-	0.00	0.06	0	0.02
ins13	5	228	57	0	20	18	19	3600.02	0.16%	-	0.95	8.44	13.42	7.47	-	0.00	0.11	0	0.04
ins14	5	242	62	0	12	33	17	3600.03	0.04%	-	1.33	11.09	14.88	10.24	-	0.00	0.18	0	0.1
ins15	6	296	67	0	17	30	20	3600.06	2.57%	-	1.12	9.7	14.50	8.96	-	0.00	0.23	0	0.1
ins16	6	283	70	0	18	34	18	3600.36	4.80%	-	1.06	10	14.94	8.97	-	0.00	0.21	0	0.1
ins17	6	301	72	0	25	23	24	3600.05	1.04%	-	1.2	9.26	14.17	8.1	-	0.00	0.13	0	0.04
ins18	7	347	75	0	23	38	14	3605.05	3.21%	-	1.26	10.45	13.79	8.25	-	0.00	0.18	0	0.09
ins19	7	316	78	0	32	31	15	3600.06	0.36%	-	1.06	9.9	13.07	6.88	-	0.00	0	0	0
ins20	7	332	82	0	23	39	20	3600.05	0.45%	-	1.13	10.59	14.20	8.82	-	0.00	0.28	0	0.13

Table 5: Results of phase 1.

				MIP						CP					
Instance	$\mathcal{K}$	$\mathcal{P}$	$\hat{\mathcal{P}}$	obj.	gap	bound	avg. TW violation	avg. deviation	avg. changes	obj.	gap	bound	avg. TW violation	avg. deviation	avg. changes
ins01	1	36	10	**328**	0.00%	328.00	1.50	16.13	0.00	**328**	0.00%	328.00	1.50	16.13	0.00
ins02	1	55	15	**1,147**	0.00%	1147.00	2.58	32.73	0.00	**1,147**	78.47%	247.00	2.58	32.73	0.00
ins03	1	54	17	**2,209**	0.00%	2209.00	1.43	32.70	0.68	**2,209**	100.00%	0.00	1.43	32.70	0.68
ins04	2	96	15	**2,325**	0.00%	2325.00	0.98	25.00	0.40	2,992	100.00%	0.00	0.99	19.60	0.54
ins05	2	94	16	**3,081**	0.00%	3081.00	4.56	49.17	0.27	3,501	100.00%	0.00	4.34	44.08	0.38
ins06	3	121	30	2,629	24.44%	1986.58	2.52	40.19	0.04	**2,437**	100.00%	0.00	2.35	36.31	0.04
ins07	3	137	33	**5,912**	0.00%	5912.00	2.95	47.92	0.52	7,128	100.00%	0.00	2.86	47.92	0.72
ins08	3	147	35	**3,167**	0.00%	3167.00	1.90	27.81	0.25	3,379	99.73%	9.00	1.84	27.59	0.29
ins09	4	172	42	**6,331**	7.68%	5844.84	2.94	38.24	0.45	11,102	99.95%	6.00	2.87	38.45	1.07
ins10	4	226	47	**24,407**	43.63%	13758.93	3.18	42.15	1.99	36,585	100.00%	0.00	2.86	37.81	3.16
ins11	4	191	52	1,742,662	99.77%	4033.42	14.41	56.63	207.25	**19,394**	100.00%	0.00	2.82	39.53	1.86
ins12	5	259	55	39,575	69.14%	12211.87	2.01	47.94	3.00	**30,871**	100.00%	0.00	1.74	36.91	2.33
ins13	5	228	57	**25,362**	53.07%	11901.97	2.42	54.57	2.08	31,657	100.00%	0.00	1.62	41.92	2.80
ins14	5	242	62	11,504	53.97%	5295.58	2.49	42.78	0.64	**24,383**	100.00%	0.00	1.90	34.54	1.93
ins15	6	296	67	2,897,823	99.77%	6694.22	12.80	67.22	209.59	**33,247**	100.00%	0.00	2.42	47.86	2.04
ins16	6	283	70	290,391	94.93%	14723.17	6.16	67.40	21.98	**35,862**	99.94%	21.00	2.25	39.73	2.47
ins17	6	301	72	340,997	95.48%	15404.76	4.56	62.23	24.30	**68,576**	100.00%	0.00	2.65	42.00	4.67
ins18	7	347	75	105,778	87.04%	13711.29	3.09	54.75	6.12	**37,538**	99.99%	2.00	1.81	42.15	2.06
ins19	7	316	78	155,149	91.35%	13423.60	3.79	59.50	10.42	**29,414**	100.00%	0.00	1.60	31.13	1.85
ins20	7	332	82	267,168	95.81%	11204.17	2.89	64.75	17.33	**31,719**	99.97%	8.00	2.29	36.76	1.79

Table 6: Results of phase 2 by CP and MIP with one hour of run time. The table reports (1) objective value; (2) optimality gap; (3) lower bound; (4) the average violation in time preferences of new curative patients per section (in time blocks); (5) the average deviation of appointment time during the treatment of new curative patient (in time blocks); and (6) the average changes to appointment times of fixed patients (in time blocks).

and *stochastic* (S). The two sets are generated as follows. Given a patient flow, two instances are created for each set, following the procedure in Section 4. The only difference is that for the instance in set *O*, no stochastic factors are introduced when creating the instance scenario, i.e. no appointments are swapped or shifted. Therefore, the scenarios of instances in set *O* are more "orderly", with patients having identical appointment times every day, while those in set *S* are more stochastic, with inconsistency in their partially-filled schedules. Instances in set *S*, hence, are closer to realistic instances. The number of fixed patients each day, for each linac, are the same for both instances. Therefore, there is no difference for phase 1 in solving the two instances. We evaluate the performance of phase 2 in the two instance sets. From 40 instance flows, 80 instances are generated, divided into two sets. All instances consist of one linac. The number of patients ranges from five to eight patients. The planning horizon is set as seven days. Similar to Section 5.1, two algorithms with MIP and CP for phase 2 are tested. Each algorithm is given five minutes of run time. The amount of time taken to obtain optimal solutions is compared. If run time exceeds five minutes, it means that the optimality gap is not closed given the time budget. The results are plotted in Figure 3. If a dot is located in the upper half of the graph, the algorithm takes more time to obtain the optimal solution for the corresponding instance from set *S* compared to set *O*, i.e. the stochastic instance is more difficult to solve. As can be seen from the figure, stochastic instances are significantly more difficult for both solvers (MIP and CP) to solve than orderly instances. This observation is confirmed by a paired

				MIP					CP				
Instance	$\mathcal{K}$	$\mathcal{P}$	$\hat{\mathcal{P}}$	obj.	gap	avg. TW violation	avg. deviation	avg. changes	obj.	bound	avg. TW violation	avg. deviation	avg. changes
ins01	1	36	10	**328**	0.00%	1.50	16.13	0.00	**328**	0.00%	1.50	16.13	0.00
ins02	1	55	15	397,564	99.90%	15.98	75.45	163.28	**1,150**	100.00%	2.53	34.36	0.00
ins03	1	54	17	22,996	96.37%	6.65	70.00	9.24	**2,389**	100.00%	1.43	32.70	0.76
ins04	2	96	15	**2,325**	0.00%	0.98	25.00	0.40	2,992	100.00%	0.99	19.60	0.54
ins05	2	94	16	13,433	91.40%	6.28	58.17	2.36	**6,393**	100.00%	4.38	39.17	1.01
ins06	3	121	30	-	-	-	-	-0.01	**2,863**	100.00%	2.36	36.38	0.12
ins07	3	137	33	1,197,944	100.24%	13.08	66.52	190.66	**40,774**	100.00%	1.69	39.56	6.24
ins08	3	147	35	1,267,286	100.24%	11.48	49.26	187.72	**3,446**	99.74%	1.91	29.07	0.29
ins09	4	172	42	1,605,072	100.20%	17.67	61.00	204.25	**19,513**	99.97%	2.85	39.38	2.15
ins10	4	226	47	1,723,898	100.18%	10.94	61.19	159.77	**89,749**	100.00%	2.31	36.41	8.14
ins11	4	191	52	1,742,662	100.26%	14.41	56.63	207.25	**33,031**	100.00%	2.76	38.30	3.50
ins12	5	259	55	2,378,546	100.16%	5.63	60.61	193.86	**72,368**	100.00%	1.49	32.24	5.75
ins13	5	228	57	2,223,679	100.19%	7.07	74.46	215.88	**74,031**	100.00%	1.49	40.30	6.95
ins14	5	242	62	2,190,090	100.26%	12.99	59.26	201.35	**137,492**	100.00%	1.34	31.16	12.47
ins15	6	296	67	2,897,823	100.20%	12.80	67.22	209.59	**233,343**	100.00%	2.22	38.16	16.66
ins16	6	283	70	2,877,519	100.20%	13.24	59.98	223.91	**212,492**	99.99%	2.78	41.25	16.25
ins17	6	301	72	2,823,252	100.19%	13.69	64.45	204.33	**277,954**	100.00%	2.61	34.70	19.93
ins18	7	347	75	-	-	-	-	-	**280,244**	100.00%	2.03	37.90	16.93
ins19	7	316	78	-	-	-	-	-	**176,977**	100.00%	1.03	25.13	12.24
ins20	7	332	82	3,024,708	100.19%	9.61	61.20	199.72	**251,682**	100.00%	1.97	38.02	16.48

Table 7: Results of phase 2 by CP and MIP with 5 minutes run time.

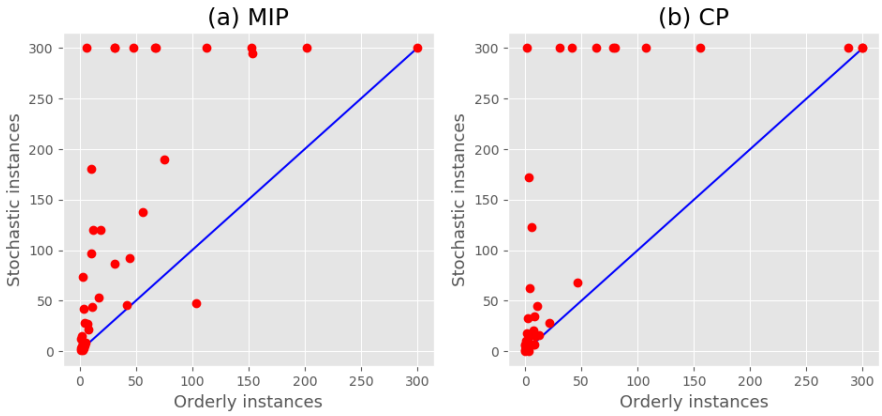

Fig. 3: Comparison of run time (in seconds) in phase 2 for orderly and stochastic instances.

t-test (p-value is 0.00005 for MIP and 0.00024 for CP). We therefore conclude that the presence of stochastic factors in an instance scenario, i.e. the way the partial schedules are filled, strongly affects algorithm performance in phase 2.

6 Simulation to evaluate scheduling policies

In this section, we put the models into use in a simulation where patients arrive at the center daily following a Poisson distribution. We aim to evaluate the effect of different scheduling policies on the waiting time of patients in a long-term, real-world setting. Currently, CHUM employs a sequential policy where appointments are made and linacs are pre-booked once a patient's treatment plan is approved. To examine whether delaying the scheduling decision to get more information for batch scheduling can offer better schedules, the simulation mimics a patient flow of daily hospital admissions and performs batch scheduling at different time points following different policies. The final schedules at the end of the simulation are then compared to evaluate the effect of batch scheduling.

At CHUM, once a patient is admitted, his or her appointments are scheduled manually by a scheduling clerk. For palliative patients, the earliest available slots are selected. For curative patients, the scheduling clerk will usually look for an available slot about two weeks from the current date. In this simulation, we want to compare CHUM's current policy with batch scheduling, as well as evaluate the benefit of delaying appointments for two weeks instead of scheduling patients as soon as possible. With this goal in mind, we propose seven scheduling policies, shown in Table 8. Policy 1, currently used at CHUM, is a greedy heuristic where patients are scheduled one by one at admission, and treatments of curative patients are postponed until a later time. Delaying treatments is enforced by altering the ready date of a

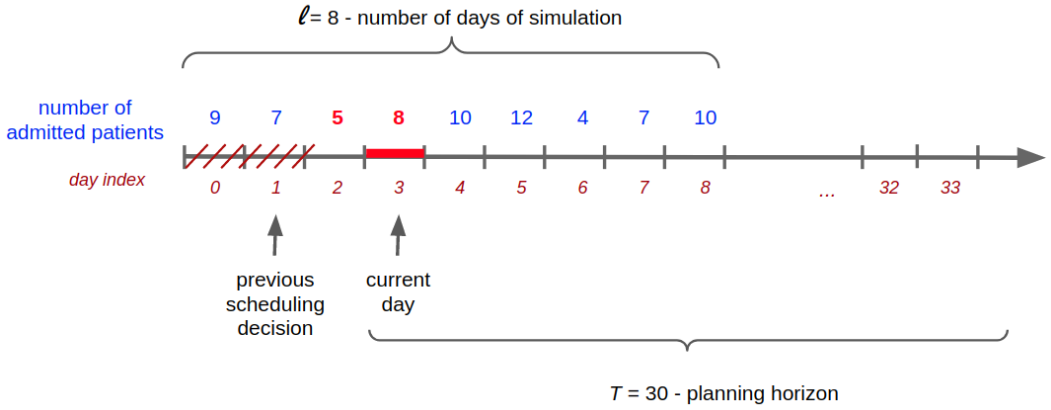

Fig. 4: Simulation process. The previous scheduling decision was at day one. Day three is the current day when the scheduling decision takes place. Thirteen patients admitted on day two and day three will be scheduled.

Policy	Delaying appts.	Scheduling palliative patients	Scheduling curative patients
1	✓	at admission*	at admission*
2		every day	every day
3		every day	every Tuesday & Friday
4		every day	every Friday
5	✓	every day	everyday
6	✓	every day	every Tuesday & Friday
7	✓	every day	every Friday

Table 8: Scheduling policies. The first policy is the greedy heuristic currently used at CHUM.

(curative) patient to the midpoint of his or her ready date and due date, i.e. seven days after admission for patients in category P3 and 14 days after admission for category P4. The remaining six policies apply batch scheduling using the two-phase approach. Palliative patients are scheduled daily, while curative patients are scheduled either daily or on some pre-defined days of the week. The final three policies combine batch scheduling with delaying treatments. In addition, to avoid filling up linacs too quickly at the early scheduling decisions, which may lead to a shortage of space for emergency patients at later scheduling decisions, we reserve a portion of linac capacity for emergency patients. Emergency patients are either palliative patients or curative patients with due dates approaching within two days of the current scheduling decision. Based on the fact that palliative patients account for about 30% (see Table 1) of all patients, we choose to reserve 40% of linac capacity for prioritized patients.

At each scheduling decision, patients to be scheduled include all patients admitted after the previous scheduling decision until the current day. An illustration of the simulation can be found in Figure 4. The scheduling horizon in this experiment is set to 60 days, which is large enough to incorporate all patients admitted at each scheduling decision. Scheduling decisions are repeated until all patients are scheduled.

Forty instances are generated. The arrival rate (λ) ranges from three to ten patients per day and the number of linacs ranges from two to eight linacs. The number of days of simulation (l) is set to 15 days or three work weeks. For each policy, we report the average number of waiting time and overdue time (in days), both overall and by patient category. The results are shown in Table 9. The box plots of the average waiting time and overdue time are shown in Figure 5.

We first compare the greedy heuristic (policy 1) to batch scheduling. As can be seen in the table and the boxplot, policy 1 results in higher waiting time and overdue time compared to policies with batch scheduling. The average waiting time for P4 patients in policy 1 is lower than the others, but this leads to higher waiting times for P1 patients. The average overdue time is likewise much higher compared to the other policies, especially in P1 (3.31 days compared to the lowest 0.41 days in policy 6) and P2 patients (6.42 days compared to the lowest 1.14 days in policy 7).

The remaining policies are divided into two groups. The first group consists of policies without treatment delays (policies $2-4$) while the second group delays treatments to a later time point (policies $5-7$). From the data in the table and the boxplot, there is virtually no difference in the overall average waiting time between the two groups. However, differences do emerge in waiting time according to patient priority. The first group tends to schedule P4 patients much ealier (about 12 days after admission) than the second group (about 18 days). Consequently, palliative patients tend to wait longer in the first group. The average waiting time for P1 patients is around two to three days in the first group, but only around 0.6 days in the second group. Similarly, P2 patients wait for five to eight days on average in the first group and around three days in the second group. Also, the second

Scheduling policy	Waiting time					Overdue time				
	Overall	P1	P2	P3	P4	Overall	P1	P2	P3	P4
1*	11.04	3.45	8.78	12.29	11.90	2.28	3.31	6.42	0.71	0.01
2	10.81	2.94	8.07	11.96	12.31	1.97	2.80	5.65	0.53	0.00
3	10.61	2.62	7.10	12.07	12.51	1.66	2.48	4.72	0.49	0.00
4	**10.22**	2.04	5.82	11.86	12.79	1.21	1.90	3.45	0.33	0.01
5	10.91	0.60	3.57	11.64	18.11	0.61	0.53	1.60	0.31	0.00
6	10.87	0.54	3.37	11.64	18.16	0.53	0.41	1.40	0.29	0.00
7	10.82	0.62	3.11	11.64	18.22	**0.45**	0.55	1.14	0.25	0.00

Table 9: Average waiting time and overdue time by scheduling policy and patient category (in days).

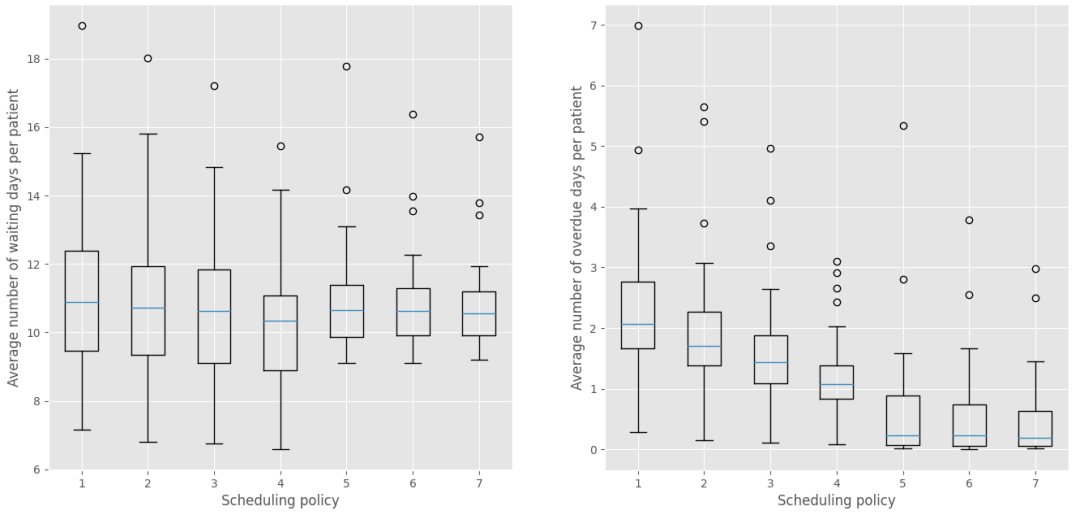

Fig. 5: Average waiting time and overdue time by scheduling policy.

group demonstrates less deviation in waiting time between instances. Regarding overdue time, the second group obviously gives better schedules. Most overdue time occurs in P1 and P2 patients, which can be easily explained by their short treatment deadline. Within each group, batch scheduling helps reduce both waiting time and overdue time. In the first group, daily scheduling (policy 2) yields an average of 10.81 days of waiting time and 1.97 days of overdue time. Batch scheduling once a week (policy 4) reduces those numbers to 10.22 and 1.21 days, respectively. The reduction in waiting time mostly occurs in P1 and P2 patients, while waiting times for P3 and P4 patients are not similarly impacted. Overdue time is most problematic for P2 patients (5.65 days with daily scheduling and 3.45 days with weekly scheduling). In the second group, waiting time seems to be stable in all scheduling policies. Regardless, P4 patients experience virtually no overdue time in either group. Batch scheduling in the second group also helps to reduce overdue time, although to a lesser extent than in the first group. Daily scheduling (policy 5) results in 0.61 days of overdue time, while weekly scheduling (policy 8) results in 0.45 days of overdue time.

On the numerical results from this experiment we have two remarks. First, batch scheduling using our two-phase approach improves the schedules compared to the greedy heuristic currently used by CHUM. Second, these results confirm our intuition that delaying the scheduling decision to get more information extends the search space and hence offers better schedules. This is an important indicator to help the hospitals adjust their policy toward better treatment standards.

7 Conclusions and future work

In this paper, we introduced a two-phase approach for the Radiotherapy Scheduling Problem. An IP model is proposed for phase 1, which decides the most important information of a schedule including patients' starting dates and the linacs for treatments. An MIP and a CP model are proposed for phase 2, which decides the order of patients on each linac and the exact appointment times. Numerical results show that our approach provides good solutions for instances with up to seven linacs, i.e. the instance size that we target in this

paper. The comparison of results given by CP and MIP after a 5-minute run time and one-hour run time shows that CP is able to find good solutions much faster than MIP, but fails to provide good lower bounds compared to MIP. We point out that the versatility of the CP model and its ability to find good solutions quickly make CP a promising approach for real-world applications.

We also present a simulation to evaluate the effect of the two-phase approach on waiting time and overdue time of patients in a long-term, real-world setting. Different scheduling strategies are evaluated. The results show that batch scheduling with different scheduling policies has an effect on the total waiting time and overdue time of the final schedule. These results suggest that the sequential scheduling policy currently employed by CHUM could be replaced by a better decision-making scheme. Further analysis in this regard is one of our future directions.

In another contribution, we present how realistic instances are generated based on real data from a cancer center in Montréal. This contribution is important since the characteristics of test instances strongly impact the performance of the algorithms, which has also been proven in the paper by numerical results.

Acknowledgements We would like to thank Stefan Michalowski for providing us with the radiotherapy treatment data from CHUM and Marc-Andre Renaud for his suggestions on how to improve the performance of our algorithms. This research received funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" program.

References

1. Michael B Barton, Susannah Jacob, Jesmin Shafiq, Karen Wong, Stephen R Thompson, Timothy P Hanna, and Geoff P Delaney. Estimating the demand for radiotherapy from the evidence: a review of changes from 2003 to 2012. *Radiotherapy and oncology*, 112(1):140–144, 2014.
2. Edmund K Burke, Pedro Leite-Rocha, and Sanja Petrovic. An integer linear programming model for the radiotherapy treatment scheduling problem. *arXiv preprint arXiv:1103.3391*, 2011.
3. Elkin Castro and Sanja Petrovic. Combined mathematical programming and heuristics for a radiotherapy pre-treatment scheduling problem. *Journal of Scheduling*, 15(3):333–346, 2012.
4. Zheng Chen, Will King, Robert Pearcey, Marc Kerba, and William J Mackillop. The relationship between waiting time for radiotherapy and clinical outcomes: a systematic review of the literature. *Radiotherapy and Oncology*, 87(1):3–16, 2008.
5. CE Coles, L Burgess, and LT Tan. An audit of delays before and during radical radiotherapy for cervical cancer–effect on tumour cure probability. *Clinical oncology*, 15(2):47–54, 2003.
6. Domenico Conforti, Francesca Guerriero, and Rosita Guido. Optimization models for radiotherapy patient scheduling. *4OR*, 6(3):263–278, 2008.
7. Domenico Conforti, Francesca Guerriero, and Rosita Guido. Non-block scheduling with priority for radiotherapy treatments. *European Journal of Operational Research*, 201(1):289–296, 2010.
8. Sara Frimodig and Christian Schulte. Models for radiation therapy patient scheduling. In *International Conference on Principles and Practice of Constraint Programming*, pages 421–437. Springer, 2019.
9. Yasin Gocgun. Simulation-based approximate policy iteration for dynamic patient scheduling for radiation therapy. *Health care management science*, 21(3):317–325, 2018.
10. Truword Kapamara and Dobrila Petrovic. A heuristics and steepest hill climbing method to scheduling radiotherapy patients. In *Proceedings of the 35th International Conference on Operational Research Applied to Health Services, Catholic University of Leuven, Belgium*. Citeseer, 2009.
11. Philippe Laborie, Jérôme Rogerie, Paul Shaw, and Petr Vilím. Ibm ilog cp optimizer for scheduling. *Constraints*, 23(2):210–250, 2018.
12. SN Larsson. Radiotherapy patient scheduling using a desktop personal computer. *Clinical Oncology*, 5(2):98–101, 1993.
13. Antoine Legrain, Marie-Andrée Fortin, Nadia Lahrichi, and Louis-Martin Rousseau. Online stochastic optimization of radiotherapy patient scheduling. *Health care management science*, 18(2):110–123, 2015.
14. William J. Mackillop. Killing time: the consequences of delays in radiotherapy. *Radiotherapy and Oncology 84.1*, 1(4), 2007.
15. Johannes Maschler and Günther R Raidl. Particle therapy patient scheduling with limited starting time variations of daily treatments. *International Transactions in Operational Research*, 2018.
16. Johannes Maschler, Martin Riedler, Markus Stock, and Günther R Raidl. Particle therapy patient scheduling: First heuristic approaches. In *Proceedings of the 11th Int. Conference on the Practice and Theory of Automated Timetabling. Udine, Italy*, 2016.
17. Sanja Petrovic and Pedro Leite-Rocha. Constructive approaches to radiotherapy scheduling. In *Proceedings of the World Congress on Engineering and Computer Science*, pages 722–727, 2008.
18. Sanja Petrovic, William Leung, Xueyan Song, and Santhanam Sundar. Algorithms for radiotherapy treatment booking. In *25th Workshop of the UK planning and scheduling special interest group*, pages 105–112, 2006.
19. Antoine Saure, Jonathan Patrick, Scott Tyldesley, and Martin L Puterman. Dynamic multi-appointment patient scheduling for radiation therapy. *European Journal of Operational Research*, 223(2):573–584, 2012.
20. Scott Tyldesley, Geoff Delaney, Farshad Foroudi, Lisa Barbera, Marc Kerba, and William Mackillop. Estimating the need for radiotherapy for patients with prostate, breast, and lung cancers: verification of model estimates of need with radiotherapy utilization data from british columbia. *International Journal of Radiation Oncology* Biology* Physics*, 79(5):1507–1515, 2011.
21. Bruno Vieira, Derya Demirtas, Jeroen B van de Kamer, Erwin W Hans, Louis-Martin Rousseau, Nadia Lahrichi, and Wim H van Harten. Radiotherapy treatment scheduling considering time window preferences. *Health care management science*, pages 1–15, 2020.
22. Petra Vogl, Roland Braune, and Karl F Doerner. Scheduling recurring radiotherapy appointments in an ion beam facility. *Journal of Scheduling*, 22(2):137–154, 2019.