Operations Research Forum (2023) 4:83
https://doi.org/10.1007/s43069-023-00251-2

ORIGINAL RESEARCH

Comparing Optimization Methods for Radiation Therapy Patient Scheduling using Different Objectives

Sara Frimodig[1,2] · **Per Enqvist**[1] · **Mats Carlsson**[3] · **Carole Mercier**[4]

Received: 17 March 2022 / Accepted: 26 August 2023 / Published online: 24 October 2023

Abstract

Radiation therapy (RT) is a medical treatment to kill cancer cells or shrink tumors. To manually schedule patients for RT is a time-consuming and challenging task. By the use of optimization, patient schedules for RT can be created automatically. This paper presents a study of different optimization methods for modeling and solving the RT patient scheduling problem, which can be used as decision support when implementing an automatic scheduling algorithm in practice. We introduce an Integer Programming (IP) model, a column generation IP model (CG-IP), and a Constraint Programming model. Patients are scheduled on multiple machine types considering their priority for treatment, session duration and allowed machines. Expected future arrivals of urgent patients are included in the models as placeholder patients. Since different cancer centers can have different scheduling objectives, the models are compared using multiple objective functions, including minimizing waiting times, and maximizing the fulfillment of patients' preferences for treatment times. The test data is generated from historical data from Iridium Netwerk, Belgium's largest cancer center with 10 linear accelerators. The results demonstrate that the CG-IP model can solve all the different problem instances to a mean optimality gap of less than 1% within one hour. The proposed methodology provides a tool for automated scheduling of RT treatments and can be generally applied to RT centers.

Keywords Patient scheduling · Radiation therapy · Integer programming · Constraint programming · Column generation

✉ Sara Frimodig
sarhal@kth.se

[1] Department of Mathematics, KTH Royal Institute of Technology, Stockholm, Sweden

[2] RaySearch Laboratories, Stockholm, Sweden

[3] Department of Computer Science, RISE Research Institutes of Sweden, Gothenburg, Sweden

[4] Department of Radiation Oncology, Iridium Netwerk, Antwerp, Belgium

1 Introduction

Radiation therapy (RT) is a cancer treatment that uses radiation to kill malignant tumor cells. Together with chemotherapy and surgery, it is one of the most commonly used cancer therapies worldwide. Based on demographic changes such as an aging population, cancer incidents are increasing, and a 16% expected increase in the number of RT treatment courses in Europe has been estimated from 2012 to 2025 [1].

A long waiting time between referral and treatment start has negative effects on the outcome of the treatment. Reasons for this include tumor growth, psychological distress of the patient, and prolonged symptoms when the waiting times are long [2–6]. Therefore, many cancer institutes around the world have adopted waiting time targets that state the maximum allowed waiting time before treatment starts.

The RT treatment intent can be divided into either curative or palliative, where the first intends to cure the patient, and the latter mainly aims to provide symptom relief. Furthermore, cancer patients are often divided into different urgency levels depending on the site of the cancer, treatment intent, and the size and progress of the tumor. The prioritization of patients for treatment can be done in different ways in different countries or hospitals [7–10].

There are several types of RT, where external photon beam RT is by far the most common, and the one covered in this paper. Photon beam RT is delivered on machines called linear accelerators (*linacs*). Because the DNA of healthy cells is repaired to a higher degree than that of malignant cells, the radiotherapy treatment is usually divided into multiple sessions, called *fractions*. The fractions are scheduled daily with breaks on the weekends for a period of up to eight weeks. The duration of the fractions varies between patients due to treatment technique and treatment complexity. For a particular treatment, the delivery time of all fractions is the same. However, the first fraction is usually scheduled for a longer time since it includes setup times, extra time for patient education and reassurance, and additional quality checks before treatment start [11].

In the RT patient scheduling problem, the aim is to schedule RT treatments for a set of patients, given a set of linacs, for a certain planning horizon. The problem is complicated since the patients are of different priorities, and their treatments vary in the number of fractions, fraction durations and set of compatible linacs. Furthermore, the RT process includes many uncertainties, such as the random arrival of new patients.

This paper considers the RT scheduling problem arising at Iridium Netwerk, an RT center located in Antwerp, Belgium. In 2020, they operated 10 linacs, delivering 5500 RT treatments to approximately 4000 patients. The scheduling at Iridium is today done manually. Designing more efficient schedules would be of great significance as it could potentially improve patient outcomes by shortening waiting times. For this reason, this paper makes the following contributions:

– The *main contribution* is a comparative study of the performance of three exact optimization approaches to the RT scheduling problem. By evaluating the suitability of different optimization technology for the problem, this serves as a

foundation for further research in the area, and more importantly, as a decision support when implementing an automatic scheduling algorithm in practice.
- The main *technical novelty* lies in the original models developed: an integer linear programming (IP) model, a column generation IP (CG-IP) model, and a constraint programming (CP) model, as well as a method combining the CP and IP models. To the best of our knowledge, these models are the first to simultaneously assign all fractions of the patients to both linacs and specific time windows, while including all the medical and technical constraints necessary for the scheduling to work in practice. Furthermore, it is the first time column generation has been used for the RT patient scheduling problem. The problem instances solved are larger than in previous studies in terms of number of linacs, number of patients and length of the planning horizon.
- Different cancer centers may have different goals when creating the RT schedules. In order to study the suitability of the above-mentioned optimization models for various cancer centers, each model is solved using multiple different objective functions. Six different objectives are evaluated, such as minimizing waiting times and maximizing the satisfaction of time window preferences among the patients. Furthermore, a sensitivity analysis and a study of competing objectives is performed to further evaluate the modeling approach.

The paper is organized as follows. Section 2 presents the related work. Section 3 describes the problem. Section 4 presents the models. The setup for the computational experiments is explained in Section 5, followed by the results in Section 6. Section 7 contains the discussion, and Section 8 presents the conclusions.

2 Related Work

A literature review on the use of operations research for resource planning in RT was published in 2016 by Vieira et al. [12]. The authors found 12 papers addressing the problem of scheduling RT patients on linacs. The first use of integer programming (IP) for optimization of RT appointments was published in 2008 by Conforti et al. [13], where a block scheduling model is presented. The day is divided into blocks of equal duration and each treatment is assigned to one block. The same authors later developed a non-block scheduling model, which allows for different treatment durations [14]. Another IP model for non-block scheduling is presented by Jacquemin et al. [15], where the notion of treatment patterns is introduced to allow non-consecutive treatment days. A limitation of these papers is that they do not consider all the constraints present in RT scheduling, such as multiple machine types and partial availability in the schedule.

Sauré et al. [16] present a method for advance RT patient scheduling using a discounted infinite-horizon Markov decision process, and show that their proposed policy can increase the percentage of treatments initiated within 10 days from 73% to 96%. In [17], Gocgun extends the same problem setup to also include patient cancellations. The setting used in these papers is a simplified model of a cancer center, equipped with three identical machines and 18 treatment types. The

resulting policies assign a start day to each patient, with no sequencing of patients throughout the day.

In order to schedule RT appointments one by one in an online fashion, Legrain et al. [18] propose a hybrid method combining stochastic and online optimization using a block-scheduling strategy. The results show that their method works well on small real instances, with two linacs and less than 3.5 requests per day. Aringhieri et al. [19] also present methods for online RT scheduling, and develop three online optimization algorithms for a block-scheduling formulation and one machine.

Li et al. [20] model the RT patient scheduling as a queueing system with multiple queues. A new class of scheduling policies is proposed, where the parameters are selected through simulation-based optimization heuristics. All treatments are assumed to have the same length, and all machines are assumed to be identical.

The type of RT that uses high-energy particles (protons or helium ions) is referred to as particle therapy (PT), in which a single particle beam is shared between multiple treatment rooms. This gives very different technical and medical constraints than in conventional photon beam RT. Two papers that present methods for optimizing the sequencing of patients throughout the day in PT are Vogl et al. [21] and Maschler et al. [22], both aiming to schedule treatments close to a pre-defined target time and both using different heuristic methods. Braune et al. [23] present a model for planning appointment times in PT under uncertain activity durations, and solve the resulting stochastic optimization model using a combination of a Genetic Algorithm and Monte Carlo simulation.

The first paper to include the sequencing of patient throughout the day in photon beam RT was published in 2020 by Vieira et al. [24]. The authors create weekly schedules with the objective to maximize the fulfillment of the patients' time window preferences using a mixed-integer programming (MIP) model together with a preprocessing heuristic to divide the problem into subproblems for clusters of machines. In a second paper they test their method in two Dutch clinics [25], with results showing that the weekly schedule was improved in both centers. However, the problem studied in these papers is different from the one in this paper, as they create weekly schedules for a time horizon of five days, whereas we aim to assign all fractions to machines and therefore have a significantly longer time horizon (typically around 80 days). Using a data-driven approach, Moradi et al. [26] study the patient sequencing problem in a simplified clinical setup, where all treatment durations are equal and all machines are identical and independent. To improve the weekly schedules, the authors utilize patient information in a MIP model to determine the optimal sequence of patients for a list of patients that have been previously assigned to a treatment day. The results show that it is favorable to schedule reliable patients early on to reduce idle time on machines caused by delayed patients or no-shows.

Constraint Programming (CP) is a technique for solving combinatorial problems with origins in computer science and artificial intelligence. CP solvers use rule-based inference, logical reasoning, and search techniques. For an overview, see [27]. CP has been used in RT treatment planning [28], in chemotherapy patient scheduling [29] and in operating room scheduling [30]. Overall, scheduling is a field where CP has shown to be effective, see for example [31]. Frimodig et al. [32] present and compare two CP models and one IP model for the RT scheduling problem. The

CP models are shown to be efficient at finding feasible solutions, but are generally slower than the IP model at proving optimality. A limitation for these models is that they only consider one machine type.

Pham et al. [33] propose a two-stage approach for the RT scheduling problem. In a first phase, an IP model is used to assign patients to linacs and days, and in the second phase the patients' specific appointment times are decided using either a MIP or a CP model. The test data is generated based on data from CHUM, a cancer center in Canada. The test instances have seven linacs and a time horizon of 60 days. The results in the second phase show that CP finds good solutions faster, while MIP is better at closing optimality gaps, which agrees with the results in [32]. Some simplifications in their models are that all patients can be treated on all machines, that all machine switches are allowed, and that the first fraction has the same duration as the rest. These assumptions make the models less general than the ones presented in this paper, and not suited for the scheduling problem at Iridium Netwerk.

Column generation (CG) is a method that is often successful in solving certain classes of large scale integer programs. The method alternates between a restricted master problem and a column generation subproblem. CG has been applied in various areas within the medical treatment field, such as for surgeon and surgery scheduling [34], for patient admission [35], and for nurse scheduling [36]. In RT, it has been used for brachytherapy scheduling using deteriorating treatment times [37, 38]. In brachytherapy, the radiation is produced from a radioactive source placed within the patient, and the problem differs significantly from patient scheduling in conventional RT.

3 Problem Formulation

The RT scheduling problem consists of assigning each fraction for each patient to a day, a time window, and a machine. This section presents the real-world constraints and objectives present at Iridium Netwerk.

Patients A *priority* is assigned to each patient based on urgency and treatment intent. The prioritization can be done differently in different countries or hospitals [7–10], but at Iridium Netwerk it is done by a physician. In 2020 at Iridium, there were three priority groups, and approximately 42% patients were priority A, 18% priority B, and 40% priority C. Furthermore, each patient is assigned to a *treatment protocol*, which states the *fractionation scheme* (that is, how many days the patient is to be treated and with which frequency), and the *duration* of the first and subsequent fractions. An example of a fractionation scheme is shown in Fig. 1. Different protocols have different allowed *start days*: palliative patients can start any weekday, whereas curative patients cannot start on Fridays. Both the number of priority groups and the allowed start days are easily generalizable in the models. Examples of some different treatment protocols can be seen in Table 1.

Urgent patients must start treatment soon after arrival. Since the patients are of different priority groups, and the fractionation schemes typically span multiple weeks, this must be considered when creating the schedules. In practice, most clinics reserve empty time slots on each machine for urgent patients. In this paper, the

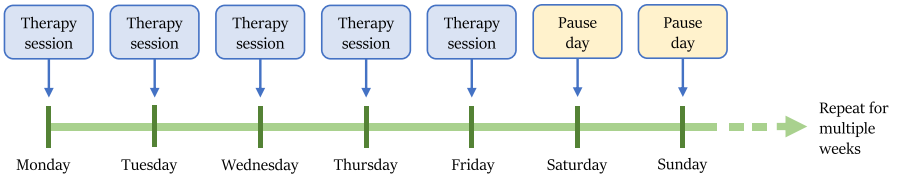

Fig. 1 A typical fractionation scheme of an RT patient

expected value of the future arrivals of urgent patients for the coming weeks are included in the models as placeholder patients (i.e., *dummy* patients) to predict the expected utilization of resources. The models are deterministic, and the placeholder patients are handled as regular priority A-patients in the patient list, with earliest start day set from the expected arrival day.

Time RT clinics have different routines for scheduling patients on linacs. Some gather patients into a *batch* and schedule them once or several times per day, while others schedule each patient at admission. In [33], different scheduling strategies are evaluated using a simulation. Preliminary results show that daily batch scheduling reduces patients' waiting time and overdue time. This paper focuses on batch scheduling and assumes that the scheduling is done at the end of each day taking patients from previous days into account.

There are two different scheduling systems used at RT centers: *block* or *non-block* scheduling systems. In the block system, the day is divided into blocks of equal duration and each patient is assigned to a block. This is commonly used in clinics, but has severe drawbacks since there is no way to control the variability of treatment time, which can generate costs related to machine under utilization, staff overtime, and patient waiting time. This paper uses a *non-block* scheduling strategy. The day is divided into *time windows* that are typically $2-4$ hours long, while a treatment duration is $10-60$ minutes depending on the type of treatment. This is different from a block scheduling system where each treatment is assumed to have the same duration as one or multiple blocks. Patients are assigned to windows instead of specific start times, as this leads to simpler and more efficient models while maintaining an adequate level of detail from a clinical perspective. The specific start time for each treatment within the time window is given in a post-processing step.

During the first treatment fraction, extra time is needed for both instructing the patient and for linac setup [39]. Therefore, auxiliary time must be assigned to each new patient, which is done by assigning the first fraction to a longer time duration (see Table 1). Furthermore, at Iridium no patients are treated during weekends. This fact is used to simplify the models; the time horizon is adjusted to only contain weekdays ($\mathcal{D}_w$). In general, at most cancer centers in the world only patients undergoing an oncologic emergency are treated during weekends, and in that case, the care is not planned more than a day in advance [40, 41].

Table 1 Examples of some different treatment protocols at Iridium Netwerk

Protocol	Priority	Duration first fraction (minutes)	Duration other fractions (minutes)	Average number fractions	Preferred machines	Allowed machines
Bladder (VMAT)	B	24	12	20	M2, M3, M5, M6, M7, M9	M1, M4, M8
Brain STX 3x	A	40	40	3	M10	
Breast bilateral	C	48	24	13	M1, M3, M4, M5, M6, M8	M2, M7
Head-Neck (VMAT)	A	24	12	26	M2, M3, M5, M6, M9	M1, M4
Prostate SBRT	C	24	12	5	M10	
Rectum 25x	B	24	12	25	M2, M3, M5, M6, M8	M1, M4, M7

Table 2 Beam-matches machines at Iridium Netwerk

Machine	Completely matched	Partially matched
M1	-	M4, M8
M2	-	M3, M5, M6, M7, M9
M3	M9	M2, M5, M6, M7
M4	-	M1, M8
M5	M6	M2, M3, M7, M9
M6	M5	M2, M3, M7, M9
M7	-	M2, M3, M5, M6, M9
M8	-	M1, M4
M9	M3	M2, M5, M6, M7
M10	-	-

Machines The radiation is delivered on linacs. This paper assumes that there are multiple machine types, which is the case in almost all clinics. At Iridium there are ten linacs distributed over four different hospitals. The treatment protocol for each patient states one or more machines that can deliver the protocol, however, some machines are preferred over the others. An example can be seen in Table 1.

Some linacs are so called *beam-matched*, meaning that a patient can switch between these linacs between fractions. Two linacs are considered completely beam-matched if they are the same machine type at the same hospital, and partially beam-matched if they are the same machine type, but at different hospitals. Switching between completely beam-matched machines can be done at no cost, whereas there is a cost for switching to a machine that is only a partially matched. The beam-matched machines are presented in Table 2. To generalize the models, the cost for machine switches between partially beam-matched machines is not active in all objective functions investigated.

Objectives Different RT centers can have different scheduling objectives. In order to evaluate how suitable the different optimization models are for solving the RT scheduling problem, the models are tested for multiple objective functions to evaluate their performance, and also to evaluate if some particular model is better for a certain scheduling objective. The following objectives will be tested in different combinations:

(i) Minimize a weighted sum of the waiting times
(ii) Minimize a weighted sum of the violations of the target dates
(iii) Minimize the number of time window switches
(iv) Minimize violations of time window preferences
(v) Minimize the number of fractions scheduled on non-preferred machines
(vi) Minimize the number of switches between machines that are not completely beam-matched

The weights in the weighted sums in (i) and (ii) should reflect the severeness of delaying treatment start for the different priority groups. In objective (ii), the waiting time targets are assumed to be 2 days for priority A, 14 days for priority B, and 28 days for priority C patients, but this is easily generalizable. The waiting time targets differ between countries, and advanced methods to determine the waiting time targets have recently been studied [42]. The aim for objective (iii) is to schedule patients at approximately the same time each day, since this is something the booking administrators usually try to do. Literature shows that patients have different preferences regarding the time of their appointments [43], which is what should be captured in objective (iv). As many fractions as possible should be scheduled on the machines preferred for the particular treatment, hence objective (5) states that the number of fractions scheduled on a non-preferred machine should be minimized. Finally in objective (vi), the number of switches between machines that are not completely beam-matched should be minimized. In Section 5.3, the objectives will be combined into different objective functions. For example, a combination of (i)-(vi), with (i) being most important, is most similar to what is used at Iridium Netwerk.

4 Models

Multiple models are developed: an IP model, a CG-IP model, and a CP model, as well as a method combining CP and IP. They are designed to capture the same real-world constraints and objectives. Their inputs are presented in Table 3. As stated in Section 3, no patients are treated during weekends. Therefore, the time horizon is adjusted so that $\mathcal{D}_w$ only contains weekdays.

4.1 Integer Programming Model

The variables in the IP model are presented in Table 4 and the formulation is stated in (1)–(19).

Constraints Constraint (2) is formulated to ensure that all fractions are scheduled after each other, and that they are all scheduled on beam-matched machines. Constraint (3) forces the fth fraction to be scheduled exactly one time for each patient. Constraint (4) states that the first fraction for patient p is scheduled on machine m on day d, in any window, whereas Constraint (5) also gives the correct window w for the first fraction.

$$\text{minimize} \quad 1 + \sum_{p \in \mathcal{P}} (\alpha_1 f_{1,p} + \alpha_2 f_{2,p} + \alpha_3 f_{3,p} + \alpha_4 f_{4,p} + \alpha_5 f_{5,p} + \alpha_6 f_{6,p}) \tag{1}$$

$$\text{subject to} \quad \sum_{m \in b_{\mathcal{M}}} q_{p,m,d,f} = \sum_{m \in b_{\mathcal{M}}} q_{p,m,d+1,f+1}, \quad \forall p \in \mathcal{P}, b_{\mathcal{M}} \in \mathcal{B}_{\mathcal{M}}, \; d = \{1, \dots, D_w - 1\}, f = \{1, \dots, F_p - 1\} \tag{2}$$

$$\sum_{m \in \mathcal{M}} \sum_{d \in \mathcal{D}_w} q_{p,m,d,f} = 1, \quad \forall p \in \mathcal{P}, f \in \mathcal{F}_p \tag{3}$$

Table 3 Notations for the models

Parameter	Description
$\mathcal{P} = \{1, \dots, P\}$	Set of all patients, $P \in \mathbb{N}$
$\mathcal{P}_h \subset \mathcal{P}$	Set of patients treated with protocol $h \in \mathcal{H}$
$\mathcal{D}_w = \{1, \dots, D_w\}$	Set of weekdays. $D_w \in \mathbb{N}$ is the number of weekdays in the planning horizon
$\mathcal{W} = \{1, \dots, W\}$	Set of time windows in a day, $W \in \mathbb{N}$
$L_w \in \mathbb{N}$	The window length for window $w \in \mathcal{W}$ in number of minutes
$\mathcal{M} = \{1, \dots, M\}$	Set of machines, $M \in \mathbb{N}$
$\mathcal{M}_p \subseteq \mathcal{M}$	Set of machines allowed for patient p
$\mathcal{M}_p^{pref} \subseteq \mathcal{M}_p$	Set of machines preferred for patient p
$\mathcal{C}_\mathcal{M}$	List of sets of completely beam-matched machines
$\mathcal{P}_\mathcal{M}$	List of sets of partially beam-matched machines
$\mathcal{B}_\mathcal{M} = \mathcal{C}_\mathcal{M} \cup \mathcal{P}_\mathcal{M}$	List of sets of all beam-matched machines
$\mathcal{H} = \{1, \dots, H\}$	Set of treatment protocols, $H = 72$
$dur_{p0} \in \mathbb{N}$	Duration of first fraction for patient p (minutes)
$dur_p \in \mathbb{N}$	Duration of fractions other than first for patient p (minutes)
$\mathcal{F}_p \in \{1, \dots, F_p\}$	Set of all fractions for patient p. $F_p \in \mathbb{N}$ is the number of fractions
$S \in \mathbb{R}^M \times \mathbb{R}^D \times \mathbb{R}^W$	The number of occupied timeslots in each window, machine and day, i.e. $S_{m,d,w} \in \{0, ..., L_w\}$
$\mathcal{A}_p \in \mathcal{D}_w$	The set of allowed start days for the protocol of patient p
$c_p \in \{10, 3, 1\}$	Weights for patient p in priority group A, B or C
$d_{L,p} \in \mathcal{D}_w$	The day limit for latest allowed treatment start for patient p, adjusted for days already waited
$d_{\min,p} \in \mathcal{D}_w$	The earliest day for patient p to be scheduled
$\mathcal{P}^{pref} \subset \mathcal{P}$	The set of patients that have a time window preference
$w_p^{pref} \in \mathcal{W}$	The window preference of patient $p \in \mathcal{P}^{pref}$

Table 4 Variables in the IP model

$q_{p,m,d,f} \in \{0, 1\}$	1 if patient $p \in \mathcal{P}$ has their fth fraction ($f \in \mathcal{F}_p$) on weekday $d \in \mathcal{D}_w$ on machine $m \in \mathcal{M}$, 0 otherwise
$x_{p,m,d,w} \in \{0, 1\}$	1 if patient $p \in \mathcal{P}$ is scheduled in window $w \in \mathcal{W}$ on machine $m \in \mathcal{M}$ on weekday $d \in \mathcal{D}_w$, 0 otherwise
$t_{p,m,d,w} \in \{0, 1\}$	1 if patient $p \in \mathcal{P}$ starts treatment in window $w \in \mathcal{W}$ on machine $m \in \mathcal{M}$ on weekday $d \in \mathcal{D}_w$, 0 otherwise
$y_{p,d,w} \in \{0, 1\}$	1 if patient $p \in \mathcal{P}$ is scheduled in window $w \in \mathcal{W}$ on day $d \in \mathcal{D}_w$
$z_{p,d} \in \{0, 1\}$	1 if patient $p \in \mathcal{P}$ has switched windows from day d to $d + 1$, 0 otherwise
$u_{p,d} \in \{0, \dots, W - 1\}$	The violation of the time window preference for patient $p \in \mathcal{P}$ on day $d \in \mathcal{D}_w$
$v_{p,f} \in \{0, 1\}$	1 if patient $p \in \mathcal{P}$ is scheduled on non-preferred machine on fraction $f \in \mathcal{F}_p$, 0 otherwise
$s_{p,f} \in \{0, 1\}$	1 if patient $p \in \mathcal{P}$ switches to a partially beam-matched machine from fraction f to $f + 1$, 0 otherwise

$$q_{p,m,d,1} = \sum_{w \in \mathcal{W}} t_{p,m,d,w}, \quad \forall p \in \mathcal{P}, m \in \mathcal{M}, d \in \mathcal{D}_w \tag{4}$$

$$t_{p,m,d,w} \leq x_{p,m,d,w}, \quad \forall p \in \mathcal{P}, m \in \mathcal{M}, d \in \mathcal{D}_w, w \in \mathcal{W} \tag{5}$$

$$q_{p,m,d,1} = 0, \quad \begin{array}{r} \forall p \in \mathcal{P}, m \in \mathcal{M}, d \in \mathcal{D}_w \text{ if } d > d_{\max,p} \\ \text{or } d < d_{\min,p} \text{ or } d \notin \mathcal{A}_p \end{array} \tag{6}$$

$$q_{p,m,d,f} = 0, \quad \begin{array}{r} \forall p \in \mathcal{P}, m \in \mathcal{M} \text{ if } m \notin \mathcal{M}_p, d \in \mathcal{D}_w, \\ f \in \mathcal{F}_p \text{ if } f \notin \mathcal{F}_{p,d} \end{array} \tag{7}$$

$$\sum_{w \in \mathcal{W}} x_{p,m,d,w} = \sum_{f \in \mathcal{F}_p} q_{p,m,d,f}, \quad \forall p \in \mathcal{P}, m \in \mathcal{M}, d \in \mathcal{D}_w \tag{8}$$

$$\sum_{p \in \mathcal{P}} \Big((x_{p,m,d,w} - t_{p,m,d,w}) dur_p + t_{p,m,d,w} dur_{p0} \Big) + S_{m,d,w} \leq L_w, \quad \forall m \in \mathcal{M}, d \in \mathcal{D}_w, w \in \mathcal{W} \tag{9}$$

$$\sum_{d \in \mathcal{D}_w} d \sum_{m \in \mathcal{M}} q_{p,m,d,1} \leq \sum_{d \in \mathcal{D}_w} d \sum_{m \in \mathcal{M}} q_{p+1,m,d,1}, \quad \forall h \in \mathcal{H}, p \in \mathcal{P}_h \text{ where } d_{L,p} \leq d_{L,p+1} \tag{10}$$

$$y_{p,d,w} \geq \sum_{m \in \mathcal{M}} x_{p,m,d,w}, \quad \forall p \in \mathcal{P}, d \in \mathcal{D}_w, w \in \mathcal{W} \tag{11}$$

$$\sum_{w \in \mathcal{W}} y_{p,d,w} = 1, \quad \forall p \in \mathcal{P}, d \in \mathcal{D}_w \tag{12}$$

$$z_{p,d} \geq y_{p,d,w} - y_{p,d+1,w}, \quad \forall p \in \mathcal{P}, d = \{1, \ldots, D_w - 1\}, w \in \mathcal{W} \tag{13}$$

$$z_{p,d} \geq y_{p,d+1,w} - y_{p,d,w}, \quad \forall p \in \mathcal{P}, d = \{1, \ldots, D_w - 1\}, w \in \mathcal{W} \tag{14}$$

$$u_{p,d} = \sum_{m \in \mathcal{M}} \sum_{w \in \mathcal{W}} x_{p,m,d,w} |w - w_p^{pref}|, \quad \forall p \in \mathcal{P}^{pref}, d \in \mathcal{D}_w, \tag{15}$$

$$u_{p,d} = 0, \quad \forall p \notin p \in \mathcal{P}^{pref}, d \in \mathcal{D}_w, \tag{16}$$

$$s_{p,f} \geq \sum_{d \in \mathcal{D}_w} \sum_{m \in c_{\mathcal{M}}} (q_{p,m,d,f} - q_{p,m,d,f+1}), \quad \forall p \in \mathcal{P}, c_{\mathcal{M}} \in \mathcal{C}_{\mathcal{M}}, f = \{1, \ldots, F_p - 1\} \tag{17}$$

$$v_{p,f} = \sum_{m \in \mathcal{M}} \sum_{d \in \mathcal{D}_w} q_{p,m,d,f} \mathbb{1}_{(m \notin \mathcal{M}_p^{pref})}, \quad \forall p \in \mathcal{P}, f \in \mathcal{F}_p \tag{18}$$

$$x \in \{0,1\}, q \in \{0,1\}, y \in \{0,1\}, z \in \{0,1\}, u \text{ integer}, s \in \{0,1\}, v \in \{0,1\} \tag{19}$$

The earliest day to start treatment is $d_{\min,p}$, and the last day to start treatment is $d_{\max,p} = D_w - F_p + 1$. A treatment can only start on an allowed start day given by $\mathcal{A}_p$. Furthermore, patient p can only be scheduled on a machine that is allowed for the patient protocol given by $\mathcal{M}_p$. Finally, the set of allowed day-fraction pairs for patient p is denoted $\mathcal{F}_{p,d} = \{d \in \mathcal{D}_w, f \in \mathcal{F}_p : f < d \text{ and } d - f < D_w - F_p\}$ (e.g., cannot schedule fraction 2 on day 1, or fraction 3 on day 50 if $F_p > 3$ and the planning horizon $D_w = 50$). In total, this is captured in Constraints (6) and (7).

Constraint (8) states that each patient is scheduled in exactly one time window for each fraction. Constraint (9) ensures that all treatments fit within each time window. The first term sums the session duration of all patients scheduled in window w on machine m on day d, except if it is the first fraction (since it will then evaluate to zero). The second term sums the durations of first fractions for patients starting their treatment in window w on machine m on day d. The sum of all scheduled patients' durations plus the already occupied time slots $S_{m,d,w}$ in that window should be less than or equal the window length L_w.

For two patients with the same treatment protocol, the one with shorter day limit should start treatment first. Constraint (10) enforces this by multiplying the variable $q_{p,m,d,1}$ with the day to get the start day, and force the ordering of the start days. Note the abuse of notation, where $p + 1$ denotes the next entry in $\mathcal{P}_h$.

Objective Function The objective functions (i)–(vi) presented in Section 3 are formulated. An offset set to 1 is included to enable computation of the relative gap also when the optimal value is zero. The different objectives are combined with weights $\alpha_1, \ldots, \alpha_6$ in (1).

Objective (i) is to minimize a weighted sum of the waiting times, which is formulated in (20). The number of waiting days after $d_{\min,p}$, the earliest day to be scheduled for patient p, are linearly penalized with weight c_p corresponding to the priority group of patient p.

$$f_{1,p} = c_p \sum_{m \in \mathcal{M}} \sum_{d=d_{\min,p}}^{D_w} q_{p,m,d,1}(d - d_{\min,p}) \tag{20}$$

Objective (ii) is to minimize a weighted sum of the violations of the waiting time targets, formulated in objective (21). The days past the waiting time target $d_{L,p}$ are linearly penalized with weight c_p.

$$f_{2,p} = c_p \sum_{m \in \mathcal{M}} \sum_{d=d_{L,p}}^{D_w} q_{p,m,d,1}(d - d_{L,p}) \tag{21}$$

Objective (iii) is to minimize the number of time window switches for each patient. Therefore, the variable $y_{p,d,w}$ is defined according to Equations (11) and (12), stating that if patient p is scheduled on day d, $y_{p,d,w} = 1$ only for the window where p is scheduled. Every time window switch between two days is computed by $|y_{p,d,w} - y_{p,d+1,w}| \quad \forall p \in \mathcal{P}, d \in \mathcal{D}_w, w \in \mathcal{W}$. To avoid having the absolute value in the objective function, the variable $z_{p,d}$ is instead defined according to Constraints (13) and (14) and used in the objective function (22).

$$f_{3,p} = \sum_{d \in \mathcal{D}_w} z_{p,d} \tag{22}$$

To form the objective corresponding to (iv), the variable $u_{p,d}$ is defined by Constraints (15) and (16). The time preference violation is zero if the patient does not have a preference, and is otherwise measured by the deviation from the preference on each day the patient is scheduled. Summing all violations gives (23).

$$f_{4,p} = \sum_{d \in \mathcal{D}_w} u_{p,d} \tag{23}$$

Objective (v) is to minimize the number of fractions scheduled on a non-preferred machine stated by the treatment protocol. Therefore, the variable $v_{p,f}$ is introduced and (18) is used to compute the fractions where a patient is scheduled on a non-preferred machine. The preference violations are summed in (24).

$$f_{5,p} = \sum_{f \in \mathcal{F}_p} v_{p,f} \tag{24}$$

There is a cost for switching to a machine that is only a partially beam-matched, which should be minimized according to objective (vi). If fraction f is scheduled on a machine in a group of completely beam-matched machines, but $f + 1$ is not, then it must be scheduled on a partially matched machine by (2). The variable $s_{p,f}$ is one if there is a switch to a partially matched machine, enforced by constraint (17). All machine switches to partially matched machines are summed in (25).

$$f_{6,p} = \sum_{f=1}^{F_p - 1} s_{p,f} \tag{25}$$

4.2 Column Generation IP Model

The problem is reformulated as a set covering model, where the decision variables represent schedules for each patient. Each patient has an associated index set $\mathcal{K}_p$ of feasible schedules, and the variable $a_{p,i} = 1$ if schedule $i \in \mathcal{K}_p$ is allocated to $p \in \mathcal{P}$, and 0 otherwise. Since generating all feasible schedules would be too expensive, a column generation model is presented, which consists of a (restricted) master problem and one subproblem for each patient $p \in \mathcal{P}$. The master problem is the schedule

1: Generate a reduced set of schedules $\mathcal{K}'_p$ for each patient $p \in \mathcal{P}$ using Algorithm 2
2: **while** schedule with negative reduced cost exists **do**
3: Solve linear relaxation of restricted master problem
4: **for** $p \in \mathcal{P}$ **do**
5: Update subproblem objective (33) with dual variables λ, γ and η from continuous restricted master problem solution and solve
6: **if** negative reduced cost **then**
7: Add new schedule to $\mathcal{K}'_p$. Coefficients of the new column are given by the optimal solution vector to the subproblem
8: Solve restricted master problem using integer values

Algorithm 1 Column generation

selection problem, which is solved to make the overall schedule feasible and optimal. In the subproblems, for each patient a new schedule is generated that fulfills all medical and technical constraints, and the sets of feasible schedules are dynamically updated by the column generation procedure presented in Algorithm 1. The algorithm gives a nearly optimal solution, but since the problem is converted from a linear program to an IP in the last step, some schedules not generated by the procedure could potentially improve the integer solutions. The algorithm to generate the initial schedules is presented in Algorithm 2. The number of initial schedules is set to 75, as a larger number does not seem to decrease solution times.

$$\text{minimize} \quad 1 + \sum_{p \in \mathcal{P}} \sum_{i \in \mathcal{K}_p} c_{p,i} a_{p,i} \tag{26}$$

$$\text{subject to} \quad \sum_{i \in \mathcal{K}_p} a_{p,i} = 1 \quad \forall p \in \mathcal{P} \tag{27}$$

$$\sum_{p \in \mathcal{P}} \sum_{i \in \mathcal{K}_p} a_{p,i} \left((x^i_{p,m,d,w} - t^i_{p,m,d,w}) dur_p + t^i_{p,m,d,w} dur_{p0} \right) + S_{m,d,w} \leq L_w, \quad \forall m \in \mathcal{M}, d \in \mathcal{D}_w, w \in \mathcal{W} \tag{28}$$

$$\sum_{i \in \mathcal{K}_p} a_{p,i} \sum_{d \in \mathcal{D}_w} d \sum_{m \in \mathcal{M}} \sum_{w \in \mathcal{W}} t^i_{p,m,d,w} \leq \sum_{i \in \mathcal{K}_{p+1}} a_{p+1,i} \sum_{d \in \mathcal{D}_w} d \sum_{m \in \mathcal{M}} \sum_{w \in \mathcal{W}} t^i_{p+1,m,d,w} \quad \forall h \in \mathcal{H}, p \in \mathcal{P}_h \text{ where } d_{L,p} \leq d_{L,p+1} \tag{29}$$

$$a_{p,i} \in \{0, 1\} \quad \forall p \in \mathcal{P}, i \in \mathcal{K}_p \tag{30}$$

\

Sort $p \in \mathcal{P}$ by increasing order of the day limits $d_{L,p}$
for number of initial schedules **do**
  for each protocol $h \in \mathcal{H}$ in randomized order **do**
    for patient $p \in \mathcal{P}_h$ in sorted order **do**
      Assign p to a random machine $m \in \mathcal{M}_p$
      Schedule p randomly on one of the two first available days in a random time window $w \in \mathcal{W}$ according to the partially occupied input schedule S
      Insert feasible schedule to $\mathcal{K}_p$

Algorithm 2 Schedule generation procedure

Master Problem Model (26)–(30) is the master problem: the restricted master problem is made of a subset $\mathcal{K}'_p \subset \mathcal{K}_p$ of feasible schedules for each $p \in \mathcal{P}$. A column in the master problem corresponds to a feasible schedule $i \in \mathcal{K}_p$ for patient $p \in \mathcal{P}$. The pure IP variables are now parameters: $x^i_{p,m,d,w}$, $t^i_{p,m,d,w}$, $q^i_{p,m,d,f}$, $y^i_{p,d,w}$, $z^i_{p,d}$, $u^i_{p,d}$, $v^i_{p,f}$, and $s^i_{p,f}$ for $p \in \mathcal{P}, m \in \mathcal{M}, d \in \mathcal{D}_w, f \in \mathcal{F}_p, w \in \mathcal{W}$ have fixed values that satisfy the scheduling constraints presented in Section 4.1. Each schedule has an associated (fixed) cost $c_{p,i}$ that is computed using (20)–(25) with weights $\alpha_1, \cdots, \alpha_6$ according to (31):

$$c_{p,i} = \alpha_1 f^i_{1,p} + \alpha_2 f^i_{2,p} + \alpha_3 f^i_{3,p} + \alpha_4 f^i_{4,p} + \alpha_5 f^i_{5,p} + \alpha_6 f^i_{6,p}. \tag{31}$$

The objective function (26) states that the aim is to minimize the total cost of the chosen schedules, plus an offset of 1 to make it equivalent with the other models. Constraint (27) states that exactly one schedule is chosen for each patient. Constraint (28) ensures that all chosen schedules will fit in the schedule. Constraint (29) states that the start day of a patient with shorter day limit should always be before or equal to the start day of a patient with longer day limit if they have the same treatment protocol, by multiplying the master variable with the start day of the corresponding schedule. Note the abuse of notation, where $p + 1$ denotes the next entry in $\mathcal{P}_h$.

Relaxing the integer assumption and solving the LP yields the dual variables λ_p associated with (27), $\gamma_{m,d,w}$ associated with (28) and $\eta_{h,p}$ associated with (29).

Subproblems One subproblem is formed for each patient $p \in \mathcal{P}$, with the aim to generate a new feasible schedule to add to $i \in \mathcal{K}'_p$, i.e., as a column to the restricted master problem. The constraints are the same as the pure IP model (2)–(8), (11)–(19). The schedule availability constraint (9) is replaced by (32), since the subproblems only deal with one patient at a time.

$$\begin{aligned}(x^i_{p,m,d,w} - t^i_{p,m,d,w})dur_p + \\ t^i_{p,m,d,w} dur_{p0} + S_{m,d,w} \le L_w,\end{aligned} \qquad \forall m \in \mathcal{M}, d \in \mathcal{D}_w, w \in \mathcal{W} \tag{32}$$

The subproblem objective function (33) is the cost of the schedule defined by (31), minus the master dual variables λ_p, γ_{mdw} and $\eta_{h,p}$ multiplied by the coefficients

given from their respective constraints in the master problem. The dual variable $\eta_{h,p}$ associated with (29), with h being the protocol of patient p, is partly shared between patient p and $p+1$ for $p \in \mathcal{P}_h$, because of the formulation of (29) (using the same abuse of notation).

$$\begin{aligned} \text{minimize} \quad & c_{p,i} - \lambda_p - \sum_{m\in\mathcal{M}} \sum_{d\in\mathcal{D}_w} \sum_{w\in\mathcal{W}} \gamma_{m,d,w} \left((x^i_{p,m,d,w} - t^i_{p,m,d,w}) dur_p + t^i_{p,m,d,w} dur_{p0} \right) \\ & - (\eta_{h,p-1} - \eta_{h,p}) \sum_{m\in\mathcal{M}} \sum_{d\in\mathcal{D}_w} \sum_{w\in\mathcal{W}} t^i_{p,m,d,w} d \end{aligned} \tag{33}$$

Since the subproblems are isolated from each other, constraint (32) can be satisfied as long as the input schedule $S_{m,d,w}$ together with the current patient's duration do not require more than the entire window capacity L_w. From this point of view, the subproblems are very easy to solve. On the other hand, the optimization is exclusively guided by the values of the dual variables which might lead to a larger number of iterations.

4.3 Constraint Programming Model

In [32], the authors found that the CP model that used bin packing constraints was more efficient than the CP model that used scheduling constraints for the RT patient scheduling problem. Therefore, we present a CP model that uses a global bin packing constraint. The variables in the CP model are presented in Table 5. The aim is to assign a $start\_day_p$, a $machine_{p,d}$, and a time $window_{p,d}$ to each patient $p \in \mathcal{P}$ for each day $d \in \mathcal{D}_w$ using the formulation (34)-(44). The variable $machine\_group_p$ is a function of $machine_{p,start\_day_p}$.

Constraints Constraint (35) states that each patient must start on an allowed start day. Constraint (36) states that the treatment must be scheduled on a machine allowed for that patient. Constraint (37) and (38) make all fractions scheduled on machines from a group of beam-matched machines. Constraint (39) limits the earliest start day and that the start day for each patient should be at least F_p days from the end of the planning horizon. Equation (40) states that the variable $window_{p,d}$ should be nonzero if and only if treatment has started but not ended. The active machine days should be the same as the active window days, which is stated in (41). The number of active machine days should be the same as the number of fractions and is stated in (42). This is a redundant constraint, as it is already enforced by (40) and (41), but added as it helps performance during search.

$$\text{minimize} \quad 1 + \sum_{p\in\mathcal{P}} (\alpha_1 k_{1,p} + \alpha_2 k_{2,p} + \alpha_3 k_{3,p} + \alpha_4 k_{4,p} + \alpha_5 k_{5,p} + \alpha_6 k_{6,p}) \tag{34}$$

$$\text{subject to} \quad start\_day_p \in \mathcal{A}_p \quad \forall p \in \mathcal{P} \tag{35}$$

Table 5 Variables in the CP model

$window_{p,d} \in \{0, \ldots, W\}$	Window patient $p \in \mathcal{P}$ is scheduled in on day $d \in \mathcal{D}_w$, where 0 represents patient p not being scheduled day d
$machine_{p,d} \in \{0, \ldots, M\}$	Machine patient $p \in \mathcal{P}$ is scheduled on day $d \in \mathcal{D}_w$, where 0 represents patient p not being scheduled day d
$start\_day_p \in \mathcal{D}_w$	Start day for patient $p \in \mathcal{P}$
$machine\_group_p \in \mathcal{B}_\mathcal{M}$	Group of beam-matched machines patient $p \in \mathcal{P}$ is scheduled on

$$machine_{p,d} \in \mathcal{M}_p \quad \forall p \in \mathcal{P}, d \in \mathcal{D}_w \tag{36}$$

$$machine\_group_p = b_\mathcal{M} \text{ if } machine_{p,start\_day_p} \in b_\mathcal{M} \quad \forall p \in \mathcal{P}, b_\mathcal{M} \in \mathcal{B}_\mathcal{M} \tag{37}$$

$$machine_{p,d} \in machine\_group_p \quad \forall p \in \mathcal{P}, d \in \mathcal{D}_w \tag{38}$$

$$d_{\min,p} \leq start\_day_p \leq D_w - F_p + 1 \quad \forall p \in \mathcal{P} \tag{39}$$

$$window_{p,d} > 0 \iff d \geq start\_day_p \wedge d < start\_day_p + F_p \qquad \forall p \in \mathcal{P}, d \in \mathcal{D}_w \tag{40}$$

$$machine_{p,d} > 0 \iff window_{p,d} > 0 \quad \forall p \in \mathcal{P}, d \in \mathcal{D}_w \tag{41}$$

$$\sum_{d \in \mathcal{D}_w} machine_{p,d} = F_p \quad \forall p \in \mathcal{P} \tag{42}$$

$$\begin{aligned} &\texttt{bin\_packing}([\infty, L_1, \ldots, L_W], \\ &\quad [\text{for each } p \in \mathcal{P}: \\ &\quad \text{if } m = machine_{p,d} \wedge d = start\_day_p \text{ then} \\ &\quad\quad (window_{p,d}, dur_{p0}) \\ &\quad \text{else if } m = machine_{p,d} \text{ then} \\ &\quad\quad (window_{p,d}, dur_p) \\ &\quad \text{else } (0,0)] + + \\ &\quad [\text{for each } w \in \mathcal{W}: (w, S_{m,d,w})]) \end{aligned} \qquad \forall m \in \mathcal{M}, d \in \mathcal{D}_w \tag{43}$$

$$start\_day_p \leq start\_day_{p+1} \qquad \forall h \in \mathcal{H}, p \in \mathcal{P}_h \text{ where } d_{L,p} \leq d_{L,p+1} \tag{44}$$

To ensure that the patients fit in each window, a global bin packing constraint [44] is used. In (43), the first line states that the capacity of window 0 is infinite (corresponding to not being scheduled). Window $1, \ldots, W$ have capacity $L_1, \ldots, L_W$. In (43), the bin choice and required allocation are created together as a list of pairs

(*bin*, *size*) for each patient. The first value in the pair corresponds to the bin, here the $window_{p,d}$, and the second is the size of the item, which corresponds to the duration of the treatment. If the patient is not scheduled on the particular machine, window 0 is chosen with item size 0. This list is concatenated (using the ++ operator) with a list of pairs used to include already occupied timeslots $S_{m,d,w}$ in window w.

The same dominance breaking as in the IP model is included: if two patients have the same treatment protocol, the one with shorter treatment target should start treatment first, which is stated in Constraint (44), using the same abuse of notation as before, i.e., $p+1$ denotes the next entry in $\mathcal{P}_h$.

Objective Function Equation (34) shows the generalized objective function, which is divided into six parts $k_1, \ldots, k_6$ according to Section 3, and combined with weights $\alpha_1, \ldots, \alpha_6$. An offset of 1 is included to make it equivalent to the IP objective function (1).

Objective (i) is to minimize a weighted sum of the waiting times, which is done in (45) by penalizing the number of days between the first allowed start day $d_{\min,p}$ and the start day, multiplied with weight c_p corresponding to the priority group of patient p.

$$k_{1,p} = c_p(start\_day_p - d_{\min,p}) \tag{45}$$

Objective (ii), to minimize a weighted sum of the violations of the waiting time targets, is formulated in (46). The target violation is zero if the start day is before the waiting time target, and otherwise penalized linearly.

$$k_{2,p} = c_p \max(0, start\_day_p - d_{L,p}) \tag{46}$$

Objective (iii) is to minimize the number of window switches. Therefore, a penalty of value one is added each time the window is switched. Since only the days when treatment has started but not finished are relevant, i.e., when $window_{p,d} \neq 0$, we let the active treatment days form a set $\mathcal{D}_a \subset \mathcal{D}_w$ and compute the number of window switches on that set in (47). Note the abuse of notation, where $d+1$ denotes the next entry in $\mathcal{D}_a$.

$$k_{3,p} = \sum_{d \in \mathcal{D}_a} (window_{p,d} \neq window_{p,d+1}) \tag{47}$$

Objective (iv) is to minimize the violations of the window preferences, which is formulated in (48) for $p \in \mathcal{P}^{pref}$ (otherwise $k_{4,p} = 0$).

$$k_{4,p} = \sum_{d \in \mathcal{D}_a} |window_{p,d} - w_p^{pref}| \tag{48}$$

Objective (v), to minimize the fractions scheduled on a non-preferred machine, is formulated in (49).

$$k_{5,p} = \sum_{d \in \mathcal{D}_a} (machine_{p,d} \notin \mathcal{M}_p^{pref}) \tag{49}$$

The Luby restart strategy [48] with parameter 75 is used
Assign the sum $\sum_{p \in \mathcal{P}}$ of all window switches $k_{3,p}$ (47) to the minimum value
Assign the sum $\sum_{p \in \mathcal{P}}$ of fractions on non-preferred machine $k_{5,p}$ (49) to the minimum value
Assign the sum $\sum_{p \in \mathcal{P}}$ of machine switches to partially beam-matched machines $k_{6,p}$ (50) to the minimum value
Create patient list sorted by priority group, with all priority A patients first, followed by priority B and C
for p in sorted list of patients **do**
    Assign $start\_day_p$ to the minimum value
    Assign $machine_{p,d}$ a random value for each $d \in \mathcal{D}_w$
    Assign the window preference violation $k_{4,p}$ in (48) to the minimum value
    Assign $window_{p,d}$ a random value for each $d \in \mathcal{D}_w$

Algorithm 3 CP Search Heuristic

Objective (vi), to minimize the number of switches to a machine that is only partially beam-matched, is formulated in (50). It states that if $machine_{p,d+1}$ is in the set of partially beam-matched machines $p_{\mathcal{M}}$ for the next day after $machine_{p,d}$, then there has been a switch between day d and $d+1$.

$$k_{6,p} = \sum_{d \in \mathcal{D}_a} (machine_{p,d+1} \in p_{\mathcal{M}}^{machine_{p,d}}) \tag{50}$$

Search Heuristic In CP, the solvers rely on backtracking algorithms that are used in the tree search-based heuristics. When using backtracking search, a sequence of decisions are made regarding what *variable* to branch on next, and which *value* to assign to the variable. It is well known that the choice of variable and value ordering, also called search heuristic, can be crucial to solving a problem efficiently, see e.g. [45, 46].

For the CP model in this paper, many different choices of variable and value orderings were investigated. Our initial experiments showed that randomization and restarts are necessary to obtain good results: the restart search helps avoid getting stuck in a non-productive area of the search tree [47]. Several restart strategies were evaluated, and the strategy with best performance was the Luby restart strategy [48], which gives a specific scheme for when search is restarted.

When deciding what variables to assign random values in the search heuristic, the best performing search heuristic was shown to be tightly related to the objective function (34). It is obvious that assigning the $start\_day_p$ variable a random value will not result in good quality solution, whereas assigning $machine_{p,d}$ a random value will give the benefit of a wider search tree. The search heuristic is described in Algorithm 3. Large Neighborhood Search [49] was also tested but did not improve overall performance.

1: Sort patients by their priority
2: Sort patients of same priority by the day of arrival
3: **for** p in sorted list of patients **do**
4: Randomly assign p to a machine in $\mathcal{M}_p$
5: Schedule p randomly on one of the two first available days in the first available time window $w \in \mathcal{W}$ according to the partially occupied input schedule S
6: D_p = the last day in the schedule
7: $D_w = \max_p(D_p) + 30$ ▷ Augment search space by 30

Algorithm 4 Time Horizon Heuristic

4.4 Combined CP/IP Method

For difficult MIP problems, providing the solver a good quality input solution ("warm start") can improve performance. The solver processes the input solution before starting branch-and-cut to get a lower/upper bound to use during the optimization, which allows it to eliminate parts of the search space. For this reason, the CP model is used to find a feasible solution to use as a warm start in the IP model. The first feasible solution found by the CP model is generally of good quality because of the search heuristic, which gives the MIP solver a useful upper bound during branch-and-cut. The CP solution is transformed to the format of the IP x-variables by letting $x_{p,m,d,w} = 1$ if and only if $w = window_{p,d}$ and $m = machine_{p,d}$ for $p \in \mathcal{P}, m \in \mathcal{M}, d \in \mathcal{D}_w, w \in \mathcal{W}$. This is provided to the MIP-solver using the built-in functionality for advance starting.

5 Experimental Setup

This section presents the setup for the experiments. Section 5.1 presents how the time horizon is computed. Section 5.2 describes the historical patient data from Iridium Netwerk and how the problem instances are generated. Section 5.3 presents the objective functions.

The experiments are run on a Windows 10 machine with an Intel® Core™ i9-7940X X-series processor and 64 GB of RAM. The patient arrival model used when generating benchmarks is built with Python 3.8. The IP models are solved using the MIP solver of CPLEX 12.10 in the Python API with the default parameters. The CP model is written in MiniZinc 2.5.5 [50], and uses the Gecode 6.3.0 solver [51]. Other CP solvers were tested, such as the lazy clause generation solver Chuffed [52], but Gecode gave the best overall results on the tested problem instances. The maximum allowed CPU time was set to 1 h per run.

5.1 Computing the Time Horizon D_w

The models all depend on the number of days in the time horizon. D_w should be large enough to schedule all treatments, but a larger D_w may weaken performance due to larger problem dimensions. A heuristic to compute D_w is presented in Algorithm 4.

A random schedule is computed, and the value of D_w is set to the last utilized day out of all patients, plus 30 days that are added to augment the search space.

5.2 Generating Problem Instances using Historical Clinic Data

In 2020, 4070 patients received 5500 treatments at Iridium Netwerk, with each treatment following a treatment protocol. From the historical data, the empirical distribution of the 72 different treatment protocols can be computed. Each protocol states the machines that are equipped for treating the particular tumor, what machines that are preferred for treating the target, and the duration for the first and subsequent fractions. The average number of fractions for each treatment protocol is computed from historical data. Furthermore, each treatment protocol is given a priority (A, B or C) by a radiation oncologist (MD) at Iridium, which will give an equivalent patient priority. In 2020, Iridium Netwerk operated 10 linacs and 255 days were used to treat the 4070 patients, resulting in an average arrival rate of 16 patients per working day. No records were kept over the patients' time window preferences, but the booking administrators estimate that 80% of the patients have a preference, of which 65% prefer a treatment before noon and 35% prefer the afternoon, and that 20% of the patients have no preference.

Literature shows the majority of patients find it reasonable to receive a notification of the treatment three days in advance [43]. Therefore, in this paper the duration of notice is three days for priority B and C patients, while priority A patients are notified immediately. All fractions are communicated and cannot be re-planned, as this is the current practice at Iridium Netwerk. The schedule can change until being communicated; booking decisions are postponed to the next day for patients scheduled after the notification period. The notification period length is straightforward to change.

Problem Generation A model for patient arrivals is developed. The goal is to mimic scheduling behavior to generate realistic problem instances based on the historical data from Iridium Netwerk. Each problem instance should represent different scenarios, altering the number of patients to be scheduled and the partially occupied input schedule. The problem generation algorithm simulates each day at the clinic; the patients arriving, and the resulting schedules. An overview of the steps during each simulated day can be seen in Fig. 2.

In the first step, new patients are assumed to arrive according to a Poisson process based on historical arrival rates. Each patient is randomly assigned a treatment protocol from the empirical distribution of protocols. Secondly, priority A and B patients that are expected to arrive in the coming four weeks are added to the problem as placeholder (*dummy*) patients. Their treatment target dates and earliest start day are set from when they are expected to arrive. Thirdly, Algorithm 4 is run to determine the time horizon. Next, the IP model is run to generate a schedule (but any model could be used in the problem generation phase). The schedule assigns each patient a machine, treatment days, and time windows. Next, the results are post-processed. Patients that start treatment within the duration of notice are fixed to the schedule, whereas the booking decisions are postponed to the next day for patients that are scheduled more than three days away. Finally, the schedule is saved and is given together with the list of unscheduled patients as input to the next day.

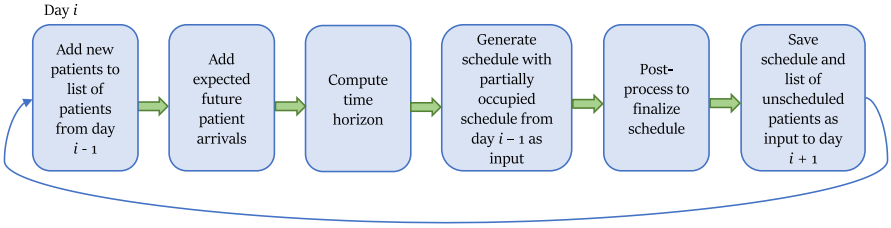

Fig. 2 Problem generation algorithm. Each day i, the patient arrivals are simulated and a schedule is computed. The resulting schedule and list of unscheduled patients are saved as input to day $i + 1$

Problem Benchmarks In order to evaluate how well the models scale to different problem sizes, the problem instance generator is used for four setups: two average arrival rates, $\lambda = \{16, 18\}$, and two different number of time windows, $W = \{2, 4\}$. When $W = 2$, the time window preferences from Iridium Netwerk are used. For $W = 4$, an estimation is that 25% prefer the first window, 25% prefer the last window, and 50% have no preference. For each setup, 20 days are randomly chosen between day 50 and 300 in the simulation to form the problem benchmarks. These instances represent different scenarios, altering the patient flow and the partially occupied input schedule.

In the generated problem benchmarks, the number of patients to schedule, including expected future arrivals as placeholder patients, varies. When $\lambda = 16$, $P \in [225, 239]$ with an average of 230.2, and when $\lambda = 18$, $P \in [250, 268]$ with an average of 256.5. The time horizon D_w also varies; $D_w \in [79, 89]$ when $\lambda = 16$, and $D_w \in [78, 94]$ when $\lambda = 18$. The average occupancy on all machines except M10 (which is specialized and always has a lower occupancy) of the first day is 65.7% for $\lambda = 16$ and 73.5% when $\lambda = 18$. All problem instances used in this paper are publicly available.[1]

5.3 Objective Functions

As presented in Section 3, there are several objectives considered: (i) is to minimize a weighted sum of the waiting times, (ii) is to minimize a weighted sum of the violations of the target dates, (iii) is to minimize the number of time window switches, (iv) is to minimize violations of time window preferences, (v) is to minimize the number of fractions scheduled on non-preferred machines, and (vi) is to minimize the number of times a patient switches between machines that are only partially beam-matched. These objectives are combined into different objective functions using weights $\alpha_1, \ldots, \alpha_6$ as presented in (1), (31) and (34). The different combinations that form the objective functions used in the experiments are presented in Table 6.

[1] Access through this link: https://osf.io/45qw2/?view_only=4a0a67e21cb542df8f9a0f74241de825

Table 6 The different objective function combinations

Objective function number	Combination	Weights
#1	(i) + (ii) + (iii) + (v) + (vi)	$\alpha_1 = 50, \alpha_2 = 100, \alpha_3 = 1, \alpha_4 = 0, \alpha_5 = 10, \alpha_6 = 10$
#2	(i) + (ii) + (iii) + (iv)	$\alpha_1 = 50, \alpha_2 = 100, \alpha_3 = 1, \alpha_4 = 1, \alpha_5 = 0, \alpha_6 = 0$
#3	(i) + (iii) + (v)	$\alpha_1 = 100, \alpha_2 = 0, \alpha_3 = 1, \alpha_4 = 0, \alpha_5 = 10, \alpha_6 = 0$
#4	(i) + (iii) + (iv) + (v) + (vi)	$\alpha_1 = 100, \alpha_2 = 0, \alpha_3 = 1, \alpha_4 = 5, \alpha_5 = 10, \alpha_6 = 10$

The objective functions are designed to mimic the scheduling policies at different clinics. For example, in some countries there are no official treatment target dates, and therefore objective (ii) is not active in objective functions #3 and #4. Some clinics do not consider the patients' preferences of treatment time of the day, which is why objective (iv) is not included in #1 and #3. For clinics that do not have multiple hospitals, it is unlikely that the problem with machine switches to partially beam-matched machines exists, hence objective (vi) is irrelevant and therefore not included in #2 and #3. Some clinics may not state preferred machines, thus (v) is not included in objective function #2.

The waiting time has a large negative effect on the patient outcome, especially for acute patients, see e.g. [2]. Therefore, both objective (i) and (ii) also have the weight c_p for each patient (see (20) and (21)), which reflects the severeness of delaying treatment start for the different priority groups. In objective functions #1 and #2, the weights of α_1 and α_2 show that if the patients are of the same priority group, it is never desirable to minimize a patient's waiting time at cost of another patient missing their treatment target date. However, if one patient is priority A and one is priority C, the latter is allowed to violate the treatment target date if it means the priority A patient gets a shorter waiting time. Furthermore, the weights of α_5 and α_6 compared to α_2 in objective function #1 indicate that it is preferred for a patient to start their treatment earlier at the cost of either switching linacs or scheduling on non-preferred linacs. Finally, the weight of α_3 is lower than the rest; if possible, all fractions should be scheduled in the same time window, but never at the cost of any of the other objectives.

Objective function number #4 is most similar to what is used at Iridium Netwerk today. There are no official treatment target dates in Belgium, therefore objective (ii) is not active. Minimizing waiting times is by far the most important objective, thus the weight of this objective, $c_p\alpha_1$, is the largest. It is more important to fulfill the patient preferences regarding time windows than to schedule them in the same time window every day, thus $\alpha_3 < \alpha_4$. At Iridium, the current practice is to never schedule patients with switches between partially beam-matched machines. However, the staff at Iridium agrees that this should be allowed if it will lead to a minimized waiting time. Therefore, the penalty for scheduling patients on a non-preferred machine is the same as for scheduling patients with machine switches between partially beam-matched machines. If a treatment starts on a non-preferred machine, the aim is to switch to a preferred machine as soon as possible. This is true also in #4: the

cost of switching to a partially beam-matched preferred machine will be lower when there is more than one fraction left to schedule, since the switch is a one-time cost.

6 Results

The four different setups ($\lambda = \{16, 18\}$, $W = \{2, 4\}$) are run with the four objective functions presented in Table 6, giving a total of 16 different combinations that have 20 problem instances each.

6.1 Computational Efficiency

The models in Section 4 are run for the 20 problem instances for each of the 16 combinations. Table 7 presents the mean, median and cumulative CPU times. Table 8 presents the quality of the solutions; it shows the mean and median of the relative optimality gap, i.e., the mean or median of $(x - y)/y$, where x is the current best objective value and y is the proven optimal value. The table also presents the proportion of the problem instances without a feasible solution at timeout over the 20 runs, with the time limit set to 1 h.

Table 7 shows that the IP model often times out without having found a feasible incumbent solution. This happens in all setups except the easiest, when $\lambda = 16$, $W = 2$. When $\lambda = 18$, $W = 4$, this occurs almost all the time for the IP model for objective #1 and #4, and also in a non-negligible proportion of instances for the CP model and the combined CP/IP approach.

The results in Table 8 show that the CP model has the worst performance in almost all setups with regard to relative optimality gap after 1 h. This can be expected: both [32] and [33] showed that CP is good at finding feasible solutions for the RT scheduling problem, but not as efficient at finding an optimal solution. For all objectives and setups, the CP model frequently times out without proving optimality within the time frame. However, columns **A-B** in Table 8 show that the relative optimality gap is often small.

For $\lambda = 16$, the IP model often outperforms the Combined CP/IP model, suggesting the IP solver does not benefit from being warm started with a feasible CP solution in these cases. For more complicated instances ($\lambda = 18$), the results are the opposite; the Combined CP/IP methodology gives shorter average CPU time and fewer instances that time out without having found a feasible solution than the pure IP model.

When altering the number of time windows, Table 8 shows that $W = 2$ gives smaller relative optimality gaps and fewer instances without feasible solutions before timeout than for $W = 4$. Moreover, Table 7 shows the average solution times are also much shorter when $W = 2$ than when $W = 4$, likely due to the smaller problem dimensions. This difference in performance is largest for the IP model and smallest for the CG-IP model.

Table 7 Computational time results

Arrival rate	Number of time windows	Objective function number	**A**: Average CPU time, **B**: Median CPU time, **C**: Cumulative CPU time (all in minutes)											
			IP			CP			CG-IP			Combined CP/IP		
			A	**B**	**C**	**A**	**B**	**C**	**A**	**B**	**C**	**A**	**B**	**C**
$\lambda = 16$	$W = 2$	#1	7	7	155	21	11	435	**2**	**2**	**51**	12	7	247
		#2	**2**	**2**	57	51	60	1039	**2**	**2**	**53**	12	5	249
		#3	**2**	3	56	30	16	600	**2**	**2**	**47**	14	4	285
		#4	5	5	113	52	60	1046	**3**	**3**	**63**	14	12	289
	$W = 4$	#1	42	40	856	24	15	482	**3**	**3**	**70**	33	38	672
		#2	**6**	**6**	**136**	60	60	1200	8	**6**	163	28	36	560
		#3	5	5	111	42	40	843	**3**	**3**	**78**	8	7	173
		#4	45	53	900	60	60	1200	**9**	**8**	**182**	47	50	955
$\lambda = 18$	$W = 2$	#1	17	13	359	36	34	724	**4**	**3**	**81**	33	41	661
		#2	6	3	121	59	60	1192	**3**	**2**	**78**	18	6	375
		#3	7	**3**	143	45	46	910	**6**	**3**	**128**	20	17	405
		#4	14	8	287	57	60	1149	**4**	**3**	**91**	27	35	547
	$W = 4$	#1	60	60	1200	40	41	816	**8**	**7**	**165**	49	60	985
		#2	**28**	**25**	**570**	60	60	1200	48	60	963	40	43	818
		#3	**9**	**6**	**192**	54	60	1093	10	**6**	204	10	8	207
		#4	60	60	1200	60	60	1200	**55**	60	**1114**	60	60	1200

For each combination of arrival rate, number of time windows, objective function and model, column **A** shows the average CPU time, column **B** presents the median CPU time, and column **C** shows the cumulative CPU time for the 20 instances. The timeout is set to 1 h of CPU time. Bold text indicates best column

Overall, the CG-IP model has the smallest variance in solution times. Thus, the CG-IP model seems more robust to model size changes than the other models. This can also be seen when increasing the arrival rate from $\lambda = 16$ to $\lambda = 18$, as this increase has the smallest effect on the CG-IP model. Furthermore, the solution times are shortest for the CG-IP model, and the quality of solution is the best for this model.

The heuristic used to compute the time horizon, Algorithm 4, also generates a feasible schedule. Table 9 presents the results as means over all the problem instances, including the solution from the time horizon heuristic. The heuristic is much faster than any of the other models, but the solution quality is very poor, especially for the larger problem instances where it has a mean relative optimality gap of more than 100%.

Table 8 Solution quality results

Arrival rate	Number of time windows	Objective function number	**A**: Mean relative optimality gap at timeout (%), **B**: Median relative optimality gap at timeout (%), **C**: Proportion with no feasible solution at timeout (%)											
			IP			CP			CG-IP			Combined CP/IP		
			A	**B**	**C**	**A**	**B**	**C**	**A**	**B**	**C**	**A**	**B**	**C**
$\lambda = 16$	$W = 2$	#1	**0.0**	**0.0**	**0**	0.1	**0.0**	**0**	**0.0**	**0.0**	**0**	**0.0**	**0.0**	**0**
		#2	**0.0**	**0.0**	**0**	0.9	**0.0**	**0**	**0.0**	**0.0**	**0**	**0.0**	**0.0**	**0**
		#3	**0.0**	**0.0**	**0**	0.1	**0.0**	**0**	**0.0**	**0.0**	**0**	**0.0**	**0.0**	**0**
		#4	**0.0**	**0.0**	**0**	1.2	0.2	**0**	**0.0**	**0.0**	**0**	**0.0**	**0.0**	**0**
	$W = 4$	#1	0.2	**0.0**	**0**	0.7	**0.0**	**0**	**0.0**	**0.0**	**0**	**0.0**	**0.0**	**0**
		#2	**0.0**	**0.0**	**0**	8.0	7.5	5	**0.0**	**0.0**	**0**	**0.0**	**0.0**	**0**
		#3	**0.0**	**0.0**	**0**	0.7	**0.0**	**0**	**0.0**	**0.0**	**0**	**0.0**	**0.0**	**0**
		#4	0.3	**0.0**	30	16.5	14.6	**0**	**0.0**	**0.0**	**0**	0.3	**0.0**	5
$\lambda = 18$	$W = 2$	#1	**0.0**	**0.0**	10	0.5	**0.0**	**0**	**0.0**	**0.0**	**0**	0.5	**0.0**	5
		#2	**0.0**	**0.0**	**0**	2.0	0.9	30	**0.0**	**0.0**	**0**	**0.0**	**0.0**	**0**
		#3	**0.0**	**0.0**	**0**	0.8	**0.0**	20	**0.0**	**0.0**	**0**	0.1	**0.0**	**0**
		#4	**0.0**	**0.0**	10	6.1	6.4	**0**	**0.0**	**0.0**	**0**	**0.0**	**0.0**	5
	$W = 4$	#1	46.4	46.4	95	4.1	**0.0**	**0**	**0.0**	**0.0**	**0**	6.8	3.9	**0**
		#2	**0.0**	**0.0**	**0**	17.4	15.0	25	0.4	0.2	**0**	**0.0**	**0.0**	**0**
		#3	**0.0**	**0.0**	**0**	5.5	2.8	**0**	**0.0**	**0.0**	**0**	**0.0**	**0.0**	**0**
		#4	-	-	100	40.0	44.3	30	**0.2**	**0.0**	**0**	44.2	46.7	40

For each combination of arrival rate, number of time windows, objective function and model, column **A** shows the mean relative optimality gap, column **B** presents the median relative optimality gap, and column **C** shows the proportion of instances that did not have a feasible incumbent solution at timeout for the 20 instances. The timeout is set to 1 h of CPU time. Bold text indicates best column

6.2 Objective Function Evaluation

Evaluating the different objective functions, the results in Tables 7-8 suggest that for the IP model, objective functions #1 and #4 are much harder to solve than objective functions #2 and #3. This can be seen both for the mean relative optimality gap, the proportion of instances with no feasible solution and the CPU times. In objective function #1 and #4, objective (vi) is active, i.e., minimization of the number of switches to partially beam-matched machines. This seems to make the IP model much more complex, and the time for solving the root node relaxation alone increases from $200 - 300$ seconds for objective function #2, to $700 - 800$ seconds for objective function #4, although the problem dimensions are approximately the same.

For the CP model, it is instead objective functions #2 and #4 that are more difficult than objective functions #1 and #3. In objective functions #2 and #4, objective (iv) is active, i.e., minimization of time window preferences. This objective makes it more difficult for the CP search heuristic to find the optimal solution, or even a feasible solution within the time limit.

The CG-IP model is the only one that never times out without having found a feasible solution. This is expected, since the initial schedules generated by the heuristic in Algorithm 2 are all feasible. The time results in Table 7 show that the CG-IP model is also less sensitive to objective function changes than the other models. The results indicate that objective function #1 may be less complicated since its solution times are shorter, but the difference to the other objective functions is smaller than the differences between objective functions for the other models.

To evaluate the weights of the objective functions presented in Table 6, Fig. 3 shows different costs from the CG-IP solutions for the different objective functions and different arrival rates. Each subplot **i**. to **iv**. represents the key indicator of each of the objectives (i) to (vi). From plot **i**. to the upper left, one can see that the mean waiting time does not change much between the different objective functions, which is the expected result since objective (i) is present in all objective functions. However, the waiting times increase as the arrival rate increases from $\lambda = 16$ to $\lambda = 18$ patients per day. The plot of the mean violations of treatment target times, **ii**., shows that this violation is always close to zero. This indicates that the addition of objective (ii) (to minimize the violation of treatment target dates) does not have a large, or any, effect when simultaneously minimizing the waiting times.

To minimize the number of window switches, objective (iii), is present in all objective functions. Subplot **iii**. in Fig. 3 shows the mean number of window switches, and this value is very low for all objective functions. Objective (iv), to minimize the violation of the time window preferences, is active in objective function #2 and #4. Subplot **iv**. shows that this is reflected in the results; this violation is much higher in objective function #1 and #3. To minimize the machine preference violations, objective (v), is present in #1, #3 and #4, which agrees with the results in subplot **v**. Finally, objective (vi), to minimize the number of switches to partially beam-matched machines, is present in objective function #1 and #4, and although subplot **vi**. shows that the mean value for the number of switches is low also for #2 and #3, it is lower for #1 and #4.

In total, this shows that the weights for the objective functions presented in Table 6 are well reflected in the resulting schedules computed by the CG-IP model. It also shows that when the capacity is more limited due to a higher arrival rate, all the objectives are more difficult for the model to achieve.

6.3 Sensitivity Analysis

The parameters $\alpha_1, \ldots, \alpha_4$ are included in the sensitivity analysis. Objectives (v) and (vi) (relating to machine preferences and machine switches) are not relevant for clinics with a homogeneous machine setup. Therefore, the sensitivity analysis is focused on objective function #2, for which $\alpha_5 = \alpha_6 = 0$.

In Table 10, the base case used in the previous experiments and setups S1–S6 are presented. Due to the medical consequences, it is always more important to minimize waiting times for treatment start (i) (thereby also minimizing the violations of the treatment time targets (ii)) than to achieve a better patient experience [53, 54], in this case by maximizing the time consistency in treatments (iii) and minimizing

Table 9 Results at timeout of 60 min for both $W = 2$ and $W = 4$, for all problem instances and all objective functions

	IP		CG-IP		CP		Combined CP/IP		Heuristic	
Problem size	Easy	Difficult	Easy	Difficult	Easy	Difficult	Easy	Difficult	Easy	Difficult
Mean relative optimality gap (%)	0.1	6.2	**0.0**	**0.1**	3.5	9.6	**0.0**	6.5	11.8	124.0
No feasible solution (%)	3.8	26.9	**0.0**	**0.0**	0.6	13.1	0.6	6.3	**0.0**	**0.0**
Mean computation time (min)	15	25	4	18	43	52	22	33	**<0.01**	**<0.01**

Problem size *Easy* corresponds to $\lambda = 16$, and *Difficult* corresponds to $\lambda = 18$. Bold text indicates best column

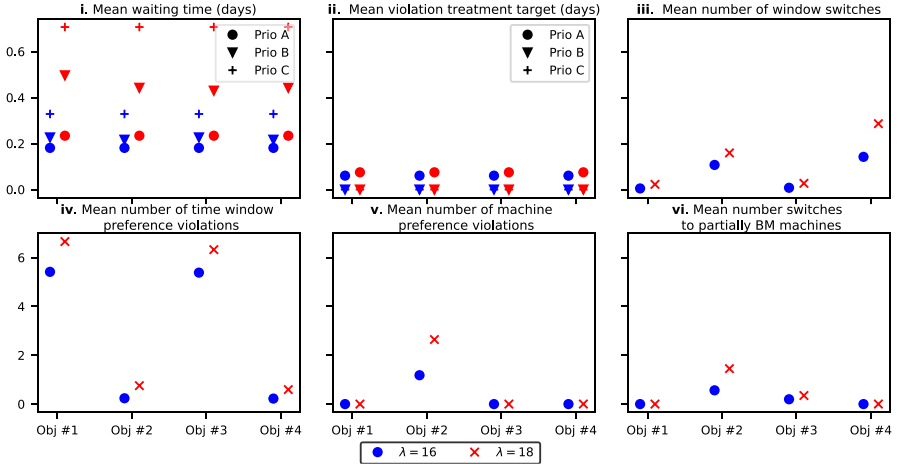

Fig. 3 Measurements from the CG-IP solutions relating to objective (**i**)–(**vi**) in Table 6 are shown for the different objective functions when $W = 4$

the violations of the patient wishes on treatment times (iv). This is reflected in the weights in the sensitivity analysis for all cases but one. In S6, it is instead prioritized to minimize the violation of the waiting time targets, and secondly to fulfill time consistency between appointments and to fulfill the patients time window preferences. Thirdly, the waiting times should be minimized.

The results for the different parameter settings using the CG-IP model are shown in Fig. 4. Since objectives (v) and (vi) are inactive in objective function #2, it can be expected that the results do not differ very much between the parameter setups, which is confirmed by the results. Furthermore, the waiting times and the violations of the treatment time targets are almost identical between the different parameter setups if excluding S6. The mutual order of α_1 and α_2 does not seem to matter as long as both are greater than α_3 and α_4: in S4, α_1 has a higher weight than α_2, which does not change the results in objectives (i) or (ii). The results for objective (iii) are similar for S1–S5, with differences only in the top 1% shown as outlier points, except for S6. In the results for objective (iv), S1 and S6 have a lower number of time window preference violations than the other parameter setups. This is likely caused by the objective weight α_4 being higher relative to α_3 than in the other setups. Since the other results are very similar for S1 in the other metrics, this shows that for a clinic with a different prioritization between the objectives, it is possible to adjust the weights to achieve the required order.

In S6, minimizing waiting time is no longer prioritized over minimizing time window switches and time window preference violations. This is probably not a relevant clinical scenario, but can give some insights in how the composite objective function works. The results in Fig. 4 show that both the number of time window switches (iii), and the number of time window preference violations (iv) are indeed lower for this setup, however, at the cost of some very long waiting times (i).

Table 10 Sensitivity analysis for objective function #2, where $\alpha_5 = 0, \alpha_6 = 0$

Case	Weights
Base case	$\alpha_1 = 50, \alpha_2 = 100, \alpha_3 = 1, \alpha_4 = 1$
S1: Sensitivity 1	$\alpha_1 = 10, \alpha_2 = 100, \alpha_3 = 1, \alpha_4 = 5$
S2: Sensitivity 2	$\alpha_1 = 10, \alpha_2 = 100, \alpha_3 = 5, \alpha_4 = 1$
S3: Sensitivity 3	$\alpha_1 = 50, \alpha_2 = 50, \alpha_3 = 1, \alpha_4 = 1$
S4: Sensitivity 4	$\alpha_1 = 100, \alpha_2 = 10, \alpha_3 = 1, \alpha_4 = 1$
S5: Sensitivity 5	$\alpha_1 = 5, \alpha_2 = 10, \alpha_3 = 1, \alpha_4 = 2$
S6: Sensitivity 6	$\alpha_1 = 1, \alpha_2 = 50, \alpha_3 = 5, \alpha_4 = 5$

The solution times for the different parameter setups are very similar to the base case times presented in Table 7. Overall, the sensitivity analysis shows that the composite objective function is not sensitive to the choice of the weights $\alpha_1 - \alpha_6$, but their relative size order matters for the results.

6.4 Conflicting Objectives

The results of case S6 in the sensitivity analysis in Fig. 4 indicate there could be a conflict between objectives (i) and (iv), i.e., to minimize waiting time and to minimize the violations of the patients' time window preferences. Using the IP model and the weighted sum method for multi-objective optimization [55, 56], this is analyzed for three randomly selected problem instances when $\lambda = 18$, $W = 4$. Figure 5 shows the pareto optimal points for the three instances when optimizing only the objectives (20) and (23) (both summed over $p \in \mathcal{P}$). It shows that there is indeed a conflict between the objectives (i) and (iv); if only minimizing the waiting times, there are more violations of the time window preferences, and if minimizing the

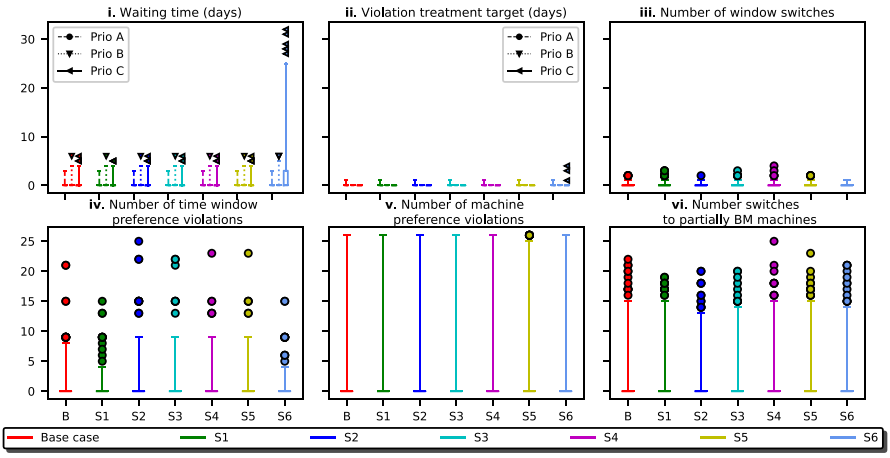

Fig. 4 Boxplots of the results in the different objectives in the sensitivity analysis using the CG-IP model. The weights of α for objective function #2 are varied according to Table 10. The top 1% are shown as outliers

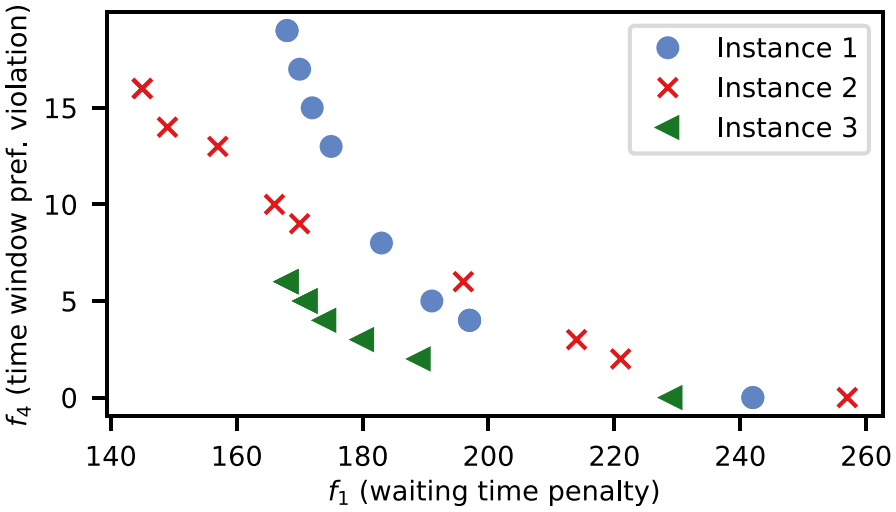

Fig. 5 Pareto optimal points for displaying the trade-off between (i) and (iv) for three instances where $\lambda = 18, W = 4$

violations of time window preferences to optimality, the waiting time penalty will be much higher. Since there are severe medical consequences for having a longer waiting time, the trade-off between these two objectives is easily managed; the weight α_1 for objective (i) should always be some magnitudes larger than α_4 for objective (iv)

6.5 Clinical Implications

The majority of the problem instances represent realistic scenarios since they are generated from clinical data. Objective function #4 is most similar to what is used at Iridium Netwerk today, and $\lambda = 16$ represent the average arrival rate at Iridium Netwerk. Therefore, this setup is used to analyze the clinical implications of using the CG-IP model for automatic scheduling.

Figure 6 shows the results for the 20 problem instances where $\lambda = 16$, $W = 4$, for objective #4, where a total of 327 patients have been scheduled to start treatment. Each of the six objectives described in Section 3 has its own boxplot, and for the waiting times and violations of waiting time targets, the results are further divided by the priority groups.

The results demonstrate that the waiting times are always below one week. The exact clinical waiting times are not available, but the staff at Iridium Netwerk certify that they are often 2–4 weeks for priority B and C patients. These preliminary results show that there could potentially be large clinical benefit of using the CG-IP model for automatic schedule generation.

7 Discussion

This section discusses the computational results, followed by a discussion of the potential for clinical implementation and directions for future work.

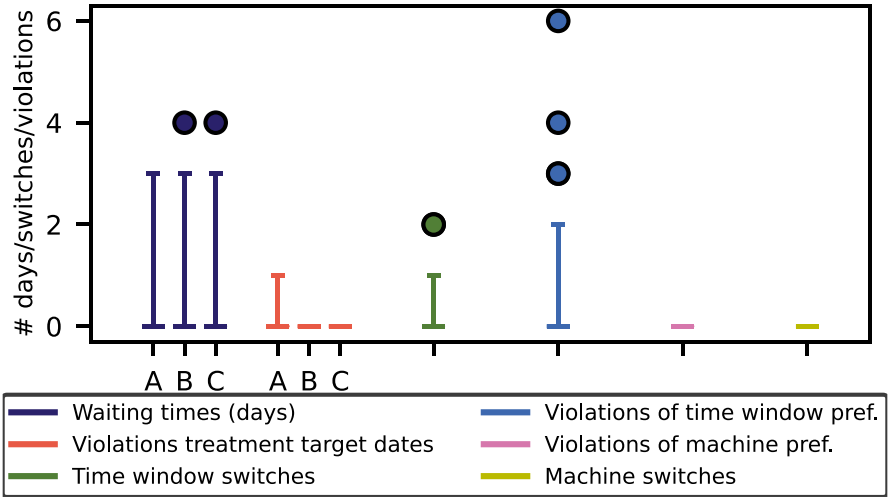

Fig. 6 Performance metrics for CG-IP for $\lambda = 16, W = 4$, with the top 1% shown as outlier points

7.1 Model Performance

The results show that the CG-IP model outperforms all other approaches in every aspect. It always finds feasible solutions and has the lowest mean optimality gap after one hour of run time. Table 7 shows these solutions are almost always found long before the time limit; when there is a nonzero mean optimality gap, it is most often because the optimal solution has not been generated by the column generation procedure. This can happen since the CG algorithm (Algorithm 1) does not guarantee the optimal solution to be found. Table 8 shows the mean deviation from the optimal value is always below 1%, which means the solutions are of very good quality. Furthermore, the solution times of the CG-IP algorithm can possibly be decreased by solving $200 - 300$ independent subproblems in parallel.

The CP model is the slowest of all models, and frequently times out without having found a feasible solution. For the smallest cases, when $\lambda = 16$ and $W = 2$, the quality of the solution is very good although it often reaches the time limit. For $\lambda = 16$ and $W = 4$, it performs well for objective #1 and #3. The CP model could therefore be considered suitable for a clinical implementation if the clinic's workload is not too high, and especially if the clinic does not try to fulfill the patients' time window preferences.

The IP model performs very well when $\lambda = 16$ and $W = 2$. Both when $\lambda = 16, W = 4$ and when $\lambda = 18, W = 2$ the IP model also performs well for objective function #2 and #3. If a clinic does not need to support partially beam-matched machines, the IP model could therefore be suitable for clinical implementation. However, it is not suited for Iridium Netwerk, or other clinics where specific machine switches is an objective to be minimized.

The results for the combined CP/IP methodology are better than the pure IP model, with fewer timeouts and better quality solutions. The disadvantage of developing and maintaining two separate models is however significant. Every time a constraint were to be altered or added, it would have to be done for both models,

which also means trying out different formulations to optimize performance, and possibly change the CP search heuristic. Although the combined CP/IP methodology may not be maintainable in practice, it shows the potential for warm starting the IP model for difficult cases. A similar advantage could possibly be gained by warm starting the IP from the feasible schedule computed when calculating the time horizon (see Section 5.1), which would require fewer computations and no extra model.

7.2 Potential for Clinical Implementation

The CG-IP model is considered most suitable for clinical implementation since it gives good quality solutions in a reasonable time, and is most robust to different setups. Robustness is crucial if the model should be generally applied to RT centers of different sizes.

The clinical staff does not necessarily know anything about mathematical programming. Thus, for a clinical implementation a user interface is needed, where the computations can be performed in the background. For the CG-IP to work in a generalized clinical setting, a number of constraints and objectives should be implemented in the model, with the possibility to activate them in the user interface by a simple click. Furthermore, different trade-offs between the objectives should be available depending on the clinic's needs, which will correspond to alterations in the α values.

The models have been developed to capture the medical and technical constraints at Iridium Netwerk, but the results should be generalizable. The majority of the constraints presented in Section 3 are applicable at other RT clinics as well, such as treatments on consecutive weekdays, specific allowed start days, specific machines suited for the treatments, different fraction durations, and machine capacity limits. When an objective is inactive in an objective function (by setting that particular $\alpha_i = 0$), the variables relating to that objective are free to take any values and do not contribute to the solutions. It is possible that a clinic has some other medical or technical constraint that is not implemented in the models, but related literature (e.g. [14, 16, 18, 25, 33]) show that the majority of the collaborating RT centers have similar constraints. The largest difference is likely the scheduling strategy; some clinics require the patients to be scheduled immediately at arrival, which would not work with the presented models as they all require multiple patients in a batch to be scheduled. The objective function evaluation and the sensitivity analysis also indicate that the models are generalizable: if a clinic has a different prioritization order among the objectives, it is straightforward to change the weights to acquire the preferred order. As long as the weights differ in some orders of magnitude, the models are not sensitive to their exact values.

7.3 Future Work

The models have so far been compared to each other. To further evaluate the CG-IP model, the automatically generated schedules should be compared to manually constructed schedules from Iridium Netwerk. The schedules obtained from the models also need to be assessed for their practical feasibility by the clinical staff.

There are various obstacles to implementing an automatic scheduling algorithm clinically. The first has been discussed above; a user interface is needed for a clinical implementation. Another is that the models do not support non-conventional treatments, such as multiple fractions per day, or treatments on non-consecutive days. At Iridium Netwerk, these are approximately 10% of all treatments, so for an automated scheduling algorithm to work in practice this extension would be necessary. Furthermore, sometimes a linac is unavailable because of maintenance or software upgrades or holidays, which is not taken into account in the models. When modeling machine unavailability, constraints on the minimum number of fractions per week are important for the treatment outcome. Finally, the patient protocols and the patient preferences regarding treatment time of the day would have to be registered instead of communicated verbally.

The method of using placeholder patients to account for expected future patients should be evaluated. Perhaps it is adequate to reserve time on the linacs each day for future patients instead, which would decrease the solution times. If not, a potential improvement of the stochastic aspect of the models is to use scenario-based probabilities instead of expected values. Another method would be to use a data-driven approach to predict future machine utilization.

Since there are multiple objectives to be optimized, multi-objective optimization could be considered. It is possible that a multi-objective approach could be relevant at some clinics, but not at Iridium Netwerk; one of the main objectives with automatic schedule creation is to minimize the manual work, and there is no desire from the clinical staff to be able to navigate trade-offs in several generated schedules. Furthermore, the computational time to generate the schedule cannot be too long for it to work in clinical practice, which makes multi-objective optimization unfit for the task.

A future extension of the automatic scheduling approach would be to be able to optimize schedules for the whole appointment series of each patient, including not only the linac scheduling, but also meetings with physicians and RTTs, the duration of the treatment planning, CT scans, and more. It would be interesting to study the trade-off between treatment plan quality and waiting time - when is it beneficial for the patient to have a longer waiting time in order to be treated on a more advanced machine, and when is a less advanced machine with shorter waiting time to prefer? Perhaps the toxicity effect of assigning patients to non-preferred machines could be incorporated in the scheduling models. To optimize the use of resources and the treatment plans simultaneously could potentially have a great impact on both the clinics and the individual patients.

8 Conclusions

As the incidences of cancer increase, the demand for RT grows. To better use resources in RT, algorithms can be used to automatically create patient schedules, a task that today is done manually in almost all clinics. The main contribution of this paper is to serve as a decision support tool when implementing a scheduling algorithm in practice. We present an extensive study of exact optimization technologies that can be used to model the RT scheduling problem. The output of the models is the assignment of all fractions of the patients to both linacs and specific time

windows, while including all the constraints necessary for the scheduling to work in practice. The models developed include an IP model, a CG-IP model, and a CP model, as well as a method combining the CP and IP models.

The models are tested using historical data from Iridium Netwerk in Antwerp, Belgium. Different cancer centers may have different intentions when creating the RT schedules, and in order to study the suitability of the different models for various cancer centers, each model is solved using multiple different objective functions. This is to evaluate if some particular optimization model is better suited to solve a certain objective.

The results demonstrate that the CG-IP model is the most robust, and that the mean optimality gap of the method is well below 1% for all the different setups and objective functions after one hour of computation time. The CP and IP models could have potential for clinical implementation depending on the size of the clinic, and more importantly, depending on their objective of scheduling.

The proposed methodology provides a tool for automated scheduling of RT treatments on linacs, and can be generally applied to RT centers. This would allow the RT staff to save time, and at the same time create optimized patient schedules that take medical and technical constraints into account. Designing more efficient schedules could potentially save lives by shortening waiting times and improving patient outcomes.

Acknowledgements The authors want to extend thanks to Kjell Eriksson, Chief Science Officer at RaySearch Laboratories, for valuable discussions, and Geert De Kerf, Medical Physics Expert at Iridium Netwerk, for help collecting the data used in the experiments and providing necessary clinical information.

Funding Open access funding provided by Royal Institute of Technology.

Data Availability The datasets generated during the current study are available in the Open Science Framework repository, and are accessible through *this link*.

Declarations

Conflicts of Interest On behalf of all authors, the corresponding author states there is no conflict of interest.

References

1. Borras JM, Lievens Y, Barton M, Corral J, Ferlay J, Bray F, Grau C (2016) How many new cancer patients in Europe will require radiotherapy by 2025? An ESTRO-HERO analysis. Radiother Oncol 119(1):5–11

2. Chen Z, King W, Pearcey R, Kerba M, Mackillop WJ (2008) The relationship between waiting time for radiotherapy and clinical outcomes: A systematic review of the literature. Radiother Oncol 87(1):3–16
3. Fortin A, Bairati I, Albert M, Moore L, Allard J, Couture C (2002) Effect of treatment delay on outcome of patients with early-stage head-and-neck carcinoma receiving radical radiotherapy. Int J Radiat Oncol Biol Phys 52(4):929–936
4. Gomez DR, Liao KP, Swisher SG, Blumenschein GR, Erasmus JJ, Buchholz TA, Giordano SH, Smith BD (2015) Time to treatment as a quality metric in lung cancer: Staging studies, time to treatment, and patient survival. Radiother Oncol 115(2):257–263
5. O'Rourke N, Edwards R (2000) Lung cancer treatment waiting times and tumour growth. Clin Oncol 12(3):141–144
6. Van Harten MC, Hoebers FJP, Kross KW, Van Werkhoven ED, Van Den Brekel MWM, Van Dijk BAC (2015) Determinants of treatment waiting times for head and neck cancer in the netherlands and their relation to survival. Oral Oncol 51(3):272–278
7. Lim KSH, Vinod SK, Bull C, O'Brien P, Kenny L (2005) Prioritization of radiotherapy in Australia and New Zealand. Australas Radiol 49(6):485–488 Radiation
8. Scoccianti S, Agresti B, Simontacchi G, Detti B, Cipressi S, Iannalfi A, Marrazzo L, Mangoni M, Paiar F, Livi L, Biti G (2012) From a Waiting List to a Priority List: A Computerized Model for an Easy-to-Manage and Automatically Updated Priority List in the Booking of Patients Waiting for Radiotherapy. Tumori J 98(6):728–735
9. Ebert MA, Li W, Jennings L, Kearvell R, Bydder S (2013) Utilitarian prioritization of radiation oncology patients based on maximization of population tumour control. Phys Med Biol 58(12):4013–4029
10. Thomsen MS, Nørrevang O (2009) A model for managing patient booking in a radiotherapy department with differentiated waiting times. Acta Oncol 48(2):251–258
11. DeVita Jr VT, Rosenberg SA, Lawrence TS (2018) DeVita, Hellman, and Rosenberg's Cancer: Principles & Practice of Oncology. Wolters Kluwer Health / Lippincott Williams and Wilkins, 11th edition
12. Vieira B, Hans EW, Van Vliet-Vroegindeweij C, Van De Kamer J, Van Harten W (2016) Operations research for resource planning and -use in radiotherapy: a literature review. BMC Med Inform Decis Mak 16(149)
13. Conforti D, Guerriero F, Guido R (2008) Optimization models for radiotherapy patient scheduling. 4OR, 6(3):263–278
14. Conforti D, Guerriero F, Guido R (2010) Non-block scheduling with priority for radiotherapy treatments. Eur J Oper Res 201(1):289–296
15. Jacquemin Y, Marcon E, Pommier P (2011) A pattern-based approach of radiotherapy scheduling. In IFAC Proceedings Volumes 44:6945–6950
16. Sauré A, Patrick J, Tyldesley S, Puterman ML (2012) Dynamic multi-appointment patient scheduling for radiation therapy. Eur J Oper Res 223(2):573–584
17. Gocgun Y (2018) Simulation-based approximate policy iteration for dynamic patient scheduling for radiation therapy. Health Care Manag Sci 21(3):317–325
18. Legrain A, Fortin MA, Lahrichi N, Rousseau LM (2015) Online stochastic optimization of radiotherapy patient scheduling. Health Care Manag Sci 18:110–123
19. Aringhieri R, Duma D, Squillace G (2020) Pattern-based online algorithms for a general patient-centred radiotherapy scheduling problem. In Health Care Systems Engineering, Springer Proceedings in Mathematics & Statistics, volume 316, pages 251–262. Springer
20. Li S, Koole G, Xie X (2020) An adaptive priority policy for radiotherapy scheduling. Flex Serv Manuf J 32:154–180
21. Vogl P, Braune R, Doerner KF (2019) Scheduling recurring radiotherapy appointments in an ion beam facility: Considering optional activities and time window constraints. J Sched 22(2):137–154
22. Maschler J, Raidl GR (2020) Particle therapy patient scheduling with limited starting time variations of daily treatments. Int Trans Oper Res 27(1):458–479
23. Braune R, Gutjahr WJ, Vogl P (2022) Stochastic radiotherapy appointment scheduling. CEJOR 30(4):1239–1277
24. Vieira B, Demirtas D, van de Kamer JB, Hans EW, Rousseau LM, Lahrichi N, van Harten W (2020) Radiotherapy treatment scheduling considering time window preferences. Health Care Manag Sci 23(4):520–534

25. Vieira B, Demirtas D, van de Kamer JB, Hans EW, Jongste W, van Harten W (2021) Radiotherapy treatment scheduling: Implementing operations research into clinical practice. PLoS ONE 16(2 February):1–13
26. Moradi S, Najafi M, Mesgari S, Zolfagharinia H (2022) The utilization of patients' information to improve the performance of radiotherapy centers: A data-driven approach. Comput Ind Eng 172(Part A):108547
27. Rossi F, van Beek P, Walsh T, editors (2006) Handbook of Constraint Programming, volume 2 of Foundations of Artificial Intelligence. Elsevier
28. Baatar D, Boland N, Brand S, Stuckey PJ (2011) CP and IP approaches to cancer radiotherapy delivery optimization. Constraints 16:173–194
29. Hahn-Goldberg S, Beck JC, Carter MW, Trudeau M, Sousa P, Beattie K (2014) Solving the chemotherapy outpatient scheduling problem with constraint programming. J Appl Oper Res 6(3):135–144
30. Doulabi SHH, Rousseau LM, Pesant G (2016) A constraint-programming-based branch-and-price-and-cut approach for operating room planning and scheduling. INFORMS J Comput 28(3):432–448
31. Barták R, Salido M, Rossi F (2010) New trends in constraint satisfaction, planning, and scheduling: A survey. Knowl Eng Rev 25:249–279
32. Frimodig S, Schulte C (2019) Models for radiation therapy patient scheduling. In: Schiex Thomas, de Givry Simon (eds) Principles and Practice of Constraint Programming. Lecture Notes in Computer Science. Springer Cham, pp 421–437
33. Pham TS, Rousseau LM, De Causmaecker P (2022) A two-phase approach for the Radiotherapy Scheduling Problem. Health Care Manag Sci 25(2):191–207
34. Wang Y, Zhang G, Zhang L, Tang J, Mu H (2018) A Column-Generation Based Approach for Integrating Surgeon and Surgery Scheduling. IEEE Access 6:41578–41589
35. Range TM, Lusby RM, Larsen J (2014) A column generation approach for solving the patient admission scheduling problem. Eur J Oper Res 235(1):252–264
36. Bard JF, Purnomo HW (2005) Preference scheduling for nurses using column generation. Eur J Oper Res 164(2):510–534
37. Shao K, Fan W, Yang Z, Yang S, Pardalos PM (2022) A column generation approach for patient scheduling with setup time and deteriorating treatment duration. Oper Res Int Journal 22(3):2555–2586
38. Shao K, Fan W, Lan S, Kong M, Yang S (2023) A column generation-based heuristic for brachytherapy patient scheduling with multiple treatment sessions considering radioactive source decay and time constraints. Omega 118:102853
39. Turner KJ, Bing Q (2002) Protocol techniques for testing radiotherapy accelerators. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 2529 LNCS. Springer-Verlag, Berlin Heidelberg, pp 81–96
40. Mitera G, Swaminath A, Wong S, Goh P, Robson S, Sinclair E, Danjoux C, Chow E (2009) Radiotherapy for oncologic emergencies on weekends: Examining reasons for treatment and patterns of practice at a Canadian cancer centre. Curr Oncol 16(4):55–60
41. Yeo R, Campbell T, Fairchild A (2012) Is weekend radiation therapy always justified? J Med Imaging and Radiat Sci 43(1):38–42
42. Babashov V, Sauré A, Ozturk O, Patrick J (2023) Setting wait time targets in a multi-priority patient setting. Prod Oper Manag 32(6):1958–1974
43. Olivotto IA, Soo J, Olson RA, Rowe L, French J, Jensen B, Pastuch A, Halperin R, Truong PT (2015) Patient preferences for timing and access to radiation therapy. Curr Oncol 22(4):279–286
44. Shaw P (2004) A constraint for Bin Packing. In: Wallace M (ed) Tenth International Conference on Principles and Practice of Constraint Programming, vol 3258. LNCS. Berlin Heidelberg, Springer, pp 648–662
45. Haralick RM, Elliott GL (1980) Increasing tree search efficiency for constraint satisfaction problems. Artif Intell 14(3):263–313
46. Ginsberg ML, Frank M, Halpin MP, Torrance MC (1990) Search lessons learned from crossword puzzles. AAAI Conference on Artificial Intelligence, pages 210–215
47. van Beek P (2006) Backtracking search algorithms. In: Rossi Francesca, van Beek Peter, Walsh Toby (eds) Handbook of Constraint Programming, volume 2 of Foundations of Artificial Intelligence. Elsevier, pp 85–134
48. Luby M, Sinclair A, Zuckerman D (1993) Optimal speedup of Las Vegas algorithms. Inf Process Lett 47:173–180

49. Shaw P (1998) Using constraint programming and local search methods to solve vehicle routing problems. In: Maher Michael, Puget Jean-Francois (eds) Fourth International Conference on Principles and Practice of Constraint Programming, vol 1520. LNCS. Heidelberg, Springer, Berlin Heidelberg, Berlin, pp 417–431
50. Nethercote N, Stuckey PJ, Becket R, Brand S, Duck GJ, Tack G (2007) MiniZinc: Towards a Standard CP Modelling Language. In: Bessière Christian (ed) Principles and Practice of Constraint Programming, vol 4741. Lecture Notes in Computer Science. Springer Cham, pp 529–543
51. Gecode Team (2006) Gecode: Generic Constraint Development Environment. Available from http://www.gecode.org
52. Ohrimenko O, Stuckey PJ, Codish M (2009) Propagation via lazy clause generation. Constraints 14(3):357–391
53. Van Lent WAM, De Beer RD, Van Triest B, Van Harten WH (2013) Selecting indicators for international benchmarking of radiotherapy centres. J Radiother Pract 12(1):26–38
54. Harden SV, Chiew KL, Millar J, Vinod SK (2022) Quality indicators for radiation oncology. J Med Imaging Radiat Oncol 66(2):249–257
55. Zadeh LA (1963) Optimality and Non-Scalar-Valued Performance Criteria. IEEE Trans Autom Control 8(1):59–60
56. Marler RT, Arora JS (2010) The weighted sum method for multi-objective optimization: New insights. Struct Multidiscip Optim 41(6):853–862