



Scheduling recurring radiotherapy appointments in an ion beam facility

Considering optional activities and time window constraints

Petra Vogl¹ · Roland Braune¹ · Karl F. Doerner¹

© The Author(s) 2018

Abstract

Ion beam radiotherapy is a modern form of cancer treatment that is offered in specialized facilities. Treatment consists of multiple, almost daily irradiation appointments, followed by optional imaging and control assignments. The corresponding problem of scheduling these recurring radiotherapy treatment appointments can be classified as a complex job shop scheduling problem with custom constraints, such as recurring activities, optional activities, and special time window constraints. The objective is to minimize the operation time of the bottleneck resource, the particle beam, while simultaneously minimizing any penalties arising from violations of time window constraints. The authors model the problem mathematically and introduce various customized constraints. Three metaheuristic solution approaches—namely a genetic algorithm with tailor-made feasibility-preserving crossover operators, an iterated local search, and a combination of the two approaches—all perform well on both small and large problem instances. However, the simple combination of the two stand-alone algorithms leads to best results when applied to real-world inspired problem instances.

Keywords Patient scheduling · Recurring treatments · Optional activities · Stable starting times · Multi-encoded genetic algorithm · Iterated local search

1 Introduction

The worldwide number of patients diagnosed with cancer has steadily increased over the past decade, from approximately 10 million cases (and 6 million deaths) in 2003 to around 14.1 million cases (8.2 million deaths) in 2012 (Steward and Wild 2014). Projections for 2030 range between 17.1 and 22.2 million cases, which equals an increase of 21.3–57.4% (see Bray et al. 2012; World Health Organization 2012). Radiation therapy, or short radiotherapy, is commonly prescribed in addition to or instead of chemotherapy or surgery. The

goal is to deliver a maximum amount of radiation to kill the cancer while sparing the healthy tissue surrounding the tumorous region (Washington and Leaver 2016). In classical photon radiotherapy, which is available in virtually every hospital worldwide, radiation is delivered using a linear particle accelerator (linac) that supplies X-rays. More advanced but less numerous ion beam centers [only roughly 70 centers exist in the world, 48 of which have multiple treatment rooms, see PTCOG (2017)] use protons and/or carbon ions to achieve superior dose conformity and thereby lower the chances of patients developing secondary tumors later in life (Ohno 2013). However, because these ion beam centers also use significantly larger particle accelerators, their operations are much more costly than is classical radiotherapy using linacs and the efficient usage of the beam resource is crucial.

In this paper, we analyze and solve a real-world radiotherapy scheduling problem arising in a recently opened, specialized ion beam center close to Vienna, Austria, which offers two particle types for radiation treatment: protons and carbon ions. It plans to treat approximately 1000 patients per

✉ Petra Vogl
petra.vogl@univie.ac.at
Roland Braune
roland.braune@univie.ac.at
Karl F. Doerner
karl.doerner@univie.ac.at

¹ Department of Business Administration, University of Vienna, Oskar-Morgenstern-Platz 1, 1090 Vienna, Austria

year. A radiation treatment consists of both a pre-treatment phase and the irradiation treatment phase itself. During the pre-treatment phase, multiple examinations take place, followed by intensive treatment planning, during which radio-oncologists (RO), together with medical physicists, determine the dose of one treatment activity (called a “fraction”) and the number of fractions a patient should receive. They also define the particle type used and develop a so-called immobilization device, which helps the patient lie still during the treatment. The treatment phase then consists of a fixed number of daily treatment activities, imaging, and examination appointments, all of which require multiple resources simultaneously. Assigning treatments to days and scheduling the exact starting times of the activities to maximize facility usage and thereby minimize patients’ waiting times before they start treatment is of utmost importance.

The problem can be formalized as a complex job shop scheduling problem with multiple custom constraints that need to be considered. We introduce recurring optional appointments that, to the best of our knowledge, have not been considered in radiotherapy treatment scheduling before. Furthermore, we consider time windows between each activity for a patient that guarantee stable start times during the treatment phase. As solution techniques, we present three metaheuristics, namely a genetic algorithm and an iterated local search, whose operators are tailored to the problem at hand, as well as a combination of these two approaches in a third algorithm. In doing so, we achieve a highly effective method to solve the problem of scheduling recurring radiotherapy appointments in the ion beam facility. The time horizon of real-world problem instances reflects various weeks and contains up to 10,000 activities to be scheduled. Due to the high problem complexity of the long-term planning and the lack of reliable long-term stochastic data, we solve the problem deterministically. Future research will be dedicated to stochastic optimization of radiotherapy schedules for a given day d , because various disruptions can occur during the execution of a given schedule (e.g., room unavailability for a longer period). More specific data on past activity durations for a given patient then would be available (e.g., treatment duration of first five treatments took 12 min instead of estimated 8 min in the long-term planning) and could be used to revise the long-term schedule.

This article is organized as follows: Sect. 2 presents the formal problem statement of the radiotherapy patient scheduling problem (RPSP) and discusses the constraints that arise at ion beam facilities in particular. Section 3 gives insight into related work on the radiotherapy patient scheduling problem. Section 4 is devoted to the mathematical programming formulation of the underlying scheduling problem. In Sect. 5, we discuss the three heuristical solution methods—two stand-alone methods and one hybrid algorithm—the results of which are in Sect. 6. Finally, Sect. 7

offers some conclusions and proposes possible extensions to our work.

2 Problem statement

Ion beam facilities are typically equipped with one particle beam that serves multiple treatment rooms, as shown in Fig. 1. It depicts the specific problem setting we address, consisting of three treatment rooms with (1) horizontally directed, (2) vertically and horizontally directed, and (3) 180-degree rotatable particle beams. The particle beam—consisting of either protons or carbon ions—first moves through a linear accelerator, followed by multiple circulations through the synchrotron, where the beam gets accelerated to two-thirds of the speed of light, until it finally moves to one of the three treatment rooms. The beam can only serve one room at a time though, so we consider the particle beam the bottleneck resource in the irradiation process. Switching between the two mentioned particle types also requires a setup time of 3 min.

The treatment of one patient $p \in \mathcal{P}$ consists of a predefined number of recurring (almost) daily irradiation appointments [daily treatments, (DTs)] with fixed duration and resource requirements (e.g., treatment room, particle type, assigned RO). A patient needs to attend, on average, 20 daily treatments with an average irradiation duration of 8–10 min, depending on the particle type. Some patients need to attend regular, additional imaging appointments [positron emission tomography (PET)] directly after the DTs to ensure the accuracy of the irradiation treatment. Between DTs, patients regularly (i.e., once within a span of five consecutive days during the treatment phase) see their assigned RO for a control examination [weekly control examination (WCE)]. The sequence of these activities is fixed, as is shown by the “activity chain” in Fig. 2, where “FDT” and “LDT” denote the first and last DT, respectively. Note that both PET and WCE are listed after each DT in the activity chain. These activities are optional in the sense that they must take place once within every span of five consecutive days, but the opti-

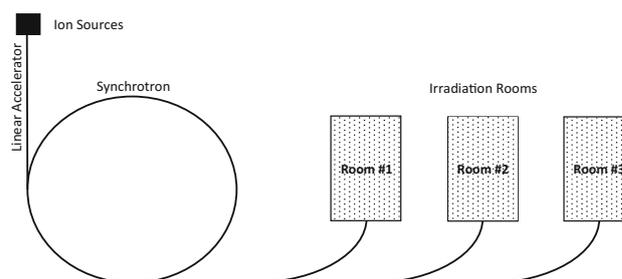


Fig. 1 Facility plan of MedAustron, Wiener Neustadt, Austria



Fig. 2 Activity chain

Fig. 3 Phases of an irradiation appointment

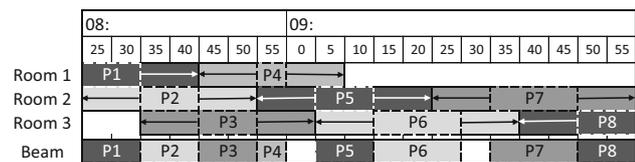
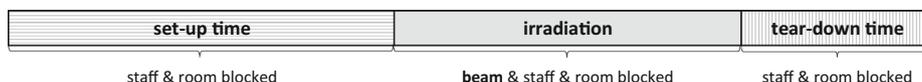


Fig. 4 Example schedule

mization algorithm must determine which WCE and PET activities should be scheduled and which ones can be skipped.

The FDT must be scheduled between the patient’s specific release date and due date. As mentioned, the irradiation appointments then take place almost every day, and only one DT can take place per day. Although the total number of DTs, N_p^{DT} , is fixed by the RO, the days to which the treatments are assigned may vary slightly: starting from the FDT until the day of the LDT, patients need at least four irradiation treatments within every span of five consecutive days. The treatment phase then corresponds to the time period between the FDT and the LDT.

Each DT activity consists of three inseparable tasks: (1) a setup time, in which the patient is prepared for the treatment inside the treatment room; (2) the irradiation itself, which requires the use of both the beam resource and the treatment room; and (3) a teardown time, during which only the treatment room is occupied. These three phases are depicted in Fig. 3. Scheduling two patients that require the same treatment room consecutively therefore causes extensive idle time on the beam resource (teardown time of the first patient and setup time of the second patient). Note, however, that the sum of the setup and teardown times equals an average of 18 min, whereas a treatment takes on average 8–12 min, depending on the particle type. Therefore, even alternating between two rooms would lead to beam idle time. An ideal schedule would interleave the three rooms, as depicted in Fig. 4. Here, even though the proposed scheduling pattern applies in the beginning, idle time on the beam resource between patient 4 and patient 5 is *unavoidable*, because patient 5 could not have started his setup in room 2 earlier. (The room was still blocked by the teardown of patient 2). The same applies to the idle time between patients 6 and 7. This situation makes it considerably harder to calculate a reasonable lower bound for the objective value, reflecting the total operation time of the beam (see Maschler et al. 2018, whose work is inspired by the same real-world problem).

The treatment activities within the activity chain of one patient are tied together using minimum and maximum time lags (“finish-start relations”). For example, to deliver accurate results, a PET appointment must start no later than 15 min after the preceding DT irradiation has finished. To maximize the patient’s convenience, the DT activities also should take place at approximately the same time on every treatment day during a week. We therefore introduce a daytime window for each week a patient receives treatment, which is defined by its midpoint. We call this midpoint the “stable starting time,” as the approximate time a patient comes in for treatment on each day of the given week. Only deviations smaller than 30 min from the defined stable starting time are accepted, creating an even tighter time window for the DT activities. Furthermore, the stable starting times of two consecutive weeks may only differ by a maximum of 4 h, allowing for changed approximate treatment times between weeks. Violations of the time windows formed by both finish-start relations and stable starting times result in penalties in the objective function.

Finally, some activities can be executed on alternative resource sets. For example, it might be preferable for the patient’s assigned RO to perform the WCE, but if he or she is busy, any other RO on duty can undertake the examination.

We aim to minimize the total operation time of the beam resource, which consists of the active time and the induced idle time and potential setup time due to particle switches, while the actual number of patients to be treated is determined by the doctors and is not part of the optimization. We thereby produce tight schedules which allow for the machine to be used for research in the field of medical physics and particle physics during the times when no patients are treated. Simultaneously, we minimize penalties arising from the belated starting times of activities that violate either general time window constraints or the patient-specific stable treatment starting time per week.

3 Related work

3.1 Radiotherapy scheduling

Appointment scheduling problems in health care systems arise in different environments, such as operating room scheduling, outpatient scheduling, and recurring treatment

appointment scheduling (as in radiotherapy scheduling) (e.g., Gupta and Denton 2008). Radiotherapy scheduling in particular has attracted the interest of research groups during the past two decades. It first appeared in the literature in a review paper by Kapamara et al. (2008), followed by basic algorithms for radiotherapy treatment booking proposed by Petrovic et al. (2006). Various studies model this problem mathematically and solve it using standard Mixed Integer Programming (MIP) solvers (e.g., Conforti et al. 2008, 2010, 2011; Burke et al. 2011). The different heuristics applied to the problem vary from pure constructive and hill climbing approaches (Kapamara and Petrovic 2009) to greedy randomized adaptive search procedures (GRASP) (Petrovic and Leite-Rocha 2008a) and (multi-objective) genetic algorithm (GA) approaches (Petrovic et al. 2009, 2011). Some authors focus on scheduling activities within the pre-treatment phase and use linear programming, simple dispatching rules, and GAs to solve this appointment scheduling problem (Petrovic and Castro 2011; Castro and Petrovic 2012). In a Ph.D. thesis, Leite-Rocha (2011) summarizes research on radiotherapy scheduling prior to 2011 and proposes various extensions to their mathematical models.

More recent publications consider stochastic and dynamic attributes of the radiotherapy scheduling problem. Sauré et al. (2012) formulate the model as discounted infinite-horizon Markov decision process to identify good policies for allocating capacity to incoming demand and thereby minimizing patient waiting times. Legrain et al. (2015) integrate stochastic patient arrival times into their model and develop a hybrid stochastic and online optimization algorithm. Gocgun (2016) also considers stochastic patient arrival times and introduces a simulation-based approximate dynamic programming approach to solve the problem. The Ph.D. thesis by Men (2009) centers the analysis on the optimal mix of patients and diagnoses for an ion beam facility, to maximize the throughput of patients. Lately, Vieira et al. (2016) published a literature review on radiotherapy resource planning and treatment scheduling and categorized the papers according to their hierarchy level, methods used, the extent of implementation and the potential impact on the performance. They conclude that future research could incorporate specialized clinical pathways and additional devices such as PETs.

This mentioned research stream thus mainly focuses on scheduling treatments for “classical” photon therapies, for which each treatment room is equipped with a distinct linear accelerator. Sequencing patients per day and thereby scheduling exact starting times for all patients is less crucial in these settings, and accordingly, two main strategies for scheduling activities within the treatment phase dominate prior literature:

1. Assign treatments to days. This approach incorporates an average resource profile and does not schedule exact starting times on each day. Therefore, it requires a second step, namely patient sequencing per day (Petrovic and Leite-Rocha 2008b; Men 2009; Conforti et al. 2010; Burke et al. 2011; Sauré et al. 2012).
2. Split the day into time slots of predefined lengths (e.g., 15 or 30 min) and allocate the treatments to these time slots (i.e., “block scheduling,” Conforti et al. 2008, 2011; Legrain et al. 2015). This approach allows for the immediate consideration of stable activity starting times, but it also assumes that the treatment duration will be more or less equal to the length of the time slot, which is not the case in ion beam therapy, for which treatment durations vary substantially according to the diagnosis received by the patient.

This vast variation in treatment durations, as well as the bottleneck resource “particle beam” that is shared among various treatment rooms, necessitates scheduling exact (“to-the-minute”) starting times at ion beam facilities. Maschler et al. (2016) propose a detailed scheduling approach using both a GRASP procedure and an iterated greedy approach, which incorporates interconnected day and time assignment phases. They extend the latter approach in a subsequent study and yield even better results on a midterm planning horizon (Maschler et al. 2017). However, they do not incorporate optional activities, and they focus on scheduling the core irradiation appointments. Using a surrogate model to estimate the lower bound for the time the beam resource is required if the patients treated on the specific day are known (i.e., the day assignment phase has already finished), Maschler et al. (2018) also apply this information iteratively to optimize the day assignment.

4 Problem formulation

The objective function and constraints described in Sect. 2 can be formulated mathematically. Table 1 lists the symbols and sets used in the formulation of the problem; Table 2 summarizes all necessary input information; and Table 3 gives an overview of the decision variables of the mathematical modeling formulation.

The objective function minimizes the operation time and thereby the idle time of the beam while simultaneously minimizing penalties arising from time window violations of activity i of patient p and stable time violations for patient p and day d . Here, f_d denotes the operation time of the beam resource on day d , $\tilde{\gamma}_{pd}$ describes penalty caused by violations of the stable starting time [see Eq. (19)] and $\hat{\gamma}_{pi}$ accounts for general time window violations [see Eq. (11)]. All three parts of the objective function are measured in time unites (min),

Table 1 Sets of the mathematical modeling formulation

Notation	Description
<i>Sets</i>	
\mathcal{P}	Set of all patients, index $p \in \{1, \dots, P\}$
\mathcal{O}_p	Set of operations/activities of patient p
\mathcal{R}	General set of resources, index $r \in \{1, \dots, R\}$
\mathcal{K}_{pi}	Set of resource requirements for patient p 's activity i
\mathcal{R}_{pik}	Set of eligible resources for activity i and patient p , and resource requirement k . If only one (compulsory) resource is available, then $ \mathcal{R}_{pik} = 1$
\mathcal{D}	Set of days in the planning horizon, index $d \in \{1, \dots, D\}$
\mathcal{W}	Set of weeks in the planning horizon, index $w \in \{1, \dots, W\}$
Φ_p	Set of activities belonging to the subgroup of daily treatment activities for patient p
Ψ_p	Set of activities belonging to the subgroup of weekly control examination activities for patient p
Ξ_p	Set of activities belonging to the subgroup of PET activities for patient p

Table 2 Parameters and input variables to the mathematical modeling formulation

Notation	Description
<i>Parameters and input variables</i>	
u_{piqjr}	Setup time between activity i for patient p and succeeding activity j (patient q) on resource r
v_{pir}	Setup time of activity i for patient p on resource r
w_{pir}	Teardown time of activity i for patient p on resource r
λ_{pi}	Processing time of activity i of patient p on all resources
$FS_{pi,p(i+\delta)}^{\min}$	Minimum time from finish of activity i to start of activity $i + \delta$
$FS_{pi,p(i+\delta)}^{\max}$	Maximum time from finish of activity i to start of activity $i + \delta$
$\underline{dw}_d, \overline{dw}_d$	Begin and end of daytime window of day d
α'	Maximum intra-week variation from the stable starting time
α''	Maximum inter-week variation from the stable starting time
r_p	Release day for patient p 's FDT
d_p	Due day for patient p 's FDT
M	Large number

Table 3 Variables of the mathematical modeling formulation

Notation	Description
<i>Decision variables</i>	
h_{pikr}	Binary variable, set to 1 if activity i of patient p is assigned to resource r for resource requirement k
s_{pir}	Starting time of activity i for patient p on resource r
\bar{s}_{pi}	Starting time of activity i for patient p
γ_{piqjr}	Binary variable for immediate successor of activity i for patient p (namely qj) on resource r
o_{pi}	Binary variable, indicating whether activity i for patient p takes place or not
θ_{pid}	Binary variable, indicating whether activity i takes place on day d or not ($= 1$ if $\underline{dw}_d \leq \bar{s}_{pi} \leq \overline{dw}_d$)
t_{pd}	Starting time of treatment for patient p on day d (independent of the activity i , time between the start of daytime window and the scheduled starting time)
\tilde{t}_{pw}	Stable starting time of treatment for patient p in week w
\bar{x}_{pd}	Binary variable, indicating whether day d is within the treatment phase of patient p
\bar{z}_{pw}	Binary variable, indicating whether week w contains the treatment phase of patient p
f_d	Finish time of last activity on the beam resource on day d
$\tilde{\gamma}_{pd}$	Stable time violation on day d for patient p
$\hat{\gamma}_{pi}$	Time window violation for patient p 's activity i

i.e., 1 min of extra beam duration accounts for 1 min of delay for the patient (either time window or stable time violation). As will be shown in Table 10 in Sect. 6, the latter two parts of the objective function tend to almost disappear during the optimization process.

$$\text{minimize } \sum_{d \in \mathcal{D}} f_d + \sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}} \tilde{\gamma}_{pd} + \sum_{p \in \mathcal{P}} \sum_{i \in \mathcal{O}_p} \hat{\gamma}_{pi}. \quad (1)$$

The model's constraints can be categorized into various subsections: resource constraints, sequencing and optional activities, linking constraints, recurring activities, stable activity starting times, and general treatment start time constraints.

4.1 Resource constraints

Each activity i requires K_{pi} resources simultaneously for every patient p . For some resource requirements, there also exist multiple alternative resources r , defined by the set of eligible resources for resource requirement k , namely \mathcal{R}_{pik} :

$$\sum_{r \in \mathcal{R}_{pik}} h_{pikr} = o_{pi} \quad \forall p \in \mathcal{P}, i \in \mathcal{O}_p, k \in \mathcal{K}_{pi}. \quad (2)$$

Constraint (2) assures that for each required resource, one of the eligible resources is chosen if the activity takes place (i.e., variable $o_{pi} = 1$). Then,

$$s_{pir} \leq h_{pikr} \cdot M \quad \forall p \in \mathcal{P}, i \in \mathcal{O}_p, k \in \mathcal{K}_{pi}, r \in \mathcal{R}_{pik}, \quad (3)$$

$$\sum_{r \in \mathcal{R}_{pik}} s_{pir} = \bar{s}_{pi} \quad \forall p \in \mathcal{P}, i \in \mathcal{O}_p, k \in \mathcal{K}_{pi}. \quad (4)$$

Constraint (3) fixes the starting times of non-chosen resources to 0. Constraint (4) assigns the exact same starting times \bar{s}_{pi} to all resources chosen for activity i of patient p , as the sum over all starting times on all eligible resources has only one positive entry per resource requirement k due to the previous constraint. In turn,

$$\sum_{\substack{i' \in \mathcal{O}_p \\ i' > i}} y_{pi'ir} + \sum_{q \in \mathcal{P} \setminus \{p\}} \sum_{j \in \mathcal{O}_q} y_{piqjr} = h_{pikr} \quad \forall p \in \mathcal{P}, i \in \mathcal{O}_p, k \in \mathcal{K}_{pi}, r \in \mathcal{R}_{pik}, \quad (5)$$

$$\sum_{\substack{j' \in \mathcal{O}_q \\ j' < j}} y_{qj'qjr} + \sum_{p \in \mathcal{P} \setminus \{q\}} \sum_{i \in \mathcal{O}_p} y_{piqjr} = h_{qjkr} \quad \forall q \in \mathcal{P}, j \in \mathcal{O}_q, k \in \mathcal{K}_{qj}, r \in \mathcal{R}_{qjk}. \quad (6)$$

Constraints (5) and (6) thus give the immediate successor structure of activities on resource r . Finally,

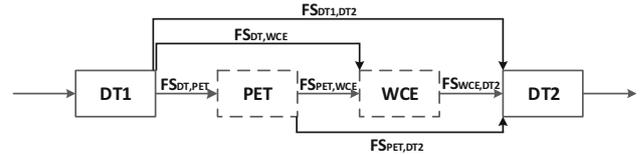


Fig. 5 Finish-start relations among optional activities

$$s_{pir} + \lambda_{pi} + u_{piqjr} \cdot y_{piqjr} - (1 - y_{piqjr}) \cdot M \leq s_{qjr} \quad \forall p \in \mathcal{P}, i \in \mathcal{O}_p, q \in \mathcal{P}, j \in \mathcal{O}_q, r \in \mathcal{R}, \quad (7)$$

$$s_{pir} + \lambda_{pi} + w_{pir} \cdot y_{piqjr} + v_{qjr} \cdot y_{piqjr} - (1 - y_{piqjr}) \cdot M \leq s_{qjr} \quad \forall p \in \mathcal{P}, i \in \mathcal{O}_p, q \in \mathcal{P}, j \in \mathcal{O}_q, r \in \mathcal{R}. \quad (8)$$

Constraints (7) and (8) confirm that both the sequence-dependent setup times and the non-sequence-dependent setup and teardown times are respected.

4.2 Sequencing and optional activities

Each activity i of patient p is associated with a binary variable o_{pi} that indicates whether the activity takes place or not. It is essential for optional activities (PETs and WCEs), and for DTs, this variable must equal 1. The starting time of activities that do not take place is then set to 0, using Constraint (9):

$$\bar{s}_{pi} \leq o_{pi} \cdot M \quad \forall p \in \mathcal{P}, i \in \mathcal{O}_p. \quad (9)$$

The problem of scheduling the activities of patient p considering the precedence relations (finish-start, FS) is further complicated by the optional activities within the activity chain, as shown in Fig. 5. If, for example, a PET does not take place, the DT and WCE need to be tied together using the corresponding FS. However, if the PET is scheduled, the link between the DT and the WCE should be deactivated. In Constraints (10) and (11), successive activities that actually take place are connected using the minimum and maximum time lags between them. Constraint (11) further quantifies the time window violation (belated scheduling) that is penalized within the objective function. Furthermore, Δ^{\max} denotes the maximum number of consecutive optional activities, in our case, $\Delta^{\max} = 2$.

$$\bar{s}_{p(i+\delta)} \geq \bar{s}_{pi} + \lambda_{pi} + \text{FS}_{pi,p(i+\delta)}^{\min} - \sum_{j=i}^{j=i+\delta} (M \cdot (1 - o_{pj})) \quad \forall p \in \mathcal{P}, i \in \mathcal{O}_p, \delta \in \{1, \dots, \Delta^{\max} + 1\}, \quad (10)$$

$$\bar{s}_{p(i+\delta)} \leq \bar{s}_{pi} + \lambda_{pi} + \text{FS}_{pi,p(i+\delta)}^{\max} + \sum_{j=i}^{j=i+\delta} (M \cdot (1 - o_{pj})) + \hat{\gamma}_{pi} \quad \forall p \in \mathcal{P}, i \in \mathcal{O}_p, \delta \in \{1, \dots, \Delta^{\max} + 1\}. \quad (11)$$

4.3 Linking constraints

Constraints (12) to (15) serve as linking constraints. If an activity starts within a daytime window, the corresponding assignment variable θ_{pid} equals 1. Constraint (13) also calculates the general daily starting time by subtracting the day start from the scheduled starting time. The treatment phase (days between FDT and LDT) of patient p can be calculated for both day d and week w using Constraints (14) and (15). Note that we display these constraints as nonlinear indicator constraints for better readability. The numeric tests in Sect. 6 use linearized versions of these constraints.

$$\theta_{pid} = 1 \iff \underline{dw}_d \leq \bar{s}_{pi} \leq \overline{dw}_d \quad \forall p \in \mathcal{P}, i \in \mathcal{O}_p, d \in \mathcal{D}, \tag{12}$$

$$t_{pd} = \sum_{i \in \Phi_p} \bar{s}_{pi} \cdot \theta_{pid} - \underline{dw}_d \iff \sum_{i \in \Phi_p} \theta_{pid} = 1 \quad \forall p \in \mathcal{P}, d \in \mathcal{D}, \tag{13}$$

$$\bar{x}_{pd} = 1 \iff \sum_{0 \leq d' \leq d} \sum_{i \in \mathcal{O}_p} \theta_{pid'} \geq 1 \wedge \sum_{d \leq d' \leq D} \sum_{i \in \mathcal{O}_p} \theta_{pid'} \geq 1 \quad \forall p \in \mathcal{P}, d \in \mathcal{D}, \tag{14}$$

$$\bar{z}_{pw} = 1 \iff \sum_{d=5 \cdot w - 4}^{5 \cdot w} \bar{x}_{pd} \geq 1 \quad \forall p \in \mathcal{P}, w \in \mathcal{W}. \tag{15}$$

4.4 Recurring activities

Constraint (16) ensures that at least four daily treatments are scheduled on five consecutive days within the treatment phase. Constraints (17) and (18) guarantee that at least one WCE (PET) is performed within five consecutive days during the treatment phase, respectively.

$$\sum_{d=s}^{s+4} \sum_{i \in \Phi_p} \theta_{pid} + M \cdot (1 - \bar{x}_{pd}) \geq 4 \quad \forall p \in \mathcal{P}, 1 \leq s \leq D - 4, \tag{16}$$

$$\sum_{d=s}^{s+4} \sum_{i \in \Psi_p} \theta_{pid} + M \cdot (1 - \bar{x}_{pd}) \geq 1 \quad \forall p \in \mathcal{P}, 1 \leq s \leq D - 4, \tag{17}$$

$$\sum_{d=s}^{s+4} \sum_{i \in \Xi_p} \theta_{pid} + M \cdot (1 - \bar{x}_{pd}) \geq 1 \quad \forall p \in \mathcal{P}, 1 \leq s \leq D - 4. \tag{18}$$

4.5 Stable activity starting times

Inequalities (19) and (20) constrain the daily starting time for each patient to the stable starting time of the corresponding week and the stable starting times of two consecutive weeks, respectively:

$$|t_{pd} - \tilde{t}_{pw}| - M \cdot (1 - \sum_{i \in \Phi_p} \theta_{pid}) \leq \alpha' + \tilde{\gamma}_{pd} \quad \forall p \in \mathcal{P}, w \in \mathcal{W}, 5w - 4 \leq d \leq 5w, \tag{19}$$

$$|\tilde{t}_{pw} - \tilde{t}_{p(w+1)}| - M \cdot (1 - \bar{z}_{pw}) - M \cdot (1 - \bar{z}_{p(w+1)}) \leq \alpha'' \quad \forall p \in \mathcal{P}, w \in \mathcal{W}. \tag{20}$$

4.6 Treatment starting time

No treatments can be assigned to patient p prior to his release day r_p , so

$$\sum_{i \in \Phi_p} \sum_{d=0}^{r_p-1} \theta_{pid} = 0 \quad \forall p \in \mathcal{P}. \tag{21}$$

However, there has to be at least one treatment scheduled for patient p between his release day and due day, so

$$\sum_{i \in \Phi_p} \sum_{d=r_p}^{d_p} \theta_{pid} \geq 1 \quad \forall p \in \mathcal{P}. \tag{22}$$

4.7 Active time of beam resource

Finally, we calculate the active time of the beam according to Constraint (23):

$$f_d \geq t_{pd} + \sum_{i \in \Phi_p} \lambda_{pi} \cdot \theta_{pid} \quad \forall d \in \mathcal{D}, p \in \mathcal{P}. \tag{23}$$

5 Solution methods

In this section we present three metaheuristic approaches to the proposed radiotherapy scheduling problem, to tackle the problem efficiently (as we show in Sect. 6, solving the exact MIP formulation, even for small to medium-sized problem instances, is intractable). We compare two metaheuristic paradigms, namely a population-based GA approach with a trajectory-based local search heuristic and combine the two to a simple hybrid algorithm. Both methods have successfully been applied to related radiotherapy scheduling problems [e.g., GAs have been used in Petrovic et al. (2009, 2011), while local search has been performed in Petrovic and Leite-Rocha (2008a) and Kapamara and Petrovic (2009)].

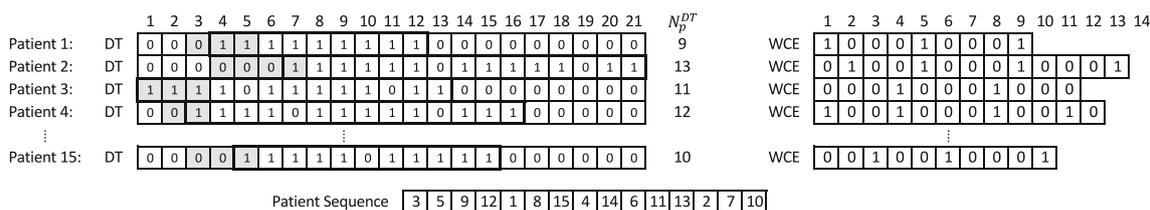


Fig. 6 Solution encoding, 15 patients

Section 5.1 describes how we map the solution attributes to a multi-encoded solution representation. Section 5.2 gives some insight into the complexity of the problem and the proposed encoding scheme. In Sect. 5.3, we describe a greedy randomized method for creating initial solutions to the problem, followed by a detailed description of the decoding algorithm that transforms the solution encoding into a schedule with exact starting times in Sect. 5.4. Finally, Sects. 5.5 to 5.7 present the three distinct solution methods: a GA, an iterated local search method (ILS), and a combination of the two approaches (combined GA and ILS, briefly cGAILS).

5.1 Solution representation

A solution to the RPSP is represented by a multi-encoded scheme that consists of three main parts. The first part (“DT assignment”) contains, for every patient $p \in \mathcal{P}$, an assignment of treatments to days. Days prior to the release time of the patient automatically remain unassigned. Between the FDT and the LDT (i.e., treatment phase), at least four treatments must be assigned to five consecutive days for the solution to be feasible. In the second part, two binary vectors for each patient $p \in \mathcal{P}$ indicate, when (i.e., after which DT) a WCE and a PET is scheduled (“WCE/PET assignment”). An entry at the i th position in these vectors implies that a WCE (PET) has been added after the i th DT activity. We must ensure a minimum of one WCE (PET) within each span of five consecutive days, resulting in at least one entry in the binary encoding over four consecutive DTs. Finally, the third part of the solution encoding consists of a vector of patient indices that displays the sequence in which the patients should be scheduled on a specific day (“patient sequence”).

Figure 6 illustrates a solution of an RPSP for 15 patients ($P = 15$), where N_p^{DT} denotes the patient-specific number of required DTs. The bold frame within the DT assignment marks the treatment phase. The gray-shaded cells indicate the release and due dates for the FDT. Out of space considerations, we only display the WCE assignments; the PET assignments would reveal different allocations but the same sizes (N_p^{DT} of the patient p). Here, patient 1 starts his treatment on day 4. Directly after his first DT, a WCE is scheduled.

The patient sequence lists patient 1 in the fifth position, so all predecessors will be scheduled prior to patient 1 on day d using the solution decoding algorithm (see Sect. 5.4).

5.2 Excursus on problem complexity

The presented solution representation already gives insight into the complexity of the underlying problem. The number of permutations of the DT assignment list for one patient p depends strongly on the number of DTs to be scheduled for this patient, namely N_p^{DT} and can be calculated using Equation (24) (assuming the day of the FDT is fixed for patient p), with h_p^{\max} denoting the maximum number of unassigned days allowed during the treatment phase. Given the above information, we define:

$$g(p) := \sum_{r=0}^{h_p^{\max}} \binom{N_p^{DT} - 3 \cdot r + 2}{r} \tag{24}$$

On average, a radiotherapy treatment consists of 20 DTs, which allows a maximum of five unassigned days during the treatment phase ($h_p^{\max} = 5$) and a total of 657 feasible DT-to-day assignments, given the day of the FDT is fixed. A real-world instance would contain an average of 100 patients, each of which is assigned to an individual permutation list of DT assignments. Additionally, for each patient p , there exist on average four different “minimally occupied” WCE and PET assignments and a multiplicity of “non-minimally occupied” assignments. Finally, given P , or the number of patients to receive radiotherapy, $P!$ permutations of the patient sequence exist. We then calculate the number of possible permutations by multiplying the three parts: $P! \times \prod_{p=1}^P g(p) \times 4^P \times 4^P$.

Therefore, instances including only five patients ($P = 5$) with 20 treatments each ($N_p^{DT} = 20$) and a fixed treatment start day already would imply $P! = 5! = 120$ permutations of the patient sequence, for each patient 657 feasible DT-to-day assignments, i.e., $\prod_{p=1}^5 657 = 1.22 \times 10^{14}$ possible assignments and both $4^5 = 1024$ possible WCE and PET assignments. This results in a total of 1.54×10^{22} solution permutations.

5.3 Initial solutions

Initial, partly randomized solutions can be created using simple construction rules. We apply different strategies to the parts of the solution representation: For each patient, we first randomly fix the day of his or her FDT, noting the release and due days. Then, the subsequent treatment days are fixed, with the premise that four treatments must be scheduled within five consecutive days. The probability of leaving one specific day d unscheduled (i.e., four prior days have a DT scheduled) is rather small, ranging between 0 and 30%. The same strategy applies for the WCE and PET assignments, where only one activity must be assigned over four consecutive DTs.

Building the patient sequence requires a more sophisticated method though. We build a “global” patient sequence for all patients, in which we neglect the fact that some patients by definition will not be treated on the same day (because their release time will be later than other patient’s LDT day, resulting in non-overlapping treatment phases). Using the assumption that all patients must be treated on a fictitious day d^* , we choose a random patient, whom we add as the first patient in the patient sequence. Then, we continuously add one more patient, who minimizes the total idle time of the beam, due to either room unavailability (i.e., the teardown time of the previous patient is not yet finished when the setup of the current patient should start) or setup time due to particle type switches (from proton to carbon ion or vice versa). This strategy results in a starting solution; only in rare cases are two patients requiring the same treatment room scheduled successively.

5.4 Solution decoding and solution evaluation

To decode this described solution representation into a schedule that provides exact starting times and resource decisions for each activity, we first transform the solution into a chronological (i.e., day-wise) prioritized activity list. Then, using Algorithm 1, we determine the exact starting time and the resource set on which the activity should be performed. We begin the search for a feasible starting time on the “preferred” resource set ($n = 1$). We use function $\text{FINDSTARTINGTIME}(i, n, \bar{s}_{pi}, l_{pi})$ to determine the earliest starting time for activity i on resource set n , with \bar{s}_{pi} as the earliest time to start and as l_{pi} the latest possible starting time. We first search all N_{pi} -eligible resource sets for a *feasible* starting time (i.e., a smaller than or equal to l_{pi}). If no such starting time arises from any resource set, the first non-feasible (i.e., belated) starting time on the “preferred” resource set is accepted as the starting time for activity i , though it results in a penalty within the objective function.

Note that function $\text{FINDSTARTINGTIME}(i, n, \bar{s}_{pi}, l_{pi})$ in Algorithm 2 does not necessarily add activities to resources chronologically. Activities can be scheduled to fill up “holes”

Algorithm 1: Solution decoding algorithm

```

1 repeat
2   Determine next activity  $i$  to be scheduled from the activity
   list;
3    $n \leftarrow 1$ ;
4    $\bar{s}_{pi} \leftarrow \infty$ ;
5    $l_{pi} \leftarrow$  latest feasible starting time of activity  $i$ ;
6   while  $\bar{s}_{pi} > l_{pi} \wedge n < N_{pi}$  do
7      $\bar{s}_{pi} \leftarrow$  earliest feasible starting time of activity  $i$ ;
8      $\bar{s}_{pi} \leftarrow \text{FINDSTARTINGTIME}(i, n, \bar{s}_{pi}, l_{pi})$ ;
9      $n \leftarrow n + 1$ ;
10  end
11  if  $\bar{s}_{pi} > l_{pi}$  then
12     $\bar{s}_{pi} \leftarrow \text{FINDSTARTINGTIME}(i, 1, l_{pi}, \infty)$ ;
13  end
14  Schedule activity  $i$  at the determined starting time  $\bar{s}_{pi}$ ;
15  Update time windows of the successive activities;
16 until all activities scheduled;
```

Algorithm 2: $\text{FINDSTARTINGTIME}(i, n, \bar{s}_{pi}, l_{pi})$

```

1 while  $\bar{s}_{pi} < l_{pi}$  do
2   for all required resources  $r$  of resource set  $n$  do
3     check availability of resource  $r$  for the time period of the
     processing time  $\lambda_{pi}$  including setup time  $v_{pir}$  and
     teardown time  $w_{pir}$ : [ $\bar{s}_{pi} - v_{pir}, \bar{s}_{pi} + \lambda_{pi} + w_{pir}$ ];
4     if resource unavailable then
5        $s_r \leftarrow$  start of next idle time window of resource  $r$ ;
6       if  $\bar{s}_{pi} < s_r$  then
7          $\bar{s}_{pi} \leftarrow s_r$ ;
8       end
9     end
10  end
11 end
12 return  $\bar{s}_{pi}$ ;
```

in resources, such as might occur if we were to schedule two treatment activities in the same treatment room successively and therefore face idle time on the beam resource due to teardown and setup times. We then would aim to schedule activities requiring any other treatment room in between those activities to minimize the idle time on the beam resource. This approach has been proven beneficial in our problem setting (see Vogl et al. 2018).

The sequential nature of the decoding algorithm requires a post-scheduling evaluation of the treatment starting times to evaluate the stable time violations, because the stable time for each patient p and each week w needs to be assigned “globally” and not when scheduling the first activity of the week. Therefore, we solve a small linear program for each patient p to reveal stable time violations. The model contains only two variable types, namely, \tilde{t}_w , the stable time of week w , and γ_d , the stable time violations on day d . The daily starting time t_d , the day assignment variable θ_d , and the weeks within the treatment phase \mathcal{W} have already been fixed during the solution decoding phase and are therefore input parameters. Accordingly,

$$\text{minimize } \sum_{d \in \mathcal{D}} \gamma_d \tag{25}$$

subject to:

$$|t_d - \tilde{t}_w| - \gamma_d - M \cdot (1 - \theta_d) \leq \alpha' \tag{26}$$

$$\forall w \in \mathcal{W}', 5w - 4 \leq d \leq 5w, \tag{26}$$

$$|\tilde{t}_w - \tilde{t}_{w+1}| \leq \alpha'' \quad \forall w \in \mathcal{W}'. \tag{27}$$

The objective function is to minimize the sum of daily deviations from the stable starting times. These deviations are calculated using Constraint (26). Constraint (27) then assures that the inter-week stable time variation is smaller than the maximum allowed deviation, namely α'' .

5.5 Genetic algorithm

Genetic algorithms have been implemented successfully to solve the radiotherapy patient scheduling problem (Petrovic et al. 2009, 2011). In a preliminary study, we introduced the solution decoding presented in Sect. 5.4 and presented initial tests of different decoding algorithms and GA settings, applied within this research setting (Vogl et al. 2018). We enhance this strategy here by introducing a more sophisticated crossover mechanism, in addition to the patient-wise crossover presented in the previous work. Algorithm 3 gives an overview of the mentioned GA components. Note that the algorithm contains two sets of solutions: \mathcal{P}_t forms the current population during iteration t and C^B contains children that did not reach the success criterion of being better than at least their worst parent (see line 12 in the algorithm) and therefore do not immediately contribute to the next generation. The latter solutions are potentially used if by producing five times the population’s size as offspring is not sufficient to build the new population from successful children.

CROSSOVER(p_1, p_2) The multi-encoded solution representation demands specific crossover and mutation operators for the individuals to remain feasible throughout the evolutionary process. We use two crossover operators: (1) The patient-wise crossover operator presented by Vogl et al. (2018), where the whole DT assignment as well as the whole PET and WCE lists of one patient p is randomly inherited either by the first or the second parent. This simple crossover only leads to limited variety in the binary encodings, because the combinations of the initial population dominate the search. So we developed a (2) a tailor-made, day-wise crossover operator for the DT assignment part of the multi-encoded chromosome, illustrated in Fig. 7. The day-wise crossover chronologically compares entries of the parents on a specific day d . If both parents have a DT assigned on day d , we naturally assign a treatment on this day (white

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
DT[patient 1] - Parent #1	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
DT[patient 1] - Parent #2	0	0	1	1	0	1	1	1	1	0	1	1	1	0	0
	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=
DT[patient 1] - Child	0	0	1	1	0	1	1	1	1	1	1	1	0	0	0

Fig. 7 Day-wise crossover

entries). The same applies for the cases in which no treatment is planned on day d , observable by two “0” entries in the parent chromosomes. Gray-shaded cells indicate days with different allocations among the parents. The assignment of every yet undecided day d is fixed randomly, again defining the allocations chronologically. We first have to determine if leaving day d unassigned leads to an infeasible solution regarding the constraint of assigning four treatments within every five consecutive days. If so, we assign a treatment to this day in any case.

The probability of assigning a treatment to day d depends on the number of still missing DTs for patient p , n_{missing} , and the number of undecided days, $n_{\text{undecided}}$: $P(\text{DT}_d) = \frac{n_{\text{missing}}}{n_{\text{undecided}}}$. We call this crossover strategy the “day-wise crossover operator.”

The day-wise crossover and patient-wise crossover then are chosen randomly during the genetic evolution of the DT assignments with equal probabilities. The WCE and PET assignments use only the patient-wise crossover operator. Finally, we apply the position-based crossover operator (PBX) to the patient sequences of two parent solutions to receive the child’s patient sequence (Syswerda 1996). **MUTATE(c)** To enhance the search, we apply mutation operators to 10% of the descendants in each generation. The DT assignment per patient is mutated by inverting the list within the treatment phase. The WCE and PET assignments are completely reversed, and the patient sequence is mutated using the well-known shift mutation operator, such that one random patient p is removed from the current sequence and reinserted at a random position within the sequence.

To choose individuals for reproduction [PERFORM SELECTION(\mathcal{P}_i)], we employ the rank selection operator. Because offspring selection (Affenzeller and Wagner 2004) is beneficial in our problem setting (Vogl et al. 2018), we again incorporate this strategy into our GA. We aim to build 70% of the offspring population from children that outperform their worse parent (see line 7 and lines 12–17 in Algorithm 3). The number of reproductive steps is limited to five times the population size in each iteration (see line 7). The rest of the population is then filled up with random individuals that did not outperform their worse parent (lines 19–23). In addition, 1% of the offspring population is composed of the best individuals of the parent population [GETELITES(\mathcal{P}_i)].

Algorithm 3: GA; see Affenzeller and Wagner (2004)

```

1  $\mathcal{P}_0 \leftarrow \text{CREATEINITIALPOPULATION}()$ ;
2  $s_{best} \leftarrow \arg \min_{p \in \mathcal{P}_0} (\text{ObjVal}(p))$ ;
3  $i \leftarrow 0$ ;
4 repeat
5    $\mathcal{P}_{i+1} \leftarrow \text{GETELITES}(\mathcal{P}_i)$ ;
6    $C^B \leftarrow \emptyset$ ;
7   while  $|\mathcal{P}_{i+1}| < 0.7 \cdot |\mathcal{P}_i| \wedge |\mathcal{P}_{i+1}| + |C^B| < 5 \cdot |\mathcal{P}_i|$  do
8      $p_1 \leftarrow \text{PERFORMSELECTION}(\mathcal{P}_i)$ ;
9      $p_2 \leftarrow \text{PERFORMSELECTION}(\mathcal{P}_i)$ ;
10     $c \leftarrow \text{CROSSOVER}(p_1, p_2)$ ;
11     $c \leftarrow \text{MUTATE}(c)$ ;
12    if  $\text{ObjVal}(c) < \min(\text{ObjVal}(p_1), \text{ObjVal}(p_2))$  then
13       $\mathcal{P}_{i+1} \leftarrow \mathcal{P}_{i+1} \cup \{c\}$ ;
14    else
15       $C^B \leftarrow C^B \cup \{c\}$ ;
16    end
17  end
18  while  $|\mathcal{P}_{i+1}| < |\mathcal{P}_i|$  do
19     $c \leftarrow \text{CHOOSERANDOMELEMENT}(C^B)$ ;
20     $\mathcal{P}_{i+1} \leftarrow \mathcal{P}_{i+1} \cup \{c\}$ ;
21     $C^B \leftarrow C^B \setminus \{c\}$ ;
22  end
23  if  $\arg \min_{p \in \mathcal{P}_{i+1}} (\text{ObjVal}(p)) < \text{ObjVal}(s_{best})$  then
24     $s_{best} \leftarrow \text{BESTOFPOPULATION}(\mathcal{P}_{i+1})$ ;
25  end
26   $i \leftarrow i + 1$ ;
27 until termination criterion met;

```

Algorithm 4: ILS; see Lourenço et al. (2010)

```

1  $s_0 \leftarrow \text{GENERATEINITIALSOLUTION}$ ;
2  $s^* \leftarrow \text{LocalSearch}(s_0)$ ;
3  $s_{best} \leftarrow s^*$ ;
4 repeat
5    $s' \leftarrow \text{PERTUBATION}(s^*)$ ;
6    $s^{*'} \leftarrow \text{LOCALSEARCH}(s')$ ;
7   if  $\text{ObjVal}(s^{*'}) < \text{ObjVal}(s_{best})$  then
8      $s_{best} \leftarrow s^{*'}$ ;
9   end
10   $s^* \leftarrow \text{ACCEPTANCECRITERION}(s^*, s^{*'})$ ;
11 until termination condition met;

```

5.6 Iterated local search

As an alternative approach to the GA, we introduce an ILS to solve the RPSP, where the local search step of the algorithm is formed by a variable neighborhood descent (VND). Algorithm 4 gives an overview of the classic ILS (Lourenço et al. 2010) components, all of which are described in more detail in the following paragraphs.

GENERATEINITIALSOLUTION The initial solution of the ILS is defined by the best solution found within a pool of randomly generated solutions, which are constructed using a partly greedy, partly random approach described in Sect. 5.3. The pool size depends on the instance size with larger, real-world inspired instances in Sect. 6 having a pool of 200 initial solutions to choose from.

LOCALSEARCH The local search part of the algorithm is formed by a VND, which operates on different parts of the solution representation in Sect. 5.1. Traditionally, the VND contains various different neighborhoods with increasing degree of disruptiveness, which allows the algorithm to leave potential local minima. Preliminary tests have shown that in our specific case, the following $k_d = 6$ neighborhoods contribute to the search for improvements to the solution fitness. For details on these preliminary tests please refer to Sect. 6.2.

1. Perform a random shift of a patient within the patient sequence (N1).
2. Invert the DT assignment of a random patient (N2).
3. Invert both the WCE and PET assignments of a random patient (N3).
4. Swap two random patients within the patient sequence (N4).
5. Rebuild the DT, WCE, and PET vectors for a random patient from scratch (random assignment, as in the starting solutions, (N5).
6. Invert a random subsequence of a size between 2 and 5 of the patient sequence (N6).

If a better solution is found, the VND continues its search within the first neighborhood. In case no better solution can be found in any of the listed neighborhoods, we reach a local minimum, leading to the termination of the VND. Because the high complexity of the underlying problem leads to vast neighborhood sizes, we impose a limit on the number of evaluated neighbors in each iteration of the VND (see Sect. 6 for details). Experiments have shown that the best improvement policy during the neighborhood search of the VND is beneficial, which is why we follow this scheme.

PERTUBATION If the local search gets stuck in a local minimum, a perturbation of the current solution is performed, to leave the local minimum and restart the local search phase from another starting point. We use two different perturbation strategies, one less and one more invasive method. Any time a local minimum is found that improves the global best, we use the less invasive perturbation method. Otherwise, we alternate these approaches.

The less invasive method inverts the DT, WCE, and PET assignments of a random patient, and it also inverts a random part of the patient sequence of size $P/7$. The second perturbation strategy rebuilds a completely new assignment of DTs, WCEs, and PETs for a random patient (as in neighborhood 5 of the VND) and inverts a random part of the patient sequence of size $P/5$.

ACCEPTANCECRITERION The acceptance criterion determines whether solution $s^{*'}$, found by the latest local search step, replaces the current incumbent solution s^* . Lourenço et al. (2010) describe two extremes of acceptance criteria dur-

ing ILS. Either the new solution is accepted only if it improves s^* (i.e., “better” acceptance criterion), or it is accepted in any case (i.e., “random walk”). Between these extremes, many intermediate acceptance criteria can be found in prior literature, such as threshold-based or probabilistic ones (simulated annealing). Preliminary results have shown that in our case, the “better” criterion outperforms all other mentioned approaches.

5.7 Combined GA and ILS

Combinations of genetic algorithms and local search-based algorithms, so-called memetic algorithms or genetic local search metaheuristics, have proven to be beneficial in the literature (see, e.g., Neri et al. (2012), who summarized work on memetic algorithms or Vela et al. (2010), who successfully interleaved a GA with a local search and called this hybrid form “genetic local search”). Because the power of ILS in our case highly depends on the quality of the initial solution, we investigated further into hybrids of the two presented approaches. However, classic memetic approaches that perform a local search on children in the GA were not competitive. Therefore, we present a simple but effective (see Sect. 6) combination of the two described stand-alone metaheuristics in a way that we first run the GA and afterward, taking the best solution of the GA, we perform the proposed ILS approach. The available CPU time is distributed equally among the approaches.

6 Results

We implemented and tested the solution approaches with a set of randomly generated problem instances of varying sizes. After a brief description of the instances and environment used for our computational study (Sect. 6.1), we give insight into preliminary tests conducted in order to measure the impact of the neighborhoods listed in Sect. 5.6. Then, we proceed with the analysis of small, toy instances to assess the performance of the metaheuristic solution approaches in comparison with the exact MIP formulation in Sect. 6.3. Finally, we discuss the computational results for large and medium-sized problem instances in Sect. 6.4 and compare the algorithms on various key figures associated with the solutions.

6.1 Experimental setup

The instances were generated using the knowledge of MedAustron staff about the underlying distribution functions. Patient-specific random variables came from the distributions summarized in Table 4. Small toy examples with 3–25 patients are created to assess the quality of the heuris-

Table 4 Probability distributions of instance specifics

Attribute	Probabilities
Beam type	$P(\text{BeamType}_p = P) = 0.5$ $P(\text{BeamType}_p = C) = 0.5$
Duration irradiation $\lambda_{pi} \quad \forall i \in \Phi_p$	$\sim \mathcal{N}(12, 5) \iff \text{BeamType}_p = P$ $\sim \mathcal{N}(8, 5) \iff \text{BeamType}_p = C$
Room	$P(\text{Room} = 1) = 0.33$ $P(\text{Room} = 2) = 0.33$ $P(\text{Room} = 3) = 0.33$
Duration in-room setup	$P(v_{pir} = 12) = 0.8$ $P(v_{pir} = 22) = 0.2$
Duration in-room teardown	$P(w_{pir} = 3) = 0.7$ $P(w_{pir} = 6) = 0.3$
aRO	$P(\text{aRO} = 1) = P(\text{aRO} = 2) =$ $P(\text{aRO} = 3) = P(\text{aRO} = 4) = 0.25$
PET	$P(\text{PET} = \text{true}) = 0.5$

tic methods in comparison with exact solution methods; the larger, more realistic instances include 35–175 patients.

Additional instance settings are as follows:

$$u_{piqjr} = \begin{cases} 3, & \text{if } p \text{ and } q \text{ have different beam} \\ & \text{types and } r \text{ is the beam resource,} \\ 0, & \text{otherwise.} \end{cases}$$

$$\lambda_{pi} = \begin{cases} 30, & \text{if } i \in \Xi_p, \\ 10, & \text{if } i \in \Psi_p. \end{cases}$$

$$\alpha' = 30 \text{ min}$$

$$\alpha'' = 120 \text{ min}$$

$$FS_{pi,p(i+\delta)}^{\min} = \begin{cases} 0, & \text{if } i \in \Phi_p \wedge (i + \delta) \in \Xi_p, \\ 15, & \text{if } i \in \Phi_p \wedge (i + \delta) \in \Psi_p, \\ 15, & \text{if } i \in \Xi_p \wedge (i + \delta) \in \Psi_p. \end{cases}$$

$$FS_{pi,p(i+\delta)}^{\max} = \begin{cases} 15, & \text{if } i \in \Phi_p \wedge (i + \delta) \in \Xi_p, \\ 60, & \text{if } i \in \Phi_p \wedge (i + \delta) \in \Psi_p, \\ 60, & \text{if } i \in \Xi_p \wedge (i + \delta) \in \Psi_p. \end{cases}$$

For instances with $P \geq 35$, we acknowledge that several patients probably have already started their treatment, prior to the beginning of the planning horizon. According to our project partner, patients require an average of 20 treatments, resulting in the following scheme: For any day in the planning horizon, approximately 1/4 of patients will be having their first week of treatments; another quarter is within the second treatment week. Finally, 1/4 of patients has already had two treatment weeks and is currently in the third week of treatments and the last quarter of patients is finishing treatment after the current week. We follow this systematic approach when generating our real-world inspired problem instance.

Table 5 DT Specifics of Larger Instances

Cat.	w_{FDT}	N_p^{DT}	r_p	d_p
1	–	$\sim \mathcal{U}(4, 5)$	0	$P(d_p = 0) = 0.8$ $P(d_p = 1) = 0.2$
2	–	$\sim \mathcal{U}(8, 10)$	0	
3	–	$\sim \mathcal{U}(12, 15)$	0	
4	0	$\sim \mathcal{U}(16, 20)$	0	1
5	1	$\sim \mathcal{U}(12, 15)$	5	6
6	2	$\sim \mathcal{U}(8, 10)$	10	11
7	3	$\sim \mathcal{U}(4, 5)$	15	16

We therefore create seven types of patients, as in Table 5. The first three categories have already started their treatment prior to the beginning of the planning horizon and have between 4 and 15 treatments left. Therefore, the release date for these patients is 0. For 20% of these patients, we assume that they could have a day off on day 0, resulting in a due date of 1 instead of 0. In categories 4–7, patients start their treatment in weeks 0–3, respectively. Because we plan a total of 4 weeks, we assign only a limited number of DTs to these patients. The release and due dates of these patients equal Monday to Tuesday of the given starting week w_{FDT} .

Instances with $P \in \{9, 15, 25\}$ follow a similar pattern but use only 3, 5, and 5 categories and planning horizons of 10, 15, and 15 days, respectively. Patients in instances with $P \in \{3, 4, 5\}$ all have equal release dates (day 0) and due dates (day 1) and every patient needs the same number of DTs. The total number of DTs for these instances is given in Table 7.

The calculation of tight lower bounds is hardly possible for the RPSP, so we manipulated the activity durations of the DTs of randomly generated instances so that there exists an (optimal) solution to the problem for which no unavoidable idle time of the beam resource is necessary. These optimal solutions also do not contain time window violations. The objective value then consists of the total durations of the DTs, that is, the minimum time the beam is used in total. For non-manipulated instances (see Table 9), we use the same measure as a lower bound to the problem.

All algorithms have been implemented in C++. The MIP was solved using Gurobi 7.0.2. The experiments have been carried out on the Vienna Scientific Cluster (VSC3), whose compute nodes are equipped with two Intel Xeon E5-2650v2, 2.6 Ghz, 8 core CPUs each.

6.2 Experimental evaluation of VND neighborhoods

We conducted preliminary experiments in order to assess the impact of all six neighborhoods used within the VND part of the ILS. Therefore, we formed multiple subsets of the $k_d = 6$ neighborhoods and performed randomized com-

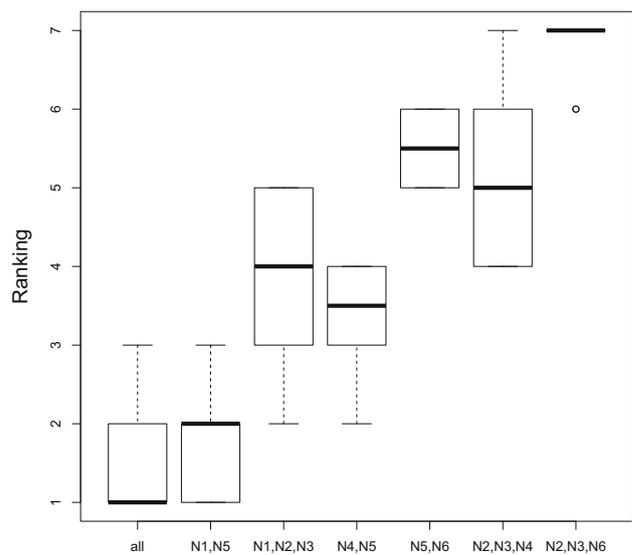


Fig. 8 Boxplot of seven subsets of neighborhoods containing ranked results averages among 16 replications on 20 different instance-neighborhood size combinations

putational tests on 10 problem instances including 35–175 patients. Five instances are randomly generated, while the remaining five are again “manipulated” instances with known lower bound. Seven subsets have been chosen such that each subset still operates on all three parts of the solution representation. These subsets are:

1. All $k_d = 6$ neighborhoods N1, N2, N3, N4, N5, and N6
2. Neighborhoods N1, N2, and N3
3. Neighborhoods N1, and N5
4. Neighborhoods N4, and N5
5. Neighborhoods N5, and N6
6. Neighborhoods N2, N3, N4
7. Neighborhoods N2, N3, N6

On all 10 instances and two neighborhood sizes (small, “SN” and large, “LN”), 16 replications of the ILS were run, equaling 20×16 runs for each of the neighborhood subsets mentioned above. Then, concerning the average results obtained by the 16 replications, the rank of the neighborhood subset was calculated for each of the 10 problem instances and the two neighborhood sizes. The best neighborhood subset obtained rank 1, while the worst one obtained rank 7 (see e.g., Hemmelmayr et al. 2012 for a similar comparison of neighborhood subsets). Figure 8 depicts the results of the ranking for all neighborhood subsets in form of box plots. The results show that for some of the tested instances, a subset of the six neighborhoods, more specifically the subset containing N1 and N5, performs best and is therefore ranked number 1. However, averaged over all instances (see the bold line in the boxes representing the median rank), the proposed

Table 6 Success rates of each neighborhood during the optimization of five instances and two algorithmic settings (small, large neighborhood) each

P	SN/LN	N1 (%)	N2 (%)	N3 (%)	N4 (%)	N5 (%)	N6 (%)
35	SN	44	31	18	17	49	32
35	LN	44	31	18	17	50	32
70	SN	50	32	21	19	57	45
70	LN	52	32	21	20	59	43
105	SN	50	40	28	18	69	57
105	LN	53	40	29	19	73	57
140	SN	52	40	35	19	77	64
140	LN	56	38	36	21	81	65
175	SN	53	36	45	20	84	73
175	LN	58	34	48	23	86	74

Table 7 Results of small instances

P	N^{DT}	Opt.	MIP								Heuristics		
			Full	Time-to (s)	Vars	Cons	Reduced	Time-to (s)	Vars	Cons	GA	ILS	Time
<i>Small instances</i>													
3	12	184	184*	4/33	580	850	184*	2/13	358	508	184	184	60
4	16	240	240*	82/5797	923	1276	240*	1/122	623	812	240	240	60
5	20	248	248^	7609	1226	1647	248^	4369	876	1078	248	248	60
3	27	414	414^	3326	2596	3818	414^	1184	1359	1411	414	414	60
5	30	372	372^	22,432	3026	4416	376	1518	1745	1704	372	372	60
4	36	540	540^	79,853	4609	6862	540^	54,769	4148	6507	540	540	60
5	45	558	591	3173	6743	9736	563	26,121	5567	7836	558	558	60
9	60	720	759	84,633	13,250	20,364	–	86,400	11,201	17,336	720	720	1800
15	135	1410	–	86,400	57,409	72,971	–	86,400	46,785	59,587	1414	1410	3600
25	225	2295	–	86,400	156,304	184,960	–	86,400	120,699	140,088	2353	2296	5400

*Gurobi has found the optimal solution, and optimality was proven. ^Gurobi has found the optimal solution, but optimality could NOT be proven
Best found solutions of over all solution methods are in bold

version of the algorithm containing the full set of neighborhoods, is clearly better than all the remaining subsets.

Additionally, we analyzed the success rate of each neighborhood during the optimization by dividing the number of times the neighborhood lead to a better solution through the number of times the neighborhood was called during the optimization, i.e., $n_{\text{success}}/n_{\text{called}}$. The corresponding results are summarized in Table 6. All six neighborhoods have considerable success rates, indicating once more that all six neighborhoods contribute to the search of the best solution. Furthermore, the success rate of each neighborhood apparently increases with the instance size. Hence, larger instances seem to profit even more from the whole range of neighborhoods, which is why we have chosen to include all six neighborhoods in the more extensive experimental tests of small and large instances.

6.3 Small instances

The initial tests entail small instances with known optimal solution (see Table 7, column “opt.”). We compare results from two versions of the linearized variant of the MIP model presented in Sect. 4: The full model contains all variables and constraints described in Sect. 4, whereas in the reduced model, optional activities are assigned prior to the optimization, thereby radically decreasing the number of variables and constraints in the model (see Table 7, columns “vars” and “cons,” respectively). The “time-to” column indicates the running time needed to achieve the best found solution (and to prove the optimality of this solution), with a maximum running time for both MIP versions of 24 h. Furthermore, the two basic versions of the GA and ILS (large neighborhood) are listed. The results show that the optimal solution was found almost for all problem instances in a reasonable time span

Table 8 Results of large, manipulated instances (one instance per P with 16 replications each)

P	Av. N^{DT}	Opt.	Time limit	Initial		GA		ILS			cGAILS	
				Average	Gap (%)	Average	Gap (%)	Av. SN	Av. LN	Gap (%)	Average	Gap (%)
<i>Large instances</i>												
35	400	3920.0	7200	5143.0	31.2	4153.9	6.0	4065.3	4068.2	3.7	4097.8	4.5
70	800	7580.0	14,400	10,515.8	38.7	8143.6	7.4	8209.3	8196.4	8.1	8163.3	7.7
105	1200	11,240.0	21,600	14,463.8	28.7	12,279.2	9.2	12,507.1	12,457.9	10.8	12,252.0	9.0
140	1600	14,900.0	28,800	20,309.6	36.3	16,426.7	10.2	16,792.3	16,630.8	11.6	16,490.1	10.7
175	2000	18,560.0	36,000	27,057.9	45.8	20,631.7	11.2	21,104.4	20,956.6	12.9	20,686.4	11.5

Best found solutions of over all solution methods are in bold

Table 9 Results of large, real-world-inspired instances (16 instances per P with 16 replications each)

P	Av. N^{DT}	LB	Time limit	Initial		GA		ILS			cGAILS	
				Average	Gap	Average	Gap (%)	Av. SN	Av. LN	Gap (%)	Average	Gap (%)
<i>Large instances</i>												
35	400	3717.6	7200	5145.7	38.4	4544.2	22.2	4476.2	4473.3	20.3	4459.6	20.0
70	800	7107.4	14,400	10,522.0	48.0	8638.9	21.5	8577.7	8572.0	20.6	8520.6	19.9
105	1200	10,603.0	21,600	14,442.9	36.1	13,070.1	23.3	12,995.2	12,997.9	22.6	12,820.4	20.9
140	1600	13,976.7	28,800	20,315.3	45.4	17,212.8	23.2	17,267.1	17,369.9	23.5	16,930.7	21.1
175	2000	17,734.8	36,000	27,016.5	52.3	22,205.6	25.2	22,288.7	22,422.1	25.7	21,946.5	23.7

Best found solutions of over all solution methods are in bold

by the GA and the ILS, with a small advantage for the latter method. The smaller instances with $N^{\text{DT}} = \sum_p N_p^{\text{DT}} \leq 36$ also could be solved to optimality by Gurobi (marked by $\hat{\cdot}$). However, Gurobi was only capable of proving optimality for the two smallest instances (marked with *). Furthermore, the running time to find (the optimal) solutions is already vast for the small instances; those with 45 or more DTs likely could not be solved to optimality by Gurobi. For instances with 15 and 25 patients, not even a single feasible solution was found after 24 h of running time. We therefore conclude that solving the RPSP to optimality using mathematical programming techniques is beyond the scope for larger problem instances.

6.4 Medium and large instances

Tables 8 and 9 summarize the results of computational tests performed using larger problem instances that include 35–175 patients. The combination of the day assignment and the in-room setup and teardown times complicate the calculation of a strict lower bound, because it makes assessing *unavoidable* idle time on the beam resource virtually impossible. Therefore, we adopt two strategies to evaluate the quality of the solutions. Table 8 lists five instances in which we

manually manipulated the proportion of time used for setup, teardown, and treatment so that there exists an optimal solution without idle time on the beam. The calculated gaps for these manipulated instances can be interpreted as optimality gaps. Then, Table 9 contains averages over 16 randomly generated instances. Here, we compare the solution to a naive lower bound consisting of the total time the beam is required within the planning horizon, equal to the sum of all irradiation durations of all patients.

The “Initial” column lists the objective value of the best solution among the initial GA population, which also serves as a starting solution to the ILS. The “average” column reveals the average of best found solution fitnesses after the time limit has been reached for all proposed methods. Note that we limited the number of evaluated neighbors within the local search of the ILS. We investigated two neighborhood sizes: The first, “small” approach evaluates \sqrt{N} neighbors per iteration, with N as the total number of potential activities to be scheduled, which we abbreviate to “SN.” The second approach allows us to examine P neighbors of the current solution (denoted “LN”).

The results show a slight advantage of the GA and cGAILS for the manipulated instances in Table 8. The corresponding gaps from the optimal values increase slightly with the

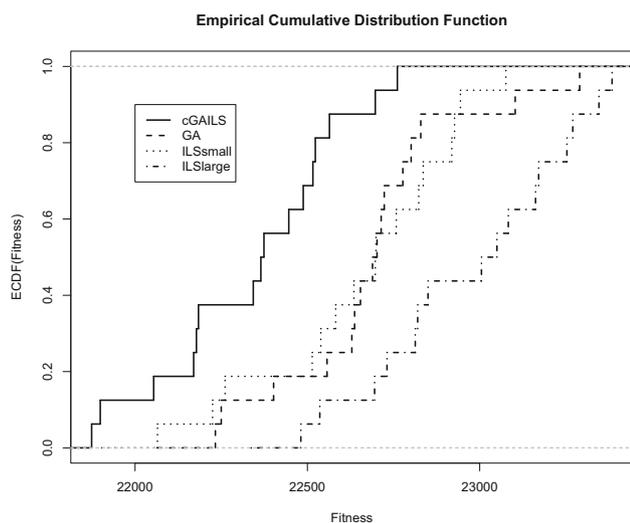


Fig. 9 Empirical cumulative distribution functions: 175 patients and 16 replications

instance size and the underlying problem complexity for these manipulated instances.

The randomly generated instances in Table 9 clearly show the benefits of combining GA and ILS, which for all problem sizes delivers the best solutions. For the real-world, non-manipulated instances, the gap from the lower bound remains more or less stable, even with increasing problem size. The dominance of the hybrid method over the stand-alone methods, for all $5 \times 16 = 80$ real-world-inspired instances, can be verified statistically using both t tests and Wilcoxon Rank tests. Specifically, these tests yield significantly better results for almost all instances: cGAILS produces significantly superior results to GA in 78 of the 80 instances. Furthermore, the combined method performs significantly better than ILS with small neighborhood in 61 and better than ILS with large neighborhood in 62 cases (tables of the results of all statistical tests can be found in “Appendix”).

Figure 9 compares the empirical cumulative distribution functions (ECDF) of the four methods: GA, ILS with small neighborhood, ILS with large neighborhood, and cGAILS for one specific instance including 175 patients. For these functions, all 16 replications per solution method were sorted according to their objective value. The graph displays the percentage of solutions (y-axis) among these 16 replications per method that lie below a given objective value (x-axis), i.e., the midpoint of the y-axis gives the median performance of the algorithms, whereas the upper line gives the worst case and the lower line the best case performances. According to this analysis, cGAILS performs better for all percentiles of the ECDF. Although GA and ILS-small depict comparable median performance, both the best and the worst solutions of the GA are weaker than the best/worst of the ILS with small neighborhood. Furthermore, ILS with large neighborhood

Table 10 Key figures of 175-patient instances

Method	Av. Fit.	Av. Pen.	Av. holes	Av. pS	Av. sr2	Av. sr1b
GA	22,205.6	28.1	44.9	134.4	33.4	509.2
ILS-small	22,288.7	37.4	48.3	305.5	27.8	564.9
cGAILS	21,946.5	26.0	54.3	179.7	25.6	535.2

does not yield comparable results for this specific instance, which is also evident for instances with 140–175 patients in Table 9.

Table 10 gives an overview of some attributes of the achieved solutions by the three metaheuristic approaches for the 175 patients instances (last row in Table 9). The “av. Fit.” column denotes the average fitness of the solutions, with “av. Pen.” indicating the penalty term of the fitnesses. The “av. holes” column lists the total number of unassigned days over all patients, and “av. pS” sums the particle switches that cause sequence-dependent setup on the beam resource. Furthermore, “av. sr2” indicates how often the same room is required for a DT consecutively, leading to immediate idle time on the beam during the in-room setup and teardown times. Finally, “av. sr1b” counts cases in which it was not possible to iterate through all rooms, so only a switch between two rooms occurred (e.g., room 1, room 2, and again room 1, with only one patient treated in a different room between the two times in the same room). We aim to minimize key figures av. pS to av. sr1b to achieve solutions with less idle time. A low number of unassigned days (“holes” within a patient’s treatment plan) do not necessarily lead to better solutions; instead, we anticipate some “optimal” number of unassigned days for each instance.

Comparing the three methods according to the key figures gives an initial impression of the advantages and drawbacks of each method. The average GA solution includes only small time window penalties and particle switches, but the same room is more often used consecutively, leading to higher idle time on the beam. In contrast, ILS results in a higher number of particle switches, simultaneously reducing av. sr2 to only 27.8. Then, cGAILS can further shorten this number to 25.6, at the same time accepting somewhat more particle switches than the GA solutions. However, particle switches cause beam idle time of 3 min, whereas using the same room twice consecutively, on average, leads to an idle time of 17.9 min. Therefore, if idle time is unavoidable, we would rather switch beam types than use the same room twice in a row. Finally, the average number of holes within the best found solution is significantly higher than in GA and ILS. Accordingly, we conclude that the optimal number of unassigned days over all patients lies between 50 and 60 for the 175-patient instances.

7 Conclusion

Scheduling recurring radiotherapy treatment appointments in ion beam facilities in which multiple rooms share one particle beam represents a complex job shop scheduling problem with custom constraints. We have introduced the specific problem setting and formulated the problem mathematically. However, in realizing that solving the MIP for real-world instances is intractable, we developed a GA and an ILS approach, both of which build on a multi-encoded solution representation and a chronological decoding algorithm. The GA contains tailor-made, feasibility-preserving, crossover operators, and an offspring selection strategy. The local search within the ILS is composed of a VND that operates on six different neighborhoods of the incumbent solution. We have shown that the two stand-alone meta-heuristic approaches lead to excellent solutions for small problem instances, even highly dominating the MIP approach with regard to running times and solution quality. For larger instances, the combination of the population-based GA and the individual-based ILS leads to significantly better results than each of the approaches achieves individually, yielding solutions that perform soundly on all measured key figures.

Future research on radiotherapy scheduling will be dedicated to stochastic optimization techniques; many parameters of the underlying problem are subject to intense uncertainty. For example, some patients might not be capable of getting the planned irradiation treatment if their condition becomes very severe. Sometimes the treatment start needs to be postponed for patients to recover from short-term illnesses. Such

requirements might be assessed by either robust scheduling techniques on a long-term basis and/or intelligent intra-day optimization approaches that handle disruptions to the planned schedule on the fly.

Acknowledgements Open access funding provided by University of Vienna. We thank EBG MedAustron GmbH for giving us insight into the problem of scheduling recurring radiotherapy treatment appointments and for its fruitful collaboration.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

Appendix: Statistical tests

To prove the superiority of the hybrid method cGAILS relative to the stand-alone methods GA, ILS with small neighborhood, and ILS with larger neighborhood, we conducted various statistical tests, the results of which are presented in the following Tables 11 and 12. We performed both *t* tests (two-sided) and Wilcoxon Rank tests and obtained comparable results, as both tests support the hypothesis that the hybrid method achieves significantly better outcomes. ***, **, and * denote *p* values ≤ 0.001 , ≤ 0.01 , and ≤ 0.05 , respectively, whereas “ns” indicates cases with *p* value > 0.05 . For each of the $5 \times 16 = 80$ instances, 16 replications were run.

Table 11 Results of *t* tests of 80 instances, batched by number of patients *P* per instance

<i>P</i>	cGAILS versus GA				cGAILS versus ILS-SN				cGAILS versus ILS-LN			
	***	**	*	ns	***	**	*	ns	***	**	*	ns
<i>t</i> test (two-sided)												
35	16	16	16	0	2	5	6	10	1	2	5	11
70	16	16	16	0	4	9	13	3	4	10	12	4
105	16	16	16	0	12	14	15	1	12	12	14	2
140	15	16	16	0	15	16	16	0	16	16	16	0
175	4	11	14	2	9	11	11	5	12	14	15	1
\sum_P	67	75	78	2	42	55	61	19	45	54	62	18

Table 12 Results of Wilcoxon rank tests of 80 instances, batched by number of patients *P* per instance

<i>P</i>	cGAILS versus GA				cGAILS versus ILS-SN				cGAILS versus ILS-LN			
	***	**	*	ns	***	**	*	ns	***	**	*	ns
<i>Wilcoxon Rank test</i>												
35	16	16	16	0	2	4	6	10	0	2	4	12
70	15	16	16	0	3	8	12	4	4	8	12	4
105	16	16	16	0	9	13	16	0	12	12	15	1
140	13	16	16	0	14	16	16	0	16	16	16	0
175	4	8	15	1	9	10	11	5	12	14	15	1
\sum_P	64	72	79	1	37	51	61	19	44	52	62	18

References

- Affenzeller, M., & Wagner, S. (2004). SASEGASA: A new generic parallel evolutionary algorithm for achieving highest quality results. *Journal of Heuristics*, 10(3), 243–267.
- Bray, F., Jemal, A., Grey, N., Ferlay, J., & Forman, D. (2012). Global cancer transitions according to the Human Development Index (2008–2030): A population-based study. *The Lancet Oncology*, 13(8), 790–801.
- Burke, E. K., Leite-Rocha, P., & Petrovic, S. (2011). An integer linear programming model for the radiotherapy treatment scheduling problem. CoRR abs/1103.3, arXiv:1103.3391v1
- Castro, E., & Petrovic, S. (2012). Combined mathematical programming and heuristics for a radiotherapy pre-treatment scheduling problem. *Journal of Scheduling*, 15, 333–346.
- Conforti, D., Guerriero, F., & Guido, R. (2008). Optimization models for radiotherapy patient scheduling. *4OR*, 6, 263–278.
- Conforti, D., Guerriero, F., & Guido, R. (2010). Non-block scheduling with priority for radiotherapy treatments. *European Journal of Operational Research*, 201, 289–296.
- Conforti, D., Guerriero, F., Guido, R., & Veltri, M. (2011). An optimal decision-making approach for the management of radiotherapy patients. *OR Spectrum*, 33(1), 123–148. <https://doi.org/10.1007/s00291-009-0170-y>.
- Gocgun, Y. (2016). Simulation-based approximate policy iteration for dynamic resource-constrained project scheduling. *Health Care Management Science*. <https://doi.org/10.1007/s10729-016-9388-9> (to appear).
- Gupta, D., & Denton, B. (2008). Appointment scheduling in health care: Challenges and opportunities. *IIE Transactions*, 40, 800–819.
- Hemmelmayr, V., Schmid, V., & Blum, C. (2012). Variable neighbourhood search for the variable sized bin packing problem. *Computers and Operations Research*, 39(5), 1097–1108. <https://doi.org/10.1016/j.cor.2011.07.003>.
- Kapamara, T., & Petrovic, D. (2009) A heuristics and steepest hill climbing method to scheduling radiotherapy patients. In *Proceedings of the international conference on operational research applied to health services (ORAHs)*, Catholic University of Leuven, Leuven, Belgium
- Kapamara, T., Sheibani, K., Haas, O., Petrovic, D., & Reeves, C. (2008). A review of scheduling problems in radiotherapy. In *Proceedings of the international control systems engineering conference (ICSE)* (pp. 207–211)
- Legrain, A., Fortin, M. A., Lahrichi, N., & Rousseau, L. M. (2015). Online stochastic optimization of radiotherapy patient scheduling. *Health Care Management Science*, 18, 110–123.
- Leite-Rocha, P. (2011). Novel approaches to radiotherapy treatment scheduling. Ph.D. thesis, University of Nottingham
- Lourenço, H. R., Martin, O. C., & Stützle, T. (2010). Iterated local search. In F. Glover & G. A. Kochenberger (Eds.), *Handbook of metaheuristics* (pp. 363–397). Norwell: Kluwer Academic Publishers.
- Maschler, J., Riedler, M., Stock, M., & Raidl, G. R. (2016). Particle therapy patient scheduling: First heuristic approaches. In: E. K. Burke (Ed.), *PATAT2016: Proceedings of the 11th international conference of the practice and theory of automated timetabling* (pp. 223–244)
- Maschler, J., Hackl, T., Riedler, M., & Raidl, G. R. (2017). An enhanced iterated greedy metaheuristic for the particle therapy patient scheduling problem. In: A. Duarte, A. Viana, J. Angel, B. Mélian, & H. Ramalhinho (Eds.), *Proceedings of the MIC and MAEB 2017 conferences*, Barcelona (pp. 465–474)
- Maschler, J., Riedler, M., & Raidl, G. R. (2018). Particle therapy patient scheduling: Time estimation for scheduling sets of treatments. In R. Moreno-Díaz, F. Pichler, & A. Quesada-Arencibia (Eds.), *Computer aided systems theory—EUROCAST 2017* (pp. 364–372). Berlin: Springer International Publishing.
- Men, C. (2009). Optimization models for radiation therapy: Treatment planning and patient scheduling. Ph.D. thesis, University of Florida
- Neri, F., Cotta, C., & Moscato, P. (Eds.). (2012). *Handbook of memetic algorithms*. Berlin: Springer. <https://doi.org/10.1007/978-3-642-23247-3>.
- Ohno, T. (2013). Particle radiotherapy with carbon ion beams. *The EPMA Journal*. <https://doi.org/10.1186/1878-5085-4-9>.
- Petrovic, D., Morshed, M., & Petrovic, S. (2009). Genetic algorithm based scheduling of radiotherapy treatments for cancer patients. *Proceedings of the Conference on Artificial Intelligence in Medicine (AIME)*, 5651, 101–105.
- Petrovic, D., Morshed, M., & Petrovic, S. (2011). Multi-objective genetic algorithms for scheduling of radiotherapy treatments for categorised cancer patients. *Expert Systems with Applications*, 38(6), 6994–7002.
- Petrovic, S., & Castro, E. (2011). A genetic algorithm for radiotherapy pre-treatment scheduling. In *Lecture notes in computer science (including subseries Lecture notes in artificial intelligence and lecture notes in bioinformatics)*, LNCS(PART 2) (Vol. 6625, pp. 454–463).
- Petrovic, S., & Leite-Rocha, P. (2008a). Constructive and GRASP approaches to radiotherapy treatment scheduling. In *Proceedings—Advances in electrical and electronics engineering—IAENG special edition of the world congress on engineering and computer science 2008*, WCECS 2008 (pp. 192–200)
- Petrovic, S., & Leite-Rocha, P. (2008b). Constructive approaches to radiotherapy scheduling. In *World Congress on Engineering and Computer Science (WCECS)* (pp. 722–727)
- Petrovic, S., Leung, W., Song, X., & Sundar, S. (2006). Algorithms for radiotherapy treatment booking. In *25th Workshop of the UK planning and scheduling special interest group* (pp. 105–112)
- PTCOG (2017) Particle therapy co-operative group. <https://www.ptcog.ch/index.php/facilities-in-operation>. Accessed 15 Jan 2018.
- Sauré, A., Patrick, J., Tyldesley, S., & Puterman, M. L. (2012). Dynamic multi-appointment patient scheduling for radiation therapy. *European Journal of Operational Research*, 223, 573–584.
- Steward, B. W., & Wild, C. P. (2014). World cancer report 2014. Technical report, World Health Organization, International Agency for Research on Cancer.
- Syswerda, G. (1996). Schedule optimization using genetic algorithms. In L. Davis (Ed.), *Handbook of genetic algorithms* (pp. 332–349). London: International Thomson Publishing Services.
- Vela, C. R., Varela, R., & González, M. A. (2010). Local search and genetic algorithm for the job shop scheduling problem with sequence dependent setup times. *Journal of Heuristics*, 16(2), 139–165. <https://doi.org/10.1007/s10732-008-9094-y>.
- Vieira, B., Hans, E. W., Van Vliet-Vroegindewij, C., Van De Kamer, J., & Van Harten, W. (2016). Operations research for resource planning and -use in radiotherapy: A literature review. *BMC Medical Informatics and Decision Making*, 16(149), 1–11. <https://doi.org/10.1186/s12911-016-0390-4>.
- Vogl, P., Braune, R., & Doerner, K. F. (2018). A multi-encoded genetic algorithm approach to scheduling recurring radiotherapy treatment activities with alternative resources, optional activities, and time window constraints. In: R. Moreno-Díaz, F Pichler, & A. Quesada-Arencibia (Eds.), *Computer aided systems theory—EUROCAST 2017*, Springer International Publishing (pp. 373–382)
- Washington, C. M., & Leaver, D. (2016). *Principles and practice of radiation therapy*. Maryland Heights: Mosby.
- World Health Organization. (2012). *GLOBOCAN 2012*. <http://globocan.iarc.fr/Default.aspx>. Accessed 1 Oct 2017.