Central European Journal of Operations Research (2022) 30:1239–1277
https://doi.org/10.1007/s10100-021-00762-5

Stochastic radiotherapy appointment scheduling

Roland Braune[1] · **Walter J. Gutjahr**[2] · **Petra Vogl**[1]

Accepted: 27 June 2021 / Published online: 11 July 2021

Abstract
When scheduling the starting times for treatment appointments of patients in hospitals or outpatient clinics such as radiotherapy centers, minimizing patient waiting time and simultaneously maximizing resource usage is crucial. Significant uncertainty in the treatment durations makes scheduling those activities particularly challenging. In addition to the treatments themselves, also preparation times and exiting times have to be considered, which are uncertain as well. To address and analyze this type of problems, the current study develops a model for planning appointment times under uncertain activity durations for a medical unit with a single "core resource" (in our application case a radiotherapy beam device), several treatment rooms, and required preparation and exiting phases for each patient. We employ a novel buffer concept based on quantiles of duration distributions and introduce a reactive procedure that adapts a pre-determined baseline schedule to the actual patient flow. For heuristically solving the resulting stochastic optimization model, a combination of a Genetic Algorithm and Monte Carlo simulation is proposed. A case study uses real-world data on activity durations gathered from an ion beam therapy facility in Austria. Experimental results comparing different variants of the method are carried out. In particular, comparisons of the stochastic optimization approach to a simpler deterministic approach are given.

Keywords Appointment scheduling · Radiotherapy · Stochastic project scheduling · Monte Carlo simulation · Genetic algorithm

✉ Roland Braune
roland.braune@univie.ac.at

[1] Department of Business Decisions and Analytics, University of Vienna, Oskar-Morgenstern-Platz 1, 1090 Vienna, Austria

[2] Department of Statistics and Operations Research, University of Vienna, Oskar-Morgenstern-Platz 1, 1090 Vienna, Austria

1 Introduction

The "global cancer burden" is projected to exceed 27 million new cancer cases per year by 2040, which exceeds the estimated figures of 2018 (the most recent estimate of 18.1 million cases) by a factor of 1.5, as indicated in the 2020 world cancer report (Wild et al. 2020). This immense rise together with the general pressure to rationalize health care expenses requires health care facilities such as radiotherapy centers to manage machines and resources more efficiently (Tancrez et al. 2013). Radiation therapy, or short radiotherapy, is a commonly used treatment for patients diagnosed with cancer (in addition to or instead of surgery and chemotherapy) with the goal of killing tumorous cells while sparing surrounding healthy tissue. In general, radiotherapy treatment appointments are planned a few days or weeks in advance and emergency patients who need to receive treatment immediately are rare. Nevertheless, real-world data uncovers high uncertainty in treatment durations for radiotherapy appointments, even though medical physicists are able to accurately estimate the planned irradiation duration during the intense treatment planning process. Uncertainty is a key challenge in any appointment scheduling process (Gupta and Denton 2008), but the underlying uncertainty in radiotherapy treatment durations has not yet been considered in the radiotherapy appointment scheduling literature (see Sect. 2).

As has been shown in previous studies (Kreitz et al. 2016, e.g.), waiting time is one of the main factors influencing patient satisfaction. Waiting time to the first treatment appointment (i.e., the start of the recurring treatment process) has been addressed as a crucial objective in several academic papers on radiotherapy appointment scheduling, but daily waiting time between the planned treatment start and the actual time when treatment is performed has not been addressed so far in this area. Since many patients are treated consecutively in a treatment facility, a delay of one single patient typically affects the starting times of several successive patients and can even cause waiting time for all upcoming patients on a given day. Practitioners tend to focus on the optimization of resource usage solely, due to high fixed costs associated with machines and staff, such that they prefer tight schedules. The variability in the actual treatment durations, however, may then lead to resource conflicts and hence delays, an effect that aggravates the tighter the schedule is planned (Gupta and Denton 2008). This suggests an explicit consideration also of waiting times when developing a planning method.

Thus, in order to enhance patient satisfaction and wellbeing, we propose an optimization model that takes both patient waiting time and resource usage into account and increases robustness during the execution of the schedule by considering the stochasticity of the activity durations. We show that for the highly constrained and stochastic problem of radiotherapy appointment scheduling in special ion beam facilities where only one beam resource is available, it is beneficial to insert activity time buffers. Our proposed buffer concept is based on quantiles of probability distributions and can thus be applied to any distribution of treatment durations.

The decision to be optimized in our approach consists of three components: we determine a treatment plan specifying the days on which each patient should receive treatments, a priority list of patients for determining the daily schedules, and a value for the buffer parameter which controls the size of the time buffers. From these three components, we compute a baseline schedule including buffer times. During actual

execution of the baseline schedule, modifications of starting times may become necessary in view of the stochastic nature of durations. To gradually compute the modified schedule from the baseline schedule while the true activity durations are revealed one after the other in the course of a current day, we use a dynamic procedure which we call "reactive procedure". The treatment plan, the patient priority list and the buffer parameter are (heuristically) optimized by a Genetic Algorithm (GA) in the planning phase based on the current data on patients, their treatment requirements and their preferred time windows for treatment. As the objective function in our optimization problem is an expectation of a complex random expression, we need a way to evaluate its value and use the Sample Average Approximation (SAA) technique for this purpose. This essentially results in applying the principle of Monte Carlo simulation.

This article is organized as follows: Sect. 2 reviews related work on radiotherapy scheduling, stochastic appointment scheduling, and general strategies to deal with uncertainty. Section 3 presents a formulation of a general optimization problem of which the stochastic radiotherapy appointment scheduling problem addressed in our application is a special case. Section 4 is dedicated to the proposed solution methodology: we specify the structure of the overall optimization approach, describe the buffer concept, present the reactive procedure, and explain the GA as well as the used SAA technique. In Sect. 5 we analyze data on appointment durations gathered from an ion beam facility in Austria, and fit distributions to the different activity categories. The results of our intensive computational tests are given in Sect. 6. Finally, Sect. 7 concludes and proposes some possible directions for further research.

2 Related work

Radiotherapy Appointment Scheduling The specific problem of scheduling radiotherapy appointments has been addressed in the literature since 2006 (Petrovic et al. 2006), and variants of the problem have been modeled mathematically and solved using heuristics such as greedy randomized adaptive search (Petrovic and Leite-Rocha 2008) or GAs (Petrovic et al. 2009; Petrovic and Castro 2011). Vieira et al. (2016) provide a broad overview of both radiotherapy treatment scheduling and resource planning. Maschler et al. (2017a, b) and Vogl et al. (2018a, b) address specific deterministic, long-term scheduling problems that arise in ion-beam facilities in which multiple rooms are supplied by only one beam resource. They build deterministic baseline schedules by applying different metaheuristic search techniques. However, they completely neglect patient waiting time as a major factor of schedule quality and focus entirely on deterministic resource usage optimization. In the present work, we deal with the same setting, but treat it in a short-term, stochastic environment.

Uncertainty in relation to radiotherapy scheduling is addressed by two papers: Sauré et al. (2012) identify effective policies for allocating demand to still unknown patients using a Markov decision process in an effort to minimize the time that patients must wait before the treatment starts. Legrain et al. (2015) also address uncertainty related to the arrival of patients to radiotherapy facilities and develop a hybrid online stochastic optimization algorithm to tackle the stochasticity. However, to the best of

our knowledge, none of the existing studies on radiotherapy scheduling considers stochastic activity durations, which we investigate in this paper.

Stochastic Appointment Scheduling Problems Stochasticity in appointment scheduling problems can be of a twofold nature: First, the patients to be treated may not be known in advance and instead get successively revealed during the planning horizon. For example, emergency patients need to be immediately treated, and planned patients do not always show up for their appointments. Second, the treatment duration may be subject to uncertainty (Gupta and Denton 2008). We concentrate on this latter aspect. Ahmadi-Javid et al. (2017) review optimization studies that consider random appointment durations (or "service times"). For the special problem of operating room scheduling, Cardoen et al. (2010) consider studies that incorporate uncertain procedure durations. A more recent review by Samudra et al. (2016) also includes a section dedicated to uncertainty.

Belien and Demeulemeester (2004) propose models for building robust cyclic surgery schedules when the procedure duration is stochastic. Robustness also plays an important role in the work by Hans et al. (2008), who use advanced optimization techniques combined with historical data on surgery durations to improve capacity utilization. Denton et al. (2007) conclude that sequencing patients according to the expected variance of their activity duration achieves the best results when the scheduling involves a single server. However, following this strategy might not be beneficial in highly constrained settings. In our case, the preferred time windows would render such a simple strategy ineffective, because a lot of constraint violations would be produced. Kaandorp and Koole (2007) propose a local search procedure to optimize a weighted average of patient waiting time and doctor idle time. They assume the activity duration to be exponentially distributed. Their objective function looks similar to our setting, yet our problem is much more constrained, involving aspects like predefined patient treatment patterns and preferred time windows.

Koeleman and Koole (2012) show that different service time distributions lead to different optimal baseline schedules when a local search algorithm is used and emergency arrivals are included into consideration. Begen et al. (2012) propose a sampling-based approach to address the problem of discrete random appointment durations, which produces a near-optimal solution with high probability in polynomial time. Erdogan and Denton (2013) present two stochastic models considering uncertain durations, as well as no-shows. Tancrez et al. (2013) take uncertainty in operating room planning on a more strategic decision-making level into account and present a Markov process which allows to evaluate the impact of stochasticity on various performance measures. The approach of Kemper et al. (2014) to schedule patients for any convex loss function and any service time distribution entails that customers should be scheduled in non-decreasing order of their scale parameter. Finally, Berg et al. (2014) propose three solution methods to address the two-stage stochastic problem of scheduling patient appointments on a single stochastic server, extending work by Denton et al. (2007).

Methodological Approaches to Address Stochasticity in Scheduling Problems Many approaches deal with uncertainty in project and appointment scheduling, as summarized by Herroelen and Leus Herroelen and Leus (2005) in their overview of literature

pertaining to reactive, stochastic, fuzzy, and proactive scheduling approaches. They mention the stochastic resource-constrained project scheduling problem (SRCPSP) as a paradigmatic problem formulation for scheduling under uncertainty and outline the most usual strategy to tackle with the SRCPSP, namely to apply a dynamic *scheduling policy* that makes decisions at certain points in time, such as the termination of an activity, based on the current state of the project. (We adopt the concept of a scheduling policy in the form of our reactive procedure.) What remains to be done when following this general approach is the optimization of parameters or input variables for the scheduling policy. Considering the stochasticity of essential variables, this leads to a stochastic optimization problem.

Van De Vonder et al. (2005) thoroughly analyze the impact of buffers in project management, and this research group has also considered various combinations of proactive and reactive approaches (Davari and Demeulemeester 2017; Demeulemeester et al. 2008; Van De Vonder et al. 2006, 2007, e.g.).

The classical approaches to stochastic combinatorial optimization problems are based on two- or multi-stage mathematical programming, or on dynamic programming. However, for many real-world problems of this type, these methods are infeasible due to their size and the limited running time. Metaheuristics then offer good alternatives for solving problems marked by uncertainty. A survey of metaheuristics for solving stochastic combinatorial optimization problems (SCOPs) given by Bianchi et al. (2009) lists three possible ways to compute objective functions for SCOPs: (1) if closed-form expressions for expected values are available, compute the objective function exactly; (2) if closed-form expressions are not available or their repeated evaluation is too time consuming, use ad hoc, fast approximations; and (3) if the problem is too complex in terms of dependencies, estimate the objective function by simulation. We consider the latter approach, because we address a highly constrained problem with extensive probabilistic dependencies. This variant also is known as *Sample Average Approximation* and was applied successfully to SCOPs by Kleywegt et al. (2002) as well as Mancilla and Storer (2012). Finally, Juan et al. (2015) review simheuristics—typically an extension of metaheuristics by the integration of simulation in the optimization procedure—and identify multiple subcategories of this term, depending on how much time is spent on simulation and optimization. Their general scheme for solving SCOPs includes a fast simulation process and few replications to estimate the quality of the solution during the optimization (Bianchi et al. 2009).

The contribution of the present paper is threefold: First, we address stochastic activity durations in radiotherapy appointment scheduling (or related applications involving a core resource, preparation and exiting activities, and assigned facilities) by a stochastic optimization approach. We do not only maximize resource usage when building schedules, but additionally strive for minimizing patient waiting time. Secondly, we present a buffer concept relying on quantiles of activity duration distributions. Third, we propose a reactive procedure that mimics the decision process of the human decision maker during the execution of a baseline schedule, and embed it into the stochastic optimization framework. For the numerical solution of the optimization model, we use a Genetic Algorithm, based on objective function evaluations obtained by Sample Average Approximation (and thus Monte Carlo simulation).

3 Problem statement

This section contains a thorough description of the problem at hand. We shall first describe the problem in a more general form to show its fairly broad range of applicability.[1] Then, we will specify the particular features of the model for our case study, the radiotherapy appointment scheduling case described in detail in Sect. 5. The symbols and abbreviations used throughout this article are listed in Tables 8 and 9 in the Appendix.

3.1 The general model

We assume a medical unit consisting of L *treatment rooms* and one *core resource*; the latter can be personnel, a medical device, or whatever. Treatments for a given set of patients are to be scheduled on certain days of a *planning period* consisting of D days in total. The set of patients is denoted by $\mathcal{P}$.

Patient $p \in \mathcal{P}$ needs N_p treatments ($1 \leq N_p \leq D$) during the planing period, each treatment on a different day. For each $p \in \mathcal{P}$, the number N_p is assumed as given in advance. Moreover, we assume that for each patient $p \in \mathcal{P}$, a set of feasible *treatment patterns* is given. A treatment pattern is a vector $x = (x_1, \ldots, x_D)$ with $x_d \in \{0, 1\}$ ($d = 1, \ldots, D$), where $x_d = 1$ if the patient is treated on day d and 0 otherwise. The set of feasible treatment patterns x for $p \in \mathcal{P}$ is denoted by $\mathcal{X}_p \subseteq \{0, 1\}^D$. For each $x \in \mathcal{X}_p$, the binary vector x contains the same number N_p of bits with value 1.

A *treatment plan* is a matrix $X = (x_{pd})_{p \in \mathcal{P}, d=1,\ldots,D}$, defining the treatment pattern for each patient. The treatment plan X is called *feasible* if $x_p \in \mathcal{X}_p$ for each patient $p \in \mathcal{P}$, where $x_p = (x_{p1}, \ldots, x_{pD})$. Let $\mathcal{X}$ denote the set of all feasible treatment plans.

The L treatment rooms are heterogeneous (allowing for different kinds of treatments) and each patient $p \in \mathcal{P}$ is assumed to be assigned already in advance to a specific treatment room, based on medical considerations. This assignment does not change during the period of D days. A basic feature of our model is that we suppose each treatment to consist of three phases: (a) a *preparation phase*, (b) a *core treatment phase* during which the core resource is needed in an exclusive way (it cannot be used for another treatment during this time), and (c) an *exiting phase*. While going through these three phases, the patient does not leave the room.

In addition to the decisions on the days when to treat each patient, *appointment times* for the start of the treatments have to be determined and communicated to the patients. As far as possible, these times should respect the preferences of the patients p which are given by a ready time r_p (earliest appointment time) and a due time τ_p (latest appointment time) for each patient $p \in \mathcal{P}$.

For each patient p, each treatment phase $i = 1, 2, 3$, and each day $d \in D$, the *activity duration* Θ_{pid} describes the time needed for carrying out the corresponding treatment activity (p, i, d). Taking sources of uncertainties discussed in the next subsection into account, we assume that the activity duration Θ_{pid} is not precisely known in advance.

[1] The application range probably even extends beyond health care applications, but we shall keep our terminology oriented to health care for the sake of a better understanding.

Rather than that, we model the variables Θ_{pid} as random variables and assume that their joint distribution $\mathcal{D}$ can be estimated. This is contrary to a simpler version of the model investigated by Vogl et al. (2018b) where deterministic activity durations are assumed. Note that although the realizations of the random variables modeling the activity durations are day-specific, they follow the same day-independent distributions.

As it is typical for the difference between deterministic and stochastic scheduling, completely different solution methods are needed for the stochastic case. The main difficulty of the stochastic case in our situation is that starting times of activities cannot be determined anymore in advance, since they have to depend on the realizations of the random variables Θ_{pid}. As a consequence, a "solution" to the problem cannot be described anymore by a treatment plan plus a static vector of starting times for treatment activities. Rather than that, a solution consists of a scheme which we shall call "design", from which the actual starting times are generated in two steps (one taking place before the planning period, the other during the execution of the treatments). Parts of this design are the treatment plan and a priority list determining in which sequence patients are inserted into the schedule to be constructed. Priority lists can in fact be seen as *priority policies*, as discussed extensively by Möhring and Stork (2000). A priority policy is used at each decision point during scheduling to decide which job (or patient, in our case) to choose from a set of eligible ones. It is commonly defined as a linear ordering of the entities to be scheduled. In the context of the problem at hand, it is simply given as a permutation of patient indices.

Moreover, when determining appointment times, we do not base them on *expected* activity durations, but include buffer times. The sizes of the buffers are a matter of optimization as well. We shall control them by a single parameter β called *buffer parameter*; larger values of β entail larger buffer sizes. The precise way how β influences the buffer sizes can be chosen in a problem-specific way. We propose a special dependence between buffer parameter and buffer sizes for our concrete application below.

From treatment plan, priority list and buffer parameter, a "baseline schedule" is computed. During the execution, actual starting times are generated "on-the-fly" using the baseline schedule and the current realizations of the random activity durations, which turns the baseline schedule into an actual schedule.

Let us now describe the approach outlined above in more formal terms. Formally, a *design* is a triple $Z = (X, \pi, \beta)$, where (1) $X \in \mathcal{X}$ is a feasible treatment plan, (2) the permutation $\pi \in \Pi$ is a *priority list* of patients (Π denotes the set of all permutations of patient indices), and (3) $\beta \in [0, 1]$ is the buffer parameter.

Our general framework requires the specification of two procedures `sched` ("schedule construction") and `react` ("reactive procedure") for the particular application under consideration. We shall specify them in Sects. 4.1 and 4.2 for the case of our radiotherapy application. The general strategy of the approach is the following:

First of all, from a given design $Z = (X, \pi, \beta)$, a *baseline schedule* S_{bas} is derived by the application of the schedule-construction procedure `sched`:

$$S_{bas} = \texttt{sched}(Z) = \texttt{sched}(X, \pi, \beta).$$

This is carried out *before* the beginning of the planning period. The baseline schedule S_{bas} contains "planned" starting times for all activities, where buffer times have already been included. The baseline schedule also defines the appointment times announced to the patients.

In view of the randomness of the variables Θ_{pid}, it is not sure whether S_{bas} can be executed as it is, considering that certain buffer times may be exceeded. This makes the above-mentioned modification during the execution of the schedule necessary. We carry it out by the application of the reactive procedure `react`, which takes S_{bas} and the durations Θ_{pid}, as they are gradually revealed during the planning period, and determines from this input the *actual* schedule $S(\omega)$. Therein, ω stands for the influence of randomness. Thus, with $\Theta = \Theta(\omega)$ representing the collection of the random activity durations Θ_{pid}, the actual schedule results from the design by

$$S(\omega) = \texttt{react}(S_{bas}, \Theta(\omega)) = \texttt{react}(\texttt{sched}(X, \pi, \beta), \Theta(\omega)).$$

Finally, for obtaining an optimization problem, an *objective function* has to be defined. Suppose that a cost function F is given that evaluates each actual schedule S by a cost value $F(S)$. In our context, the evaluation by F includes both the economic usage of the core resource (the overall time span the core resource is needed should be minimized by means of elimination of idle times) and the waiting times of the patients (which should be minimized as well). We shall specify the cost function for our concrete application in precise formal terms below.

Since the realized schedule $S(\omega)$ depends on the influence of randomness, the cost function value $F(S(\omega))$ is a random variable as well. We take its *expected value* $\mathbf{E}[F(S(\omega))]$ as our objective function. In this way, we get an evaluation of any chosen design $Z = (X, \pi, \beta)$ by a function f:

$$f(Z) = \mathbf{E}[F(S(\omega))] = \mathbf{E}[F(\texttt{react}(\texttt{sched}(X, \pi, \beta),\ \Theta(\omega)))]. \tag{1}$$

The aim of our optimization approach is to find a design that will produce minimal expected cost during execution. This produces the following stochastic optimization problem:

$$\min \mathbf{E}\left[F\left(\texttt{react}(\texttt{sched}(X, \pi, \beta),\ \Theta(\omega))\right)\right] \tag{2}$$
$$\text{s.t. } X \in \mathcal{X} \tag{3}$$
$$\pi \in \Pi \tag{4}$$
$$0 \leq \beta \leq 1 \tag{5}$$

Obviously, this is a mixed-integer stochastic optimization problem. Note that although `react` is a dynamic procedure, the problem above is not a dynamic optimization problem anymore since we fix the procedure `react` in advance. Note further that while β appears as a decision variable similar to X and π in formulation (2)–(5), it is in fact a hyperparameter of our proposed approach to solve the minimization problem given by Eq. (2) and is thus determined and fixed *before* the search for an optimal treatment plan and priority list takes place.

3.2 Application: radiotherapy scheduling

In our application study, we deal with appointment scheduling in an ion beam facility, in which one particle beam device serves multiple treatment rooms. This device is our core resource, it represents the main bottleneck of the scheduling problem. In our specific setting, we schedule patients in $L = 3$ treatment rooms, one with a vertical, one with a horizontal and one with a 90 degree flexible beam angle. The particle beam in ion therapy consists of either protons or carbon ions, and switching between those particle types demands a set-up time of a few minutes (in our case about 3 min). The beam is accelerated to two-thirds the speed of light in a linear accelerator, followed by multiple circulations through a synchrotron. As soon as the beam has reached its designated speed, it gets dispatched to one of the available treatment rooms and the patient waiting inside the room is treated. The beam device can only serve one room at a time.

This facility layout is related to the one described by Vogl et al. (2018b), who solve the long-term *deterministic* appointment scheduling problem, striving to maximize overall resource usage. This long-term optimization, however, does not consider uncertainty in appointment durations. Therefore, we go beyond that existing work by the stochastic optimization approach described above. The considerably increased computational complexity of this approach makes it necessary that we do not solve the long-term problem as a whole, but rather decompose it into short-term planning problems for $D = 5$ days each, corresponding to a week from Monday to Friday.

During the five days of our planning horizon, each patient $p \in \mathcal{P}$ must attend a predefined number N_p of irradiation treatments in a predefined treatment room. The constraints defining feasible treatment patterns are based on medical considerations and were provided by the ion beam facility:

(a) Patients *finishing* treatment in the given week have between 2 and 5 irradiation treatments left (the initial treatment plan over the long-term planning horizon forbids a final week with a single left treatment). They must engage in a treatment activity on each day starting on Monday until the number of missing treatments is met. Hence, there is no flexibility in the treatment pattern for these patients.

In our notation, this means that for a patient p finishing treatment in the given week, $\mathcal{X}_p$ is one of the four singleton sets $\{(1, 1, 0, 0, 0)\}$, $\{(1, 1, 1, 0, 0)\}$, $\{(1, 1, 1, 1, 0)\}$ or $\{(1, 1, 1, 1, 1)\}$.

(b) Patients *starting* their treatment in the given week need to receive between 3 and 5 treatments. The idea is to let those patients commence their treatments as late as possible, to ensure a transition to the next week without breaks. Consequently, the first treatment day is Wednesday, Tuesday, or Monday, for 3, 4 and 5 treatments, respectively.

Thus, for such a patient, $\mathcal{X}_p$ is one of the three singleton sets $\{(0, 0, 1, 1, 1)\}$, $\{(0, 1, 1, 1, 1\}$, or $\{(1, 1, 1, 1, 1)\}$.

(c) All other patients have either 4 or 5 treatments scheduled in the current week.

If patient p gets 5 treatments, then $\mathcal{X}_p$ is the singleton $\{(1, 1, 1, 1, 1)\}$. If patient p gets 4 treatments, an earliest weekday $d(p)$ for the treatment break is pre-defined. Hence, for such a patient, $\mathcal{X}_p$ is the set

$$\{(x_1, \ldots, x_5) \in \{0,1\}^5 \mid \sum_{d=1}^{5} x_d = 4 \quad \text{and} \quad x_d = 1 \text{ for } d < d(p)\}.$$

A daily treatment (DT) consists of three sub-activities: in-room preparation of the patient, irradiation, and post-irradiation exiting. All these three sub-activities occupy the predefined treatment room; in addition, the irradiation also blocks the beam for the exclusive use of the current patient p.

There are multiple reasons that might cause disruptions to the daily executed baseline schedule, making the radiotherapy appointment durations highly stochastic: (1) The in-room preparation might take considerably longer than expected if the positioning of the patient (i.e., the verification that the patient lies on the treatment bench correctly) fails and a second attempt is necessary. Furthermore, patients are sometimes less mobile and need additional help when entering the room. (2) If a patient moves during the irradiation or needs a break, the irradiation needs to be either interrupted or even aborted. An interruption leads to an extension of the planned activity duration; aborting the irradiation leads to a shorter than planned activity. (3) Machine error or room unavailability might cause an activity to take longer than expected. For example, a room might need to be cleaned after a patient exit, which delays the start of the next patient's treatment in the same room.

Hence, we consider the actual activity durations of the preparation ($i = 1$), irradiation ($i = 2$) and exiting ($i = 3$) activities for each patient p and each day d as random variables Θ_{pid}, in accordance with the general model introduced in the previous subsection. The distribution $\mathcal{D}$ of these random variables will be estimated in Sect. 5 for our specific application instance from data on patients that have been treated throughout one complete year.

The cost function $F(S)$ is composed of three terms that are weighted by parameters λ_1, λ_2 and λ_3. All three components of the objective function are measured in minutes and need to be minimized. The components are: (i) The actual beam active time for each day d, denoted by φ_d, which is defined by the finishing time of day d's last activity on the beam resource. This component measures the economic use of the beam resource. (ii) The time window violations, denoted by γ_{pd}, which are caused by treatment appointments that are scheduled past their due time (latest time) τ_p on day d and are measured by the amount of excess of the appointment times over the times τ_p. (iii) The actual waiting time of all patients $p \in \mathcal{P}$.

The patient waiting time consists of two parts: (i) pre-preparation waiting time, denoted by δ_{pd}, which reflects any delay in the start of the preparation activity, and (ii) pre-beam waiting time, denoted by ρ_{pd}, which gives the time span between the completion of the in-room preparation activity and the actual start of the irradiation activity.

Thus, in total,

$$F(S) = \lambda_1 \cdot \sum_{d=1}^{D} \varphi_d + \lambda_2 \cdot \sum_{d=1}^{D} \sum_{p=1}^{n} \gamma_{pd} + \lambda_3 \cdot \sum_{d=1}^{D} \sum_{p=1}^{n} (\delta_{pd} + \rho_{pd}), \tag{6}$$

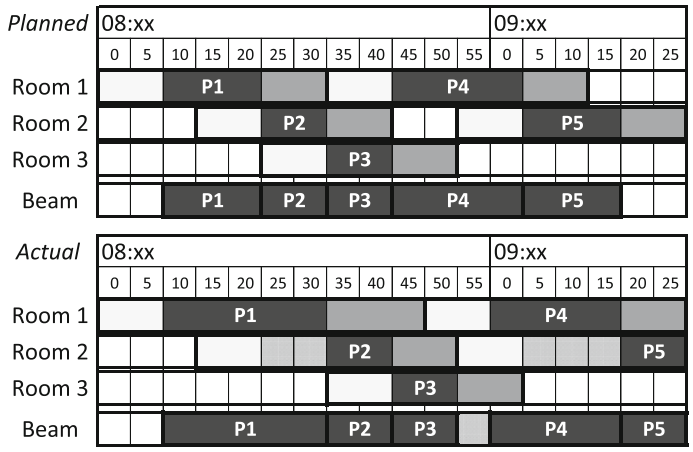

Fig. 1 Exemplary Planned Baseline Schedule and Actually Executed Schedule (white: preparation times; dark gray: irradiation times; medium gray: exiting times; light gray: pre-beam waiting times)

where patients $p \in \mathcal{P}$ have been indexed by $p = 1, \ldots, n = |\mathcal{P}|$. Recall that the cost function is not applied to the baseline schedule S_{bas}, but to the schedule $S = S(\omega)$ produced by the reactive procedure while processing the actual realizations of the random activity durations. In particular, the variables φ_d, δ_{pd} and ρ_{pd} occurring in (6) are random variables, as they depend on the random durations; the overall cost is therefore a random variable as well. Its expected value gives the objective function according to Eq. (1).

The two types of waiting times, resulting from deviations of the actual activity durations from the planned ones, are visualized in Fig. 1. Patient P1's irradiation treatment took longer than expected, which delayed P2's irradiation activity. However, when P2 started the in-room preparation, the delay of P1 was not foreseeable, and P2 had to wait for the start of the irradiation activity. In addition, P1's exit was extended. Therefore, P4 could not enter room 1 on time, leading to a delayed start of P4's preparation activity. Note that P3's preparation started later than expected as well, considering P3 was supposed to start preparation by the time P1 finished the irradiation. The delay of P1's irradiation finish has a direct impact on P3's preparation starting time. Section 4.2 details different reactions to deviations from the baseline schedule.

4 Methodology

Let us now turn to the question how the generic model architecture from Sect. 3 can be treated numerically. For this purpose, five components have to be specified: (1) the concept for computing buffer sizes from the buffer parameter β, (2) the procedure `sched`, computing a baseline schedule S_{bas} from a given design Z, (3) the reactive procedure `react`, determining "on the fly" the actual schedule $S(\omega)$ from the baseline schedule S_{bas} and the current realizations of the activity durations Θ_{pid}, (4) the way how the expected value in the objective function of (2) is evaluated, and (5) the

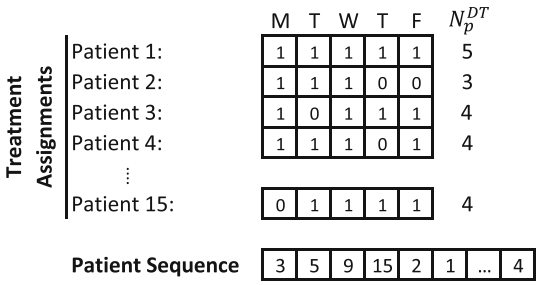

Fig. 2 Treatment plan X and patients priority list π for the radiotherapy application case, $n = 15$ patients

algorithm for performing the optimization (or: heuristic optimization, if the problem cannot be solved to optimality).

From a chronological perspective, the following steps are taken:

1. At the beginning of a week, a design has to be determined that remains fixed throughout that week. The true activity durations are of course not known in advance, but the underlying distributions are given as an input.
2. From day to day, actual decisions on the starting times have to be derived from the design and the actual observations. The realizations of the random variables representing the activity durations are revealed successively over each day and processed by the reactive procedure `react`, turning a baseline schedule into an actual schedule.

4.1 Buffer concept and schedule generation procedure

To create robust baseline schedules, we include time buffers (Van De Vonder et al. 2005) in the planned activity durations. The planned activity duration for activity (p, i, d) including the time buffer, denoted by t_{pid}, is governed by a global buffer parameter $\beta \in [0, 1]$ and is defined as the β-quantile of the marginal distribution $\mathcal{D}_{pid}$ of the corresponding activity duration Θ_{pid}. In other words, t_{pid} is that value for which $P(\Theta_{pid} \leq t_{pid}) = \beta$. Since $\beta = 0.5$ gives the median of the corresponding activity duration, only values $\beta \geq 0.5$ make sense in view of the aim of increasing robustness. We chose the determination of the buffer sizes through a quantile because the β parameter can be applied to different distributions without the need of rescaling the value of β.

The schedule generation procedure `sched` takes a design, i.e., a treatment plan X, a patients priority list π, and a buffer β, and computes from this input a baseline schedule S_{bas}. A concrete example for X and π is shown in Fig. 2.

Algorithm 1 shows the pseudocode of the schedule generation procedure `sched`. The planned starting times of the activities, $\bar{s}_{pid}$, are determined according to a strategy first suggested by Vogl et al. (2018a). For each day, treatments of patients are inserted sequentially into the schedule. The sequence of insertions for a specific day is given by the global patient sequence π, except that any patients who do not have a planned treatment on that particular day get removed from the daily list. The three appointment phases (in-room preparation, irradiation, and exiting) need to be scheduled without idle time when constructing the baseline schedule. That is, we can fix the starting time

Algorithm 1: Solution Generation Algorithm.

for *day* $d = 1, \ldots, D$ **do**
  Set pointer to the first patient p on list π;
  repeat
    if *patient* p *gets a treatment on day* d $(x_{pd} = 1)$ **then**
      Determine activity durations t_{pid} (including buffer times) using buffer parameter β;
      $\bar{s}_{p1d} \leftarrow r_p$;
      repeat
        $\bar{s}_{p2d} \leftarrow \bar{s}_{p1d} + t_{p1d}$; $\bar{s}_{p3d} \leftarrow \bar{s}_{p2d} + t_{p2d}$;
        if *resulting schedule is infeasible* **then**
          $\bar{s}_{p1d} \leftarrow \bar{s}_{p1d} + 1$
        end
      until $\bar{s}_{p1d}$ *is a feasible starting time*;
      Schedule appointment for patient p at $\bar{s}_{p1d}$;
      if $\bar{s}_{p1d} > \tau_p$ **then**
        $\gamma_{pd} \leftarrow \bar{s}_{p1d} - d_p$
      end
    end
    Proceed to the next patient p on list π (if there is still one);
  until *patient list* π *processed*;
end

of the preparation activity and deduce the other starting times from that value. The earliest time when the preparation activity for a patient p can start is the ready time r_p of the patient-specific time window $[r_p, \tau_p]$. It is examined if r_p is a feasible starting time across all required resources over all activity phases. If yes, we fix the starting time and block the resources accordingly. If not, we increment the planned starting time until we find a feasible insertion position. If the final starting time for patient p is larger than the corresponding due time q_p, we record a penalty γ_{pd} in the objective function.

This approach differs from pure chronological scheduling in that we can fill "holes" in the schedule, as was proven to be beneficial by Vogl et al. (2018a) in a deterministic and static setting. Holes might occur if a patient with a later ready time appears early in the patient list, or if two patients are assigned successively to the same treatment room, creating idle time for the beam resource. In this second scenario, we might schedule another patient who requires a different treatment room in the interim and thereby minimize beam idle time.

4.2 The reactive procedure

As already anticipated in Sect. 3.1, the procedure `react` determines an actual schedule $S(\omega)$ based on a baseline schedule S_{bas} and random activity durations. In this context, it is assumed that the random variables Θ_{pid}, corresponding to the activity durations, are not realized up front, but rather revealed successively during the execution of the reactive procedure. By adopting this scheme, the procedure tries to mimic a human planner, who also has to deal with longer (or shorter) patient preparation, irradiation or exit times *as they occur*. However, this is not the only reason for the specific design

of the procedure. It was an initial requirement of the ion beam facility's managers to make it flexible and generic enough to be able to incorporate day-time dependent probability distributions and/or correlation between treatment times of subsequent patients. In essence, the reactive procedure not only provides the "true" durations and thus the starting and completion times of activities, it also allows for a preponement of activities in case the sampled durations of preceding activities are smaller than projected in the baseline schedule. A detailed description of the reactive procedure is given in Appendix D.

4.3 Solution evaluation

In view of the complexity of the functions `sched` and `react`, it seems hopeless to look for an *analytic* representation of the objective function of (2) as a function of the design (X, π, β). Therefore, we resort to Monte Carlo simulation in order to get a sufficiently precise approximation of the expected value occurring in (2). The simulation method approximates an expected value with respect to a distribution $\mathcal{D}$ by an average over a sample of randomly selected *realizations* drawn from $\mathcal{D}$ or from a related distribution. If the distribution is not changed during sampling (the latter is done, e.g., in the so-called importance sampling method), the weights of the realizations in the computation of the average are to be chosen as identical, which is the option we implemented.

It is obvious that the accuracy of the approximation is the better, the larger the sample size is. We shall work with different sample sizes, depending on the needed precision of the estimate (for details, see Sect. 5).

The formula for the estimation of the objective function value of a given design (X, π, β) is shown in Eq. (7) below. A number H of sets of realizations of random variables, i.e., activity durations Θ_{pid}, $p \in \mathcal{P}$, $i = 1, 2, 3$, $d = 1, \ldots, D$, are generated i.i.d. from distribution $\mathcal{D}$. Note that activity index $i = 1$ corresponds to the in-room preparation, $i = 2$ to the irradation and $i = 3$ to the post-irradiation exiting. For each set h, specified by concrete realizations Θ^h_{pid} of the activity durations, we apply the reactive procedure `react` to the baseline schedule $\texttt{sched}(X, \pi, \beta)$ and the durations Θ^h_{pid}, and compute the three terms in the objective function Eq. (6). This produces the following SAA estimate:

$$\bar{F} = \frac{1}{H}\left(\sum_{h=1}^{H} \left[\lambda_1 \cdot \sum_{d=1}^{D} \varphi_d^h + \lambda_2 \cdot \sum_{d=1}^{D} \sum_{p=1}^{n} \gamma_{pd} + \lambda_3 \cdot \sum_{d=1}^{D} \sum_{p=1}^{n} (\delta_{pd}^h + \rho_{pd}^h) \right] \right). \quad (7)$$

Therein, φ_d^h denotes the actual beam finishing time in the hth realization, and δ_{pd}^h and ρ_{pd}^h are the corresponding actual waiting times. Note that the time window violations, γ_{pd}, can already be calculated directly from the baseline schedule and do therefore not depend on the sampling procedure.

Equation (7) produces an unbiased estimate of the objective function value. We abbreviate the described evaluation strategy by STO. A disadvantage is that evaluation strategy STO is computationally expensive already for a medium-sized number

of realizations. Therefore, we also investigated two faster ways of getting estimates (though not unbiased ones anymore) of the true objective function value:

1. A deterministic approach (DET) to the problem approximates actual beam active time by the deterministic beam active time of the baseline schedule X_{bas}. Potential waiting times of patients are not taken into consideration at all. Consequently, this approach systematically underestimates the true objective function value. On the other hand, as the buffer increases, waiting times will diminish in general, possibly making the deterministic approach more competitive.
2. A quasi-deterministic variant that we call "waiting time estimation strategy" (WTE) approximates actual beam active time again by the deterministic beam active time $\hat{\varphi}_d$ of the baseline schedule. However, we do not neglect waiting times in WTE, but estimate actual waiting time by leveraging the observed correlation between idle time on the beam resource (excluding set-up time to particle type switches) and actual patient waiting time. Using this correlation, we can estimate patient waiting time by fitting a linear regression of the form

$$(\text{waiting time}) = A + B \times (\text{baseline schedule idle time}). \tag{8}$$

The data for this regression is gathered during the heuristic optimization from *some* specific intermediate schedules (created by the heuristic) and evaluating the sample averages of waiting times for those schedules. The regression gets updated regularly during the execution of the algorithm.

4.4 The (Meta-)heuristic solution approach

We apply a GA metaheuristic, as implemented successfully for various radiotherapy scheduling problems (Petrovic et al. 2009; Petrovic and Castro 2011; Vogl et al. 2018a, b), all of which deal with deterministic variants of the problem presented here. We modify the GA variant published in Vogl et al. (2018b), which relies on the offspring selection GA introduced by Affenzeller and Wagner (2004), such that it operates on the problem-specific solution encoding of a "design", as presented in Sect. 3.1. Note that the patient priority list as a part of a design is held constant across all days of the planning horizon. On the one hand, allowing for individual patient sequences per day would increase the problem's combinatorics dramatically, leading to excessive running times of the GA. On the other hand, in view of the desired adherence to time windows, synchronizing the patient sequences on successive days is at least not unreasonable.

The pseudocode of the used variant of the GA is shown in Appendix B of this paper. The offspring selection process of the GA favors individuals that outperform the fitness of at least one of their parents, where the fitness of an individual is given in our radiotherapy application by Eq. (1), with F defined by Eq. (6) and approximated by Eq. (7). The number of reproductive steps to build the next generation of individuals is limited.

The designs forming the initial population within the GA are generated in a randomized greedy fashion. For each patient p we choose a random treatment pattern x_p

from the set $\mathcal{X}_p$ of feasible treatment patterns. This produces a treatment plan X. The priority list π is either purely randomized or sorted according to the due times τ_p of the patients, with slight random variations. The initial value of β, finally, is chosen at random between 0.5 and 0.99.

Selection of individuals is done using the rank selection operator. Additionally, a fixed percentage of best individuals of the current generation survive, i.e., they are included in the next generation ("elitism"). The used crossover operators should preserve feasibility of the solutions generated from two parent solutions. Hence, we use the well-known position-based crossover for the patient priority list. The treatment plan is simply inherited by either parent #1 or parent #2 according to a random choice. The value of β is inherited from one of the parents in the same way. To create more diverse descendants, we use mutation operators applied to all three parts X, π and β of the design Z: (i) the treatment pattern x_p of a single random patient p is reset and newly generated, (ii) a randomly chosen patient shifts to a random new position in the priority list, and (iii) β is multiplied by a random number between 0.75 and 1.25.

5 Case study and data analysis

This section presents the practical case we faced. To estimate the underlying probability distributions of durations of the three radiotherapy activities (preparation, irradiation, and exiting), we analyzed real-world data from 113 patients and 2270 irradiation appointments. The data was collected in a newly opened ion-beam therapy center in Wiener Neustadt, Austria, among patients treated in 2017.

The main problem was to find a good compromise between distribution models that precisely reproduce the available past data, which would lead to over-fitting, and very general distribution models on the other hand, which would lead to under-fitting. We found that widely used distribution types for activity durations such as the lognormal distribution or the beta distribution yielded only a very poor fit with our data. That is why we searched for distribution types that better represent our data. We used Easyfit 5.6[2] as a dedicated software tool for this purpose.

It was assumed that the preparation and exiting activities of all patients follow the same family of distribution functions. The best-fitting family of distribution functions for these two types of activities turned out to be that of *Burr* distributions, with the general probability density function

$$f(x) = a \cdot k \cdot (\frac{x}{b})^{a-1} \cdot \frac{1}{b \cdot (1 + (\frac{x}{b})^a)^{k+1}} \tag{9}$$

and the concrete parameter values shown in Table 10 (see Appendix C). Here, b is a scale parameter, and a and k are shape parameters. The resulting density functions are shown in Fig. 5 of Appendix C.

For the irradiation activities, we conjectured that a different family of distribution functions might probably give a better fit, and that the concrete parameters of the pdf

[2] http://www.mathwave.com/easyfit-distribution-fitting.html.

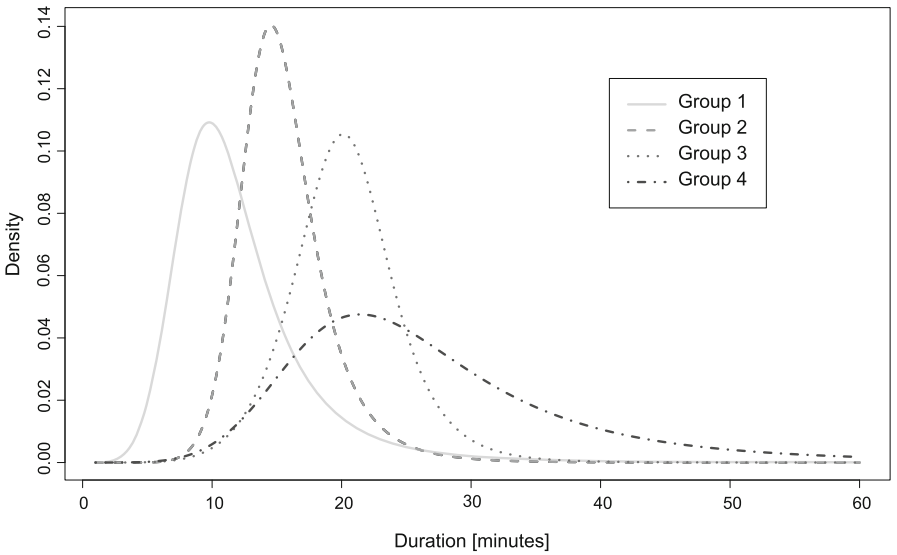

Fig. 3 Distribution of Duration of Irradiation Activity for 4 Patient Groups

might heavily depend on patient characteristics. Therefore, we clustered the patients into four groups according to treatment complexity and planned activity durations, and applied Easyfit to each of these groups. It turned out that in this case, the *Dagum* distribution, also known as the emphInverse Burr distribution, achieves the best fit. Again, the parameter values can be found in Appendix C (Table 10). A graphical representation of the four density functions is shown in Fig. 3.

King (2017) provides details on the families of Burr and Dagum distribution functions. All distributions are asymmetric and right-skewed, reflecting the comparably large probability of outliers that exhibit considerably higher duration than the expected values.

6 Results

This section provides results of extensive computational tests on randomly generated problem instances of varying sizes. In Sect. 6.1, we begin with a brief description of the randomly generated instances, optimization parameters, and environment used for the computational study. Then, Sect. 6.2 addresses the problem of determining the optimal buffer parameter β, and Sect. 6.3 thoroughly compares the solution evaluation procedures from Sect. 4.3 on the different buffer parameter values β and objective function weights. Finally, Sect. 6.4 compares our GA approach to simple rules of thumb that might be used by a human planner.

6.1 Experimental setup of computational tests

To generate instances for the computational tests, we use data from MedAustron (see Sect. 5). We analyze test instances where patients are randomly assigned to the four

Table 1 Classes of Patients and Corresponding Treatment Patterns

Class	N_p	Pattern	Instances	
			1–8 (%)	9–16 (%)
1	5	(1, 1, 1, 1, 1)	44	20
2	3	(0, 0, 1, 1, 1)	4	7
3	2	(1, 1, 0, 0, 0)	6	6
4	3	(1, 1, 1, 0, 0)	6	6
5	4	(1, 1, 1, 1, 0)	4	8
6	4	Earliest break Thursday	6	10
7	4	Earliest break Wednesday	8	12
8	4	Earliest break Tuesday	10	14
9	4	Earliest break Monday	12	16

groups described in Sect. 5 according to the following probabilities: with probability 43%, 29%, 22% and 6%, a patient is assigned to group 1 to 4, respectively. A sequence-dependent set-up time of 3 min is considered if two patients with different beam types (protons or carbon ions) are scheduled sequentially on the beam resource. We randomly assign the beam type to patients, such that 50% receive proton therapy and the other 50% are irradiated with carbon ions. The distribution of patients among the three treatment rooms is assumed to be balanced, with a probability of 33% for each room.

For the required number of treatments and the corresponding treatment pattern, we distinguish nine patient classes. The associated treatment patterns (cf. Sect. 3) are shown in Table 1. We generated 16 random instances. Instances 1 through 8 select patients from the nine classes according to the first column of probabilities in Table 1, whereas instances 9 to 16 use the second column.

The instance size varies from 30 to 100 patients. The ready times for the daily treatments and the time window length are chosen randomly, where the average length of the time window slightly increases with the ready time: the average time window length for the first half of the day is 276 min, and it increases to 360 min for the second half of the daily planning horizon. Furthermore, we assume that 20% of patients do not have time window preferences. We test multiple combinations of the objective function weights $\lambda_1, \lambda_2, \lambda_3$ for the actual beam active time, the time window violations, and the actual patient waiting time, respectively, including the balanced case $\lambda_1 = \lambda_2 = \lambda_3 = 1.0$ and two cases favoring exploitation of beam capacity over patient-centered waiting time and time window violations, namely, $\lambda_1 = 1.0, \lambda_2 = \lambda_3 = 0.5$ and $\lambda_1 = 1.0, \lambda_2 = \lambda_3 = 0.1$.

For the GA from Sect. 4.4, we use the following parameters which have proven to be beneficial in preliminary tests: the population size is 100, the mutation rate is 10%, and the elite segment is 1% of the population. We aim to build 70% of the new population from children who meet the defined success criterion, while the number of reproductive steps is limited to 500 (if we fail in building a new population within this limit, we fill up the population with random children).

All algorithms were implemented in C++, and the experiments were carried out on the Vienna Scientific Cluster (VSC3) equipped with compute nodes with two Intel Xeon E5-2650v2, 2.6 Ghz, and 8 core CPUs each. The runs consisted of two phases, which will be described below. The CPU time for in phase 1 was chosen as $n/10$ h, where n indicates the number of patients in the corresponding instance. The CPU time in phase 2 was 3600 s.

6.2 Phase 1: optimal buffer determination

As discussed in Sect. 4.1, the value β of the buffer parameter has an essential influence on the performance of the approach and is therefore part of the decision. In principle, β could be (heuristically) optimized by our GA together with the other components of the design $Z = (X, \pi, \beta)$ specifically for each new week, based on the actual instance parameters for the current week, such as the exact number of patients per group, the assigned treatment patterns, time windows, etc. However, since the distribution of patient characteristics does not essentially change from week to week, this procedure is not the most efficient one. In preliminary experiments, we observed that the optimal value of β turned out as very stable under different instances generated according to the distribution described in Sect. 6.1, as long as the total number of patients and the weights in the objective function were kept fixed. Therefore, it saves computation time and does not deteriorate the results if the value of β is considered as a *strategic* decision, made only once for a larger number of weeks in which no essential change in the distribution of patient characteristics is to be expected, whereas the choices of X and π (and thus of the baseline schedule) are *operational* decisions to be made at the beginning of each week for the current data. In the current subsection, we deal with the strategic decision on β, and will turn to the operational decision in the next subsection.

The main advantage of considering the choice of β as a strategic decision is that a much larger runtime can be devoted to a decision that has not to be repeated each week. We exploit this by creating larger samples of size 1000 during the evaluation of the objective function when applying the evaluation procedure STO from Sect. 4.3 in the context of the heuristic optimization of β, increasing in this way the precision of the estimate. For the operational (i.e., weekly) planning, a smaller samples size will be used in order to be fast. In these latter computational runs, the value of β is not varied anymore by the GA, but "frozen" to the pre-determined value from the strategic computation.

In the strategic run, we executed the GA with computation times of 3, 5, 7, and 10 h for four different instance sizes n, namely 30, 50, 70, and 100 patients. Table 2 summarizes the findings for the patient mix described in Sect. 6.1 for different values of n and the vector $(\lambda_1, \lambda_2, \lambda_3)$ of objective function weights indicated above (characterized by λ_3). It can be seen that for a given vector $(\lambda_1, \lambda_2, \lambda_3)$, only small differences over the instance sizes n result. For each line, we find clear outliers in both the minimum and maximum optimized buffer parameter. Yet the inter-quartile range is generally small, leading us to conclude that the optimal buffer sizes are approximately 0.66 to 0.70 for $\lambda_3 = 0.1$, 0.78 to 0.80 for $\lambda_3 = 0.5$, and 0.80 to 0.83 for $\lambda_3 = 1.0$.

Table 2 Statistics of optimized buffer parameters β^* for 16 random instances with patient mix probabilities {43%, 29%, 22%, 6%}; 16 replications per instance

λ_3	n	Mean	Min	25%	50%	75%	Max
0.1							
	30	0.69	0.52	0.68	0.70	0.71	0.74
	50	0.67	0.52	0.66	0.68	0.70	0.73
	70	0.67	0.50	0.64	0.68	0.70	0.75
	100	0.66	0.58	0.64	0.66	0.69	0.73
0.5							
	30	0.80	0.76	0.79	0.80	0.81	0.84
	50	0.79	0.74	0.79	0.80	0.80	0.83
	70	0.79	0.74	0.78	0.79	0.80	0.81
	100	0.78	0.71	0.77	0.79	0.79	0.82
1.0							
	30	0.83	0.78	0.82	0.83	0.84	0.86
	50	0.82	0.75	0.82	0.82	0.83	0.85
	70	0.81	0.75	0.80	0.82	0.82	0.85
	100	0.80	0.72	0.79	0.80	0.81	0.84

To investigate the dependence of the optimal buffer parameters on the patient mix, we also generated instances with artificial patient mixes. Again, each instance ran 16 times with different random seeds. Table 3 presents the average optimized buffer parameters β^* for different patient mixes and instance sizes. The instances with patients from groups 3 ({0%, 0%, 100%, 0%}) and 4 ({0%, 0%, 0%, 100%})—which feature the largest variance and simultaneously the highest expected activity durations—result in the smallest optimal buffers. Especially for patient group 4, beam active time increases drastically with a higher buffer percentile. This effect gets smaller as λ_3, the weight of waiting time in the objective function, increases and the importance of the beam active time simultaneously diminishes.

The optimized buffer parameters β^* of the other arbitrarily chosen patient mixes only slightly deviate from the optimized buffers that result from the patient mix we observed in the real-world data sets.

6.3 Phase 2: schedule optimization

The second part of our computational study focuses on the comparison of the different approaches DET, STO and WTE, respectively, to approximate the true objective function, i.e., the expected costs (see Sect. 4.3). For STO, preliminary test have shown that setting $H = 100$ during the intermediate evaluations in the GA yields a good compromise between evaluation accuracy and running time limits. In order to be able to compare the final results produced by the three approaches on safe grounds, we perform objective function evaluations based on 1,000,000 realizations of activity durations. Table 4 reports the average results for overall expected costs (i.e., the weighted objective), beam active duration, waiting times, and penalties for time win-

Table 3 Optimal buffer parameters of artificial patient mixes, 16 replications per instance

Patient mix [%]	0.1				0.5				1.0			
	30	50	70	100	30	50	70	100	30	50	70	100
{100, 0, 0, 0}	0.73	0.71	0.72	0.70	0.81	0.80	0.77	0.76	0.84	0.82	0.77	0.76
{0, 100, 0, 0}	0.71	0.72	0.70	0.72	0.82	0.80	0.80	0.79	0.84	0.84	0.82	0.82
{0, 0, 100, 0}	0.59	0.61	0.52	0.51	0.82	0.82	0.79	0.61	0.86	0.85	0.83	0.63
{0, 0, 0, 100}	0.54	0.61	0.59	0.58	0.70	0.72	0.69	0.62	0.77	0.77	0.70	0.61
{25, 25, 25, 25}	0.64	0.63	0.62	0.62	0.78	0.78	0.78	0.73	0.81	0.81	0.80	0.73
{40, 20, 20, 20}	0.69	0.59	0.69	0.64	0.79	0.79	0.76	0.73	0.83	0.82	0.78	0.74
{20, 40, 20, 20}	0.56	0.65	0.64	0.64	0.78	0.79	0.78	0.73	0.82	0.82	0.81	0.75
{20, 20, 40, 20}	0.60	0.59	0.51	0.61	0.79	0.79	0.76	0.69	0.82	0.82	0.80	0.70
{20, 20, 20, 40}	0.66	0.66	0.61	0.64	0.77	0.78	0.75	0.68	0.81	0.81	0.79	0.68

Table 4 Average results of 16 randomly generated instances and 16 replications for each instance

		Exp. objective			Beam, $\lambda_1 = 1.0$			Waiting, $\lambda_3 = 0.5$			Penalty, $\lambda_2 = 0.5$		
Pat.	β	DET	STO	WTE	DET	STO	WTE	DET	STO	WTE	DET	STO	WTE
30	0.50	4300	**3447**	4243	2400	2572	2483	3800	1740	3520	0	11	1
30	0.60	3700	**3252**	3679	2419	2591	2428	2562	1317	2501	0	5	0
30	0.70	3198	**3081**	3184	2478	2599	2481	1440	960	1405	1	4	0
30	0.78	2968	2974	**2964**	2580	2633	2580	774	675	765	3	7	4
30	0.80	2946	2958	**2941**	2615	2657	2615	654	593	645	6	9	6
30	0.83	2960	2975	**2955**	2697	2727	2697	512	479	501	15	17	15
30	0.90	**3237**	3265	3240	3019	3049	3022	312	295	307	125	138	128
50	0.50	8289	**5653**	7928	3819	4052	4219	8939	3138	7411	3	64	7
50	0.60	6670	**5272**	6536	3835	4102	4043	5663	2299	4979	7	41	8
50	0.70	5305	**4979**	5266	3926	4136	3953	2726	1633	2595	32	53	31
50	0.78	4785	4807	**4774**	4113	4218	4119	1262	1086	1227	82	93	82
50	0.80	4747	4788	**4743**	4168	4252	4172	1051	953	1035	108	119	107
50	0.83	4778	4830	**4776**	4298	4373	4301	803	748	790	156	165	160
50	0.90	**5320**	5407	5323	4833	4917	4844	467	438	460	507	543	499
70	0.50	13890	**8214**	12405	5438	5753	6007	16903	4764	12779	1	158	18
70	0.60	10505	**7521**	9857	5467	5834	5913	10071	3283	7878	5	92	9
70	0.70	7778	**7091**	7679	5594	5900	5666	4351	2336	4010	16	45	15
70	0.78	6827	6864	**6808**	5911	6060	5918	1769	1520	1721	65	86	59
70	0.80	6763	6825	**6756**	5990	6112	6002	1454	1315	1419	92	112	90
70	0.83	6810	6902	**6809**	6191	6302	6199	1087	1012	1066	151	188	156
70	0.90	**7931**	8245	7947	6966	7104	6975	613	571	608	1316	1710	1336
100	0.50	22637	**11922**	19351	7899	8275	8886	29475	7042	20898	0	252	32
100	0.60	16295	**10963**	14715	7888	8400	8730	16812	5030	11956	1	96	15
100	0.70	11363	**10220**	11134	8108	8506	8325	6508	3395	5614	3	34	3
100	0.78	9850	9928	**9846**	8580	8754	8609	2450	2135	2366	90	213	109
100	0.80	9793	9952	**9778**	8690	8854	8709	2010	1823	1961	196	372	179
100	0.83	9953	10140	**9948**	8995	9136	9013	1481	1395	1451	436	613	420
100	0.90	**12937**	13796	12961	10070	10241	10087	830	785	822	4904	6326	4927

$\lambda_3 = 0.5$, optimization time limit of 1 h. Bold numbers represent the best solutions per line. Lines with optimal buffer parameters are in gray

Table 5 Results of the Wilcoxon–Mann–Whitney test on a pairwise comparison of methods STO, DET and WTE

λ_3	β	STO vs. DET			STO vs. WTE			WTE vs. DET		
		STO	DET	Equal	STO	WTE	Equal	WTE	DET	Equal
0.5										
	0.50	16	0	0	16	0	0	12	0	4
	0.60	16	0	0	16	0	0	14	0	2
	0.66	16	0	0	16	0	0	13	0	3
	0.68	16	0	0	16	0	0	13	0	3
	0.70	16	0	0	16	0	0	11	0	5
	0.78	1	5	10	1	9	6	4	1	11
	0.80	***0***	***14***	***2***	***0***	***14***	***2***	***1***	***1***	***14***
	0.83	0	16	0	0	16	0	0	0	16
	0.90	0	16	0	0	16	0	0	0	16

dow violations, for $\lambda_1 = 1.0$, $\lambda_2 = 0.5$ and $\lambda_3 = 0.5$. Some general patterns can be observed.

First, the waiting time is the smallest for the stochastic optimization variant STO, as expected. DET and WTE may produce extremely large waiting times, especially when the buffer parameter β is small. The advantage of STO diminishes with an increasing buffer parameter β, as might be expected. The larger the buffer parameter, the longer the planned activity duration, and the smaller the probability that a patient will take longer than the planned duration. Secondly, the beam active time increases with growing buffer parameter β, yet the increase is surprisingly small. Third, the penalty term is almost negligible for most instances. However, for very large buffer parameter values, the total planned waiting time becomes so large that the time windows preferred by the patients cannot be respected anymore, leading to a significant penalty in the objective function. A buffer parameter of 0.90 or larger would only be optimal if the waiting time weight in the objective function were extremely high, which is not realistic in practice. Similar effects can be noticed when modifying the objective function weights, as shown in Appendix E for $\lambda_2 = \lambda_3 = 0.1$ (see Table 11) and $\lambda_2 = \lambda_3 = 1.0$ (see Table 12).

To assess the quality of the three approaches, we also performed statistical tests. The well-known Wilcoxon-Mann-Whitney test was applied to pairwise combinations of methods DET, STO and WTE. For each combination of β and λ_3 and for each of 16 random instances, we analyzed 16 runs of the respective methods with different seeds. We chose a significance level of $\alpha = 0.05$. Table 5 shows the results of the significance tests for 100 patients ($\lambda_3 = 0.5$). The values in the table indicate the number of instances in which one of the methods is significantly better, or where the tests reveal no significant difference among methods, respectively. The full table with results from the significance tests also for $\lambda_3 = 0.1$ and $\lambda_3 = 1.0$ is provided in Appendix E (Table 13).

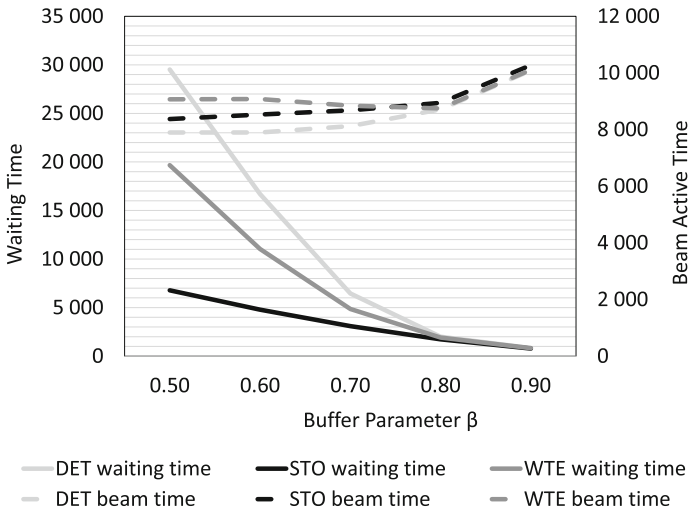

Fig. 4 Average Results for 100 Patients from Table 12, Graphically

When comparing the behavior of the three solution approaches, we can identify different patterns for different regimes of the buffer parameter β: (1) For small β, the stochastic approach STO has a clear advantage, in that it considers the waiting time of the patients directly by applying Sample Average Approximation to the evaluation of each solution candidate in the population. This effect gets stronger with a larger waiting time weight in the objective function (see Table 4). (2) Using the optimized buffer parameter values (phase 1), the picture changes slightly, and all solution approaches lead to comparable results. A small advantage accrues to the WTE approach, which performs slightly better than DET on average, followed by STO. However, the differences among the three approaches are rarely statistically significant in this buffer size regime. (3) A higher than optimal β favors DET over other approaches (though not statistically significantly, as Table 5 shows).

Although the stochastic STO approach usually does not provide superior results for operational (weekly) planning problems for optimized buffer sizes, relative to the two other approaches, it is required to solve the strategic problem of determining the optimal buffer parameter itself. For this purpose, it cannot be replaced by the other approaches.

Figure 4 shows the evolution of mean waiting times and beam active times, depending on the buffer parameter β. Again, we see that the different solution approaches DET, STO, and WTE differ substantially in the waiting times for small β. The larger the buffer parameter, the more similar the results.

6.4 Comparison to simple rule-based approaches

To be able to better assess the results achieved by the GA, as discussed in Sect. 6.3, we compare them to simple rule-based approaches that could also be employed by a human planner. The first rule, referred to as the "latest starting time" (LST) rule, has already

Table 6 Performance of simple rule-based approaches for generating patient sequences (100 patients, $\beta = 0.8$)

Method	Avg. Objective	Beam-on	Waiting	Penalty
Best GA	10,830	8746	1905	180
ERT	77,476	9208	1607	66,661
LST	13,949	9372	1682	2895
Rand. LST	17,675	9434	1586	6654
Random	91,430	9593	1389	80,447
LDV	81,285	10,164	1263	69,859

been described in Sect. 4.4 in the context of the GA's initial population. It creates a patient sequence by simply sorting the patients in non-decreasing order of their *latest* starting times. In case the same room is occupied by two directly consecutive patients, the decoder would account for this through its "hole-filling" strategy. The LST rule also exists in a randomized version, constructing the patient sequence in a stepwise manner. At each step, the patient for the next sequence position is randomly chosen from a list of patients with the ten smallest latest starting times among all remaining ones. To account for the beam-on time, it might be beneficial to schedule the patients in a sequence that causes the least idle time on the beam resource. Again, this is done in an iterative fashion, based on the patients' earliest availabilities. We refer to this rule as the "earliest release time" (ERT) rule. A completely different approach is to rank the patients according to their treatment time variance in non-decreasing order. The idea of this "least duration variance" (LDV) rule is to move patients with highly varying treatment durations to later times of the day. The stable time windows are ignored, however. Finally, we take a look at purely randomized patient sequences, to get a glimpse of what is actually the absolute baseline performance.

Table 6 summarizes the results obtained by the rules described above and compares them to the best solution obtained by the GA under the same experimental conditions. The computational evaluation was conducted using a set of 16 instances with 100 patients (see Sect. 6.1), objective function weights $\lambda_1 = \lambda_2 = \lambda_3 = 1.0$, a buffer parameter $\beta = 0.8$ and a sample size of 1000000. It can be observed that the two LST-based rules perform best among the simple rules from an overall perspective. The other rules achieve exactly what they were designed for, with obvious side effects: LDV is in fact able to achieve the lowest waiting time, but entails also the second highest time window violation penalties. A similar effect arises for ERT which in turn leads to the lowest beam on time.

6.5 Linkage to chance-constrained programming

From the facility's management perspective, it might be desirable to guarantee, or at least advertise a certain level of service or reliability. Still considering the patients' waiting time as the most crucial factor, this could be about trying to keep it within a certain range. A threshold value or upper bound on this waiting time might reflect

Table 7 Average waiting time and threshold violation penalties for instances with 100 patients and various combinations of buffers and waiting time thresholds

Buffer	Threshold	Waiting	Thr. Penalty
0.6	10	4980	172,459
0.6	20	4172	31
0.7	10	2728	8197
0.7	20	2813	0
0.8	10	1699	0

a "bearable" time span that patients are still willing to spend in the waiting area of the facility before getting too impatient. It is not always necessary to ensure ultimate adherence to that threshold value for all patients, especially under uncertainty. Rather, it is common to define a certain probability by which the threshold can be exceeded. This is usually covered by *chance-constrained programming* (CCP) (Geletu et al. 2013; Shylo et al. 2013; Deng et al. 2019, see,e.g.). However, due to the complexity of the second stage decisions, involving non-linear decision making by the reactive procedure described in Sect. 4.2, we currently see no way of capturing this part of our approach in a classical, analytical stochastic programming formulation. Therefore, also the implementation of chance constraints in the conventional sense appears out of reach. Nevertheless, we try to cover the reliability aspect by adopting a slightly different approach by modifying the objective function such that a "stochastic threshold" can be considered.

We define a threshold value $\overline{W}$ for the average (total) waiting time of a patient and count the number of cases in which that threshold value is exceeded (across a sample of size H). The corresponding relative quantity can be interpreted as the probability of violating the mean waiting time upper bound constraint. To limit that probability to a predefined value, denoted by α, we modify the existing, sampling-based objective function F in the following fashion:

$$\tilde{F} = \frac{1}{H}\left(\sum_{h=1}^{H}\left[\lambda_1 \cdot \sum_{d=1}^{D}\varphi_d^h + \lambda_2 \cdot \sum_{d=1}^{D}\sum_{p=1}^{n}\gamma_{pd}\right]\right) + M \cdot \left\{\frac{1}{H}\cdot\sum_{h=1}^{H}\mathbb{I}\left(\frac{1}{D\cdot n}\sum_{d=1}^{D}\sum_{p=1}^{n}(\delta_{pd}^h + \rho_{pd}^h) \geq \overline{W}\right) - \alpha\right\}^+, \tag{10}$$

where $\mathbb{I}$ is an indicator function, yielding the value 1 if the expression passed as an argument evaluates to "true" and 0 otherwise. The parameter M is a penalty cost coefficient that is also provided as a constant. When embedding the overall objective function $\tilde{F}$ in our GA, we gradually increase M during the run of the GA to even more strongly enforce the creation of "feasible" solutions. Given a particular value α, a solution is called feasible if the proportion of the sample violating the waiting time upper bound $\overline{W}$ stays below α. Table 7 shows an analysis of GA runs using objective function $\tilde{F}$ (Eqn. (10)). It is based on 16 instances with 100 patients (16 independent

runs per instance), $\alpha = 0.01$, $\beta \in \{0.6, 0.7, 0.8\}$, $M = 10000$, and waiting time thresholds ($\overline{W}$) of 10 and 20 min, respectively.

It can be observed that it is not possible to ensure that in less than 1% of the cases the threshold value for $\beta = 0.6$ and $\beta = 0.7$ and $\overline{W} = 10$ is exceeded. Being able to ensure average waiting times of less than 10 min with a probability of 99%, as it is the case when setting $\beta = 0.8$, might be an appealing goal, also from a practical perspective. We think that this confirms once more the effectiveness of the buffer approach also in this quite specific setting.

7 Conclusion

The present study confirms the importance of considering stochastic activity durations, well-known in the literature on appointment scheduling, for the case of radiotherapy scheduling. In this context, an issue complicating the scheduling process is the presence of pre-treatment and post-treatment activities, the durations of which are stochastic as well. In total, this gives rise to a stochastic decision process with complex dependencies resulting from the precedence constraints at the patients' level on the one hand, and from the fact that the beam resource is shared by multiple competing treatment rooms on the other hand.

To account for possible variations in activity durations and to produce more robust baseline schedules, we rely on a buffer parameter describing the quantile of the fitted distribution to which the planned activity durations are to be enlarged. The determination of the optimal buffer size itself requires stochastic sampling, as the objective function value in our model (the mathematical expectation of a quantity resulting from a complex scheduling procedure) cannot be computed analytically. In an attempt to save computation time at least in the course of the (heuristic) optimization of the schedule on the operational level of weekly planning, we also investigate two alternatives to stochastic sampling, namely a deterministic variant that completely ignores waiting time, and a quasi-deterministic variant that estimates the expected waiting time of a schedule through linear regression. Our numerical results suggest that if the buffer size has already been adjusted optimally, then on the operational decision level, stochastic sampling can be replaced by the faster regression-based approach without loss of solution quality.

Our approach might also be applicable to similar settings in other ion beam facilities, as long as only one beam resource is involved. The number of treatment rooms, on the other hand, can be arbitrary. Especially the buffer concept is quite generic, since it is independent of the fitted distribution(s).

We plan to continue studying more advanced techniques to estimate actual waiting times by identifying appropriate features of the baseline schedule. In particular, more sophisticated regression approaches from machine learning domains may be helpful. Solving the underlying problem to optimality using two-stage stochastic programming would be another interesting stream of research, which, however, would possibly

require a more stylized reactive scheduling strategy to make the expected cost of the recourse action representable by explicit mathematical expressions.

Funding Open access funding provided by University of Vienna.

A Symbols and abbreviations

Tables 8 and 9 list all symbols and abbreviations used throughout the paper and in the appendix. The tables are split into four parts: Sets, input data, variables that are fixed when building a (deterministic) baseline schedule (Table 8), and random variables, which are either drawn during the executing of the reactive procedure or can be deduced from those randomly drawn variables when applying the reactive procedure (see Table 9).

Table 8 Sets, input data, and baseline schedule variables to describe the radiotherapy appointment scheduling problem

Notation	Description
$\mathcal{P}$	Set of all patients, $p \in \{1, \ldots, n\}$
$\{1, \ldots, D\}$	Set of days in the planning horizon, $d \in \{1, \ldots, D\}$
$\mathcal{I}$	Set of activities, $i \in \{1, 2, 3\}$
$\mathcal{R}$	General set of resources, index $r \in \{1, \ldots, R\}$
$\mathcal{R}^{pi}$	Set of required resources for activity i and patient p
$\mathcal{R}^{Room}$	Set of room resources
$\mathcal{R}^{Beam}$	Set including the beam resource
$\mathcal{H}$	Set of realizations, $h \in \{1, \ldots, H\}$
β	Buffer parameter, $0.5 \leq \beta \leq 1.0$
t_{pi}	Planned duration of activity i of patient p including possible buffer, input
u_{pqr}	Setup time between patient p and patient q on resource r, input
N_p	Number of daily treatment sessions for patient p in the planning horizon, input
$\lambda_1, \lambda_2, \lambda_3$	Objective function weights for beam time, time window penalties and waiting time, respectively, input
r_p	Daily release time of patient p, input
q_p	Daily due time for patient p, input
A_{pd}	Forced treatment day for patient p on day d, binary input
$\bar{s}_{pid}$	Planned starting time for patient p's activity i on day d, variable fixed in baseline schedule

Table 8 continued

Notation	Description
s_{pidr}	Planned starting time for patient p's activity i on day d at resource r, variable fixed in baseline schedule
$\hat{f}_d$	Planned finish time of last activity on the beam resource on d, variable fixed in baseline schedule
γ_{pd}	Time window violation for patient p on day d, variable fixed in baseline schedule
y_{pqdr}	Binary variable indicating immediate successor of patient q, namely patient p, on day d and resource r, fixed in baseline schedule
a_{pd}	Binary variable indicating whether a DT takes place on day d for patient p, fixed in baseline schedule

Table 9 Random variables of the radiotherapy appointment scheduling problem

Notation	Description
$\Theta_{pid}^{(h)}$	Actual duration of activity i of patient p on day d (in realization h), random variable
$\bar{\sigma}_{pid}$	Actual starting time for patient p's activity i on day d, calculated through reactive procedure
$f_d^{(h)}$	Actual finish time of last activity on the beam resource on d (in realization h), calculated through reactive procedure
$\delta_{pd}^{(h)}$	Pre-preparation waiting time for patient p on day d (in realization h), calculated through reactive procedure
$\rho_{pd}^{(h)}$	Pre-irradiation waiting time for patient p on day d (in realization h), calculated through reactive procedure

B Genetic algorithm with offspring selection as proposed in Affenzeller and Wagner (2004)

Algorithm 2: GA with Offspring Selection (Affenzeller and Wagner 2004)

$\mathcal{P}_0 \leftarrow$ CreateInitialPopulation();
$s_{best} \leftarrow \text{argmin}_{p \in \mathcal{P}_0}(\texttt{ObjVal}(p))$;
$i \leftarrow 0$;
repeat
  $\mathcal{P}_{i+1} \leftarrow$ GetElites($\mathcal{P}_i$);
  $C^B \leftarrow \emptyset$;
  while $|\mathcal{P}_{i+1}| < 0.7 \cdot |\mathcal{P}_i| \wedge |\mathcal{P}_{i+1}| + |C^B| < 5 \cdot |\mathcal{P}_i|$ **do**
    $p_1 \leftarrow$ PerformSelection($\mathcal{P}_i$);
    $p_2 \leftarrow$ PerformSelection($\mathcal{P}_i$);
    $c \leftarrow$ Crossover(p_1,p_2);
    $c \leftarrow$ Mutate(c);
    if $\texttt{ObjVal}(c) < min(\texttt{ObjVal}(p_1), \texttt{ObjVal}(p_2))$ **then**
      $\mathcal{P}_{i+1} \leftarrow \mathcal{P}_{i+1} \cup \{c\}$;
    else
      $C^B \leftarrow C^B \cup \{c\}$;
    end
  end
  while $|\mathcal{P}_{i+1}| < |\mathcal{P}_i|$ **do**
    $c \leftarrow$ ChooseRandomElement(C^B);
    $\mathcal{P}_{i+1} \leftarrow \mathcal{P}_{i+1} \cup \{c\}$;
    $C^B \leftarrow C^B \setminus \{c\}$;
  end
  if $\text{argmin}_{p \in \mathcal{P}_{i+1}}(\texttt{ObjVal}(p)) < \texttt{ObjVal}(s_{best})$ **then**
    $s_{best} \leftarrow$ BestOfPopulation($\mathcal{P}_{i+1}$);
  end
  $i \leftarrow i + 1$;
until *termination criterion met*;

C Detailed distribution fitting results

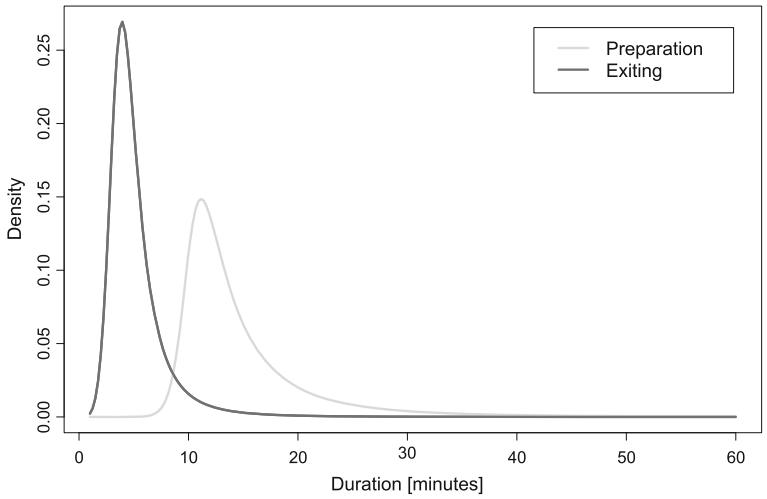

Fig. 5 Distribution of the durations of preparation and exiting activities

Table 10 Properties of fitted distributions for preparation (Pre), irradiation (Irr), and exiting (Ex) activities

Act.	Dist	Group	%P	k	a	b	Mean	Stdev	CV	25%	50%	75%
Pre	Burr	–	100%	0.2	13.4	10.3	15.1	8.9	0.6	11.0	12.9	16.3
Irr	Dagum	1	43%	1.4	4.1	10.0	12.3	5.8	0.5	8.8	11.1	14.3
Irr	Dagum	2	29%	1.3	7.7	14.5	15.6	3.7	0.2	13.3	15.2	17.4
Irr	Dagum	3	22%	0.6	10.1	21.6	20.3	4.4	0.2	17.6	20.2	22.8
Irr	Dagum	4	6%	1.5	3.8	21.4	27.7	14.1	0.5	19.3	24.7	32.3
Ex	Burr	–	100%	0.6	5.3	3.9	5.2	3.2	0.6	3.6	4.5	5.9

"Act." gives the activity type, "Dist" describes the distribution family, "Group" is the patient group and, "%" is the corresponding probability of a patient belonging to the given group according to estimates by the facility. Whereas "k," "a," and "b" are the distribution parameters, "mean," "stdev," and "CV" the mean, standard deviation, and coefficient of variation of the distribution, respectively. The columns "25%," "50%," and "75%" include the quartiles of the distribution

D A detailed look on the reactive procedure

Algorithm 3 shows the functional principle of the reactive procedure. For a given day $d \in \{1, \ldots, D\}$, it schedules the patients' activities in a chronological manner, in non-decreasing order of their planned start times. For this purpose, a queue is used as central data structure. Once an activity is dequeued, first its actual start time is determined. Depending on the type of the activity, this step is more or less complicated. In fact, it is quite simple for exit activities, because they are assumed to start as soon as the room is available and the irradiation activity has finished. The course of action taken for the irradiation activities is similar, with one exception: a setup (changeover) time may have to be considered, subject to whether a beam switch is required between the preceding irradiation and the current one. The setup time required to switch the beam between two patients q and p is denoted by u_{qp}.

The determination of an activity's preparation start time requires slightly more effort. Let p denote the current patient (at index position j) and q the patient occurring immediately before p in the patient priority list (sequence) π, i.e., $q = \pi_{[j-1]}$. The actual logic for the computation of the preparation start time $\bar{\sigma}_{p1d}$ is encapsulated in procedure `SetPreparationStartTime` (see Algorithm 4). A very important assumption in this context is the arrival time of the patients, which, according to the facility, can be expected to be 15 min before their planned starting time. This permits a preponement of activities for which we distinguish two different cases:

- *Case 1* The planned preparation start time $\bar{s}_{p1d}$ of the current patient p is larger than the actual irradiation start time $\bar{\sigma}_{q2d}$ of its predecessor q and the irradiation of q has in fact been started earlier than planned. Then the preparation activity of p can be preponed as well. Figure 6 depicts the idea of this approach: the amount by which the preparation of p is started earlier is simply determined as $\Delta = \bar{s}_{q2d} - \bar{\sigma}_{q2d}$.
- *Case 2* The planned preparation start time $\bar{s}_{p1d}$ of the current patient p is larger than the actual preparation start time $\bar{\sigma}_{q1d}$ of its predecessor q and the preparation of q has in fact been started earlier than planned. Also in this case, p's activity can

Algorithm 3: Outline of procedure `react`

In : A baseline schedule S_{bas}, providing planned start times $\bar{s}_{pid}$ for all activities, a vector Θ of *partially* revealed random activity durations and a day index d.

```
1  init activity queue Q ← ((π[1], 1));
2  init activity pointers for patients h[p] ← 1, for all p ∈ P;
3  init earliest resource starting times w_r ← 0, for all r ∈ R;
4  σ̄_pid ← ∞, for all p ∈ P, i = 1, 2, 3, d ∈ D;
5  j ← 1 ;          /* Current position in patient sequence */
6  while Q.size() > 0 do
7      (p, i) ← Q.dequeue();
8      if i = 1 then
9          q ← π[j−1];
10         SetPreparationStartTime(j, p, q, s̄, σ̄, w, r);
11         Sample Θ_p1d from prep. time distribution;
12         w_μ(p) ← σ̄_p1d + Θ_p1d;
13     else if i = 2 then
14         σ̄_p2d ← max(w_0 + u_qp, σ̄_p1d + Θ_p1d);
15         Sample Θ_p2d from irradiation time distribution;
16         w_0 ← σ̄_p2d + Θ_p2d;
17     else
18         σ̄_p3d ← max(w_μ(p), σ̄_p2d + Θ_p2d);
19         Sample Θ_p3d from exit time distribution;
20         w_μ(p) ← σ̄_p3d + Θ_p3d;
21     end
22     if j < n and h[π[j+1]] ≤ 3 then
23         p' ← π[j+1] ;      /* Check next patient in sequence */
24         i' ← h[p'];
25         if s̄_p'i'd < σ̄_pid + Θ_pid then
26             TryEnqueue(Q, h, p', i');
27             TryEnqueue(Q, h, p, i + 1);
28         else
29             TryEnqueue(Q, h, p, i + 1);
30             TryEnqueue(Q, h, p', i');
31         end
32     else TryEnqueue(Q, h, p, i + 1)
33     if i = 3 then j ← j + 1
34 end
```

be preponed by the same amount as q, that is, $\Delta = \bar{s}_{q1d} - \bar{\sigma}_{q1d}$, as shown in Fig. 7.

After an activity's start time has been calculated, its actual duration is sampled from the corresponding probability distribution, yielding a realization of the respective random variable Θ_{pid}. Then an update of the associated room's earliest availability time w_r is performed, where $\mu(p)$ denotes the room patient p is assigned to and w_0 the earliest availability time of the beam resource. After this, the procedure checks whether the next activity of the successor patient $p' = \pi_{[j+1]}$ can be scheduled next from a chronological perspective. Note that an array h is used to keep track of each

Algorithm 4: Outline of procedure `SetPreparationStartTime`

In : A position index j, patient indices p and q, and vectors $\bar{s}$, $\bar{\sigma}$, w and r for planned start times, (partially) set actual start times, earliest room availabilities, and earliest patient availabilities, respectively

if $j = 1$ **then**
  $\bar{\sigma}_{p1d} \leftarrow \bar{s}_{p1d}$;
else
  if $\bar{s}_{p1d} > \bar{\sigma}_{q2d}$ *and* $\bar{\sigma}_{q2d} < \bar{s}_{q2d}$ **then**
    $\bar{\sigma}_{p1d} \leftarrow \max(\bar{s}_{p1d} - (\bar{s}_{q2d} - \bar{\sigma}_{q2d}), w_{\mu(p)}, r_p)$;
  else if $\bar{s}_{p1d} > \bar{\sigma}_{q1d}$ *and* $\bar{\sigma}_{q1d} < \bar{s}_{q1d}$ **then**
    $\bar{\sigma}_{p1d} \leftarrow \max(\bar{s}_{p1d} - (\bar{s}_{q1d} - \bar{\sigma}_{q1d}), w_{\mu(p)}, r_p)$;
  else
    $\bar{\sigma}_{p1d} \leftarrow \max(\bar{s}_{p1d}, w_{\mu(p)}, r_p)$;
  end
end

Algorithm 5: Outline of procedure `TryEnqueue`

In : A queue Q, activity pointers h, a patient index p and an activity index i

if $i \leq 3$ *and* $i \geq \mathsf{h}[p]$ **then**
  $Q.enqueue(p, i)$;
  $\mathsf{h}[p] \leftarrow \mathsf{h}[p] + 1$;
end

patient's next (unprocessed) activity. If the start time check succeeds, the successor patient's activity is added to the processing queue, immediately followed by the next activity of the current patient. Otherwise, these two activities are enqueued in reverse order. If the current patient p is the last patient in the priority list, or if all activities of the successor patient are either already scheduled or currently queued, only the current patient's next activity is enqueued (if possible). Procedure `TryEnqueue` (see Algorithm 5) accomplishes all associated checks and updates: the provided activity index i is checked for validity in the sense that it has to correspond to an unprocessed activity. Note that an activity is considered as processed as soon as it enters the queue and therefore the activity "pointer" $\mathsf{h}[p]$ for that patient is increased right after the enqueuing operation.

Figure 8 depicts the decision process of the reactive procedure using an example that allows for early starts of preparation activities. Patient P1's preparation takes less time than scheduled, so P1 starts its irradiation as early as possible. For patient P2, case 1 applies: We prepone P2's preparation activity by the same amount that P1's irradiation is preponed (here, 5 min). The same strategy applies to P3. The decision about the potential preponement of P4 is more difficult though, because by the time P4's preparation could start, P3's preparation is still ongoing. However, P3 started preparation 5 min early, so we suggest preponing P4's preparation by the same amount (Case 2). Finally, P2's exiting activity took longer than expected. Therefore P5's preparation is delayed by 5 min.

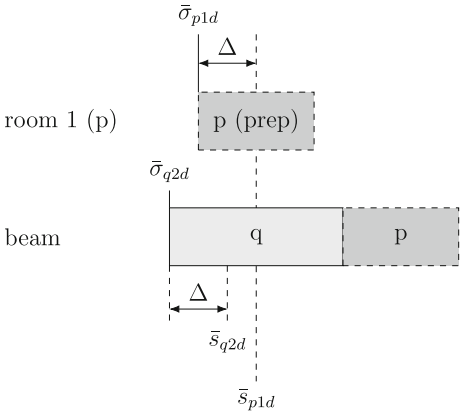

Fig. 6 Early start of a preparation activity subject to a potentially preponed predecessor beam activity

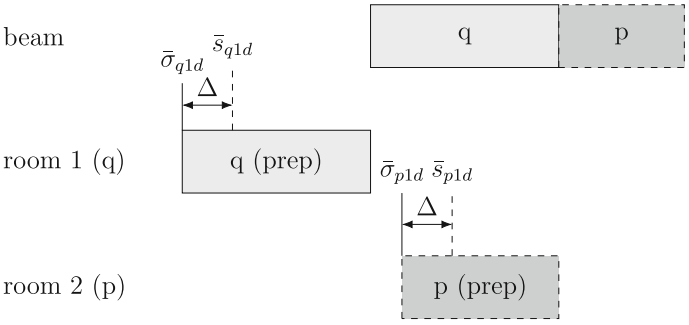

Fig. 7 Early start of a preparation activity subject to a potentially preponed predecessor preparation activity

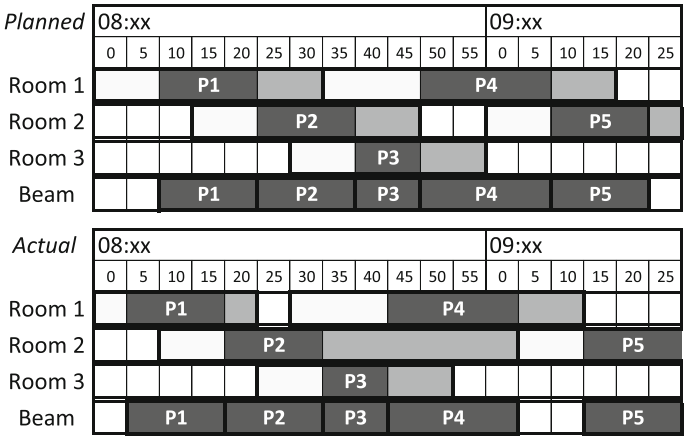

Fig. 8 Visualization of Reactive Procedure. Top: Baseline schedule including planned activity starting times. Bottom: Actual executed schedule with activity durations and starting times. White: preparation times; dark gray: irradiation times; medium gray: exiting times

In practice, preponing patients by more than 15 min is rarely possible, because they are unlikely to have arrived at the facility so early. Therefore, the sequence of patients cannot be changed without causing some waiting time or stress.

E Additional result tables

Table 11 Average results of 16 randomly generated instances and 16 replications for each instance

		Exp. objective			Beam, $\lambda_1 = 1.0$			Waiting, $\lambda_3 = 0.1$			Penalty, $\lambda_2 = 0.1$		
Pat.	β	DET	STO	WTE	DET	STO	WTE	DET	STO	WTE	DET	STO	WTE
30	0.50	2778	**2694**	2781	2396	2453	2399	3817	2392	3812	6	16	7
30	0.60	2676	**2657**	2674	2419	2466	2419	2557	1904	2554	7	9	5
30	0.66	2633	**2635**	2631	2445	2474	2447	1872	1597	1839	10	9	7
30	0.68	2623	**2630**	2622	2459	2484	2459	1638	1443	1623	5	9	8
30	0.70	2619	2627	**2616**	2474	2497	2473	1439	1295	1425	9	10	8
30	0.80	**2678**	2700	2681	2610	2634	2612	657	626	649	27	28	33
30	0.90	**3049**	3077	3051	2996	3025	2999	317	305	314	214	208	208
50	0.50	4709	**4315**	4704	3809	3890	3808	8984	4200	8934	19	57	18
50	0.60	4396	**4259**	4391	3827	3919	3831	5660	3348	5575	32	49	27
50	0.66	4247	**4228**	4248	3861	3940	3867	3814	2818	3767	53	62	43
50	0.68	4221	4220	**4216**	3888	3956	3890	3275	2571	3210	54	69	49
50	0.70	4195	4212	**4192**	3914	3978	3915	2750	2266	2717	61	80	60
50	0.80	**4273**	4320	4276	4152	4204	4156	1057	1009	1047	155	150	149
50	0.90	**4908**	4988	4917	4794	4873	4802	475	450	471	674	693	672
70	0.50	7125	**6146**	7089	5427	5534	5452	16968	5983	16358	6	133	7
70	0.60	6463	**6049**	6449	5449	5578	5465	10120	4631	9825	16	81	18
70	0.66	6156	**6020**	6144	5512	5615	5511	6408	4000	6306	24	52	23
70	0.68	6082	**6022**	6067	5544	5640	5542	5341	3767	5211	35	50	33
70	0.70	6023	**6009**	6013	5579	5664	5579	4402	3396	4309	37	57	38
70	0.80	**6127**	6198	6134	5964	6040	5973	1468	1398	1453	158	177	163
70	0.90	**7137**	7298	7150	6908	7048	6920	628	588	623	1661	1912	1671
100	0.50	10878	**8860**	10681	7891	7980	7940	29865	8570	27398	2	227	5
100	0.60	9574	**8736**	9498	7863	8053	7901	17108	6739	15964	10	89	10
100	0.66	8965	**8648**	8944	7926	8086	7952	10374	5570	9904	14	54	14
100	0.68	8825	**8671**	8815	7990	8139	8013	8327	5282	8000	19	43	18
100	0.70	8722	**8664**	8711	8055	8181	8068	6644	4776	6403	27	52	32
100	0.80	**8883**	8975	8892	8646	8733	8658	2032	1951	2012	345	461	322
100	0.90	**10624**	10939	10645	10002	10169	10019	843	798	838	5370	6905	5421

$\lambda_3 = 0.1$, optimization time limit of 1 h. Bold numbers represent the best solutions per line. Lines with optimal buffer parameters are in gray

Table 12 Average results of 16 randomly generated instances and 16 replications for each instance

		Exp. objective			Beam, $\lambda_1 = 1.0$			Waiting, $\lambda_3 = 1.0$			Penalty, $\lambda_2 = 1.0$		
Pat.	β	DET	STO	WTE	DET	STO	WTE	DET	STO	WTE	DET	STO	WTE
30	0.50	6209	**4265**	5866	2399	2616	2696	3809	1636	3167	0	13	2
30	0.60	4983	**3874**	4838	2422	2654	2660	2561	1211	2177	0	9	1
30	0.70	3915	**3509**	3889	2478	2679	2557	1437	826	1332	0	4	0
30	0.78	3356	**3296**	3338	2580	2696	2583	774	596	753	2	4	2
30	0.80	3275	**3252**	3262	2616	2701	2618	654	545	639	5	7	5
30	0.83	3220	3220	**3212**	2698	2756	2700	510	450	498	12	15	14
30	0.90	3456	3483	**3451**	3026	3068	3029	310	284	304	120	130	118
50	0.50	12743	**7207**	11239	3815	4109	4484	8927	3031	6746	1	68	9
50	0.60	9520	**6376**	8617	3846	4186	4455	5670	2141	4151	5	49	11
50	0.70	6684	**5716**	6529	3929	4247	4176	2725	1423	2323	29	46	30
50	0.78	5453	**5374**	5412	4115	4300	4137	1258	984	1195	80	90	80
50	0.80	5327	5306	**5298**	4178	4310	4185	1042	886	1009	106	110	105
50	0.83	5263	5296	**5250**	4304	4416	4316	801	715	775	158	166	158
50	0.90	5803	5895	**5789**	4851	4938	4852	462	428	456	490	529	481
70	0.50	22369	**10605**	18078	5439	5813	6376	16928	4619	11664	1	173	38
70	0.60	15490	**9150**	13232	5467	5925	6341	10020	3132	6872	4	93	19
70	0.70	9941	**8185**	9531	5602	6037	6106	4325	2096	3410	14	51	15
70	0.78	7729	**7632**	7682	5918	6147	5949	1757	1409	1676	55	76	57
70	0.80	7529	7530	**7488**	5999	6192	6017	1448	1229	1392	83	110	79
70	0.83	7419	7508	**7398**	6199	6355	6210	1082	975	1050	138	178	138
70	0.90	**8861**	9334	8884	6978	7129	6998	611	563	601	1273	1642	1286
100	0.50	37421	**15382**	28795	7899	8372	9071	29522	6778	19666	0	232	58
100	0.60	24600	**13428**	20141	7898	8532	9078	16702	4782	11031	0	113	32
100	0.70	14576	**11801**	13699	8123	8681	8844	6451	3083	4849	2	37	6
100	0.78	11125	**11055**	11024	8594	8887	8660	2440	1971	2281	91	197	82
100	0.80	10880	10989	**10830**	8712	8946	8746	1989	1727	1905	179	316	180
100	0.83	10871	11142	**10852**	9019	9193	9035	1469	1357	1428	384	592	389
100	0.90	**15607**	17397	15783	10088	10275	10111	828	780	819	4692	6342	4853

$\lambda_3 = 1.0$, optimization time limit of 1 h. Bold numbers represent the best solutions per line. Lines with optimal buffer parameters are in gray

Table 13 Wilcoxon–Mann–Whitney test to pairwise compare methods STO, DET and WTE for 16 random instances with 100 patients

		STO vs. DET			STO vs. WTE			WTE vs. DET		
λ_3	β	STO	DET	Equal	STO	WTE	Equal	WTE	DET	Equal
0.1										
	0.50	16	0	0	16	0	0	12	0	4
	0.60	16	0	0	16	0	0	7	0	9
	0.66	16	0	0	16	0	0	2	0	14
	0.68	15	0	1	14	0	2	1	0	15
	0.70	10	0	6	10	0	6	2	0	14
	0.78	0	13	3	0	11	5	0	1	15
	0.80	0	13	3	0	13	3	0	2	14
	0.83	0	15	1	0	14	2	0	0	16
	0.90	0	16	0	0	16	0	0	2	14
0.5										
	0.50	16	0	0	16	0	0	12	0	4
	0.60	16	0	0	16	0	0	14	0	2
	0.66	16	0	0	16	0	0	13	0	3
	0.68	16	0	0	16	0	0	13	0	3
	0.70	16	0	0	16	0	0	11	0	5
	0.78	1	5	10	1	9	6	4	1	11
	0.80	0	14	2	0	14	2	1	1	14
	0.83	0	16	0	0	16	0	0	0	16
	0.90	0	16	0	0	16	0	0	0	16
1.0										
	0.50	16	0	0	16	0	0	15	0	1
	0.60	16	0	0	16	0	0	15	0	1
	0.66	16	0	0	16	0	0	16	0	0
	0.68	16	0	0	16	0	0	15	0	1
	0.70	16	0	0	16	0	0	14	0	2
	0.78	14	2	0	10	2	4	12	0	4
	0.80	8	4	4	1	8	7	12	0	4
	0.83	0	12	4	0	16	0	8	0	8
	0.90	0	16	0	0	16	0	0	1	15

16 replications per run. Significance level is $\alpha = 0.05$. Entries show the number of instances where either of the two pairwise compared methods is significantly better than the other method. Column "equal" lists the number of instances where neither of the methods performed better than the other. Lines with optimal buffer parameters in gray

References

Affenzeller M, Wagner S (2004) SASEGASA: a new generic parallel evolutionary algorithm for achieving highest quality results. J Heuristics 10(3):243–267. https://doi.org/10.1023/B:HEUR.0000026895.72657.a2

Ahmadi-Javid A, Jalali Z, Klassen KJ (2017) Outpatient appointment systems in healthcare: a review of optimization studies. Eur J Oper Res 258(1):3–34. https://doi.org/10.1016/j.ejor.2016.06.064

Begen MA, Levi R, Queyranne M (2012) Technical note—a sampling-based approach to appointment scheduling. Oper Res 60(3):675–681. https://doi.org/10.1287/opre.1120.1053

Belien J, Demeulemeester E (2004) Integer programming for building robust surgery schedules. Tech. rep., Katholieke Universiteit Leuven, Department of Applied Economics, Research Report OR0446

Berg BP, Denton BT, Ayca Erdogan S, Rohleder T, Huschka T (2014) Optimal booking and scheduling in outpatient procedure centers. Comput Oper Res 50:24–37. https://doi.org/10.1016/j.cor.2014.04.007

Bianchi L, Dorigo M, Gambardella LM, Gutjahr WJ (2009) A survey on metaheuristics for stochastic combinatorial optimization. Nat Comput 8(2):239–287. https://doi.org/10.1007/s11047-008-9098-4

Cardoen B, Demeulemeester E, Beliën J (2010) Operating room planning and scheduling: a literature review. Eur J Oper Res 201(3):921–932. https://doi.org/10.1016/j.ejor.2009.04.011

Davari M, Demeulemeester E (2017) The proactive and reactive resource-constrained project scheduling problem. J Sched. https://doi.org/10.1007/s10951-017-0553-x First Online: 1–27

Demeulemeester E, Herroelen W, Leus R (2008) Proactive-reactive project scheduling. In: Artigues C, Demassey S, Neron E (eds) Resource-constrained project scheduling: models, algorithms, extensions and applications, vol 2. Wiley, London, pp 203–212. https://doi.org/10.1002/9780470611227.ch13

Deng Y, Shen S, Denton B (2019) Chance-constrained surgery planning under conditions of limited and ambiguous data. INFORMS J Comput 31(3):559–575. https://doi.org/10.1287/ijoc.2018.0835

Denton B, Viapiano J, Vogl A (2007) Optimization of surgery sequencing and scheduling decisions under uncertainty. Health Care Manag Sci 10:13–24. https://doi.org/10.1007/s10729-006-9005-4

Erdogan SA, Denton B (2013) Dynamic appointment scheduling of a stochastic server with uncertain demand. INFORMS J Comput 25(1):116–132. https://doi.org/10.1287/ijoc.1110.0482

Geletu A, Klöppel M, Zhang H, Li P (2013) Advances and applications of chance-constrained approaches to systems optimisation under uncertainty. Int J Syst Sci 44(7):1209–1232. https://doi.org/10.1080/00207721.2012.670310

Gupta D, Denton B (2008) Appointment scheduling in health care: challenges and opportunities. IIE Trans 40:800–819. https://doi.org/10.1080/07408170802165880

Hans E, Wullink G, Houdenhoven MV, Kazemier G (2008) Robust surgery loading. Eur J Oper Res 185:1038–1050. https://doi.org/10.1016/j.ejor.2006.08.022

Herroelen W, Leus R (2005) Project scheduling under uncertainty: survey and research potentials. Eur J Oper Res 165(2):289–306. https://doi.org/10.1016/j.ejor.2004.04.002

Juan AA, Faulin J, Grasman SE, Rabe M, Figueira G (2015) A review of simheuristics: extending metaheuristics to deal with stochastic combinatorial optimization problems. Oper Res Perspect 2:62–72. https://doi.org/10.1016/j.orp.2015.03.001

Kaandorp GC, Koole G (2007) Optimal outpatient appointment scheduling. Health Care Manag Sci 10(3):217–229. https://doi.org/10.1007/s10729-007-9015-x

Kemper B, Klaassen CAJ, Mandjes M (2014) Optimized appointment scheduling. Eur J Oper Res 239(1):243–255. https://doi.org/10.1016/j.ejor.2014.05.027

King M (2017) Statistics for process control engineers: a practical approach. Wiley, Hoboken. https://doi.org/10.1002/9781119383536

Kleywegt AJ, Shapiro A, Homem-de Mello T (2002) The sample average approximation method for stochastic discrete optimization. SIAM J Optim 12(2):479–502. https://doi.org/10.1137/S1052623499363220

Koeleman PM, Koole GM (2012) Optimal outpatient appointment scheduling with emergency arrivals and general service times. IIE Trans Healthc Syst Eng 2(1):14–30. https://doi.org/10.1080/19488300.2012.665154

Kreitz TM, Winters BS, Pedowitz DI (2016) The influence of wait time on patient satisfaction in the orthopedic clinic. J Patient Exp 3(2):39–42. https://doi.org/10.1177/2374373516652253

Legrain A, Fortin MA, Lahrichi N, Rousseau LM (2015) Online stochastic optimization of radiotherapy patient scheduling. Health Care Manag Sci 18:110–123. https://doi.org/10.1007/s10729-014-9270-6

Mancilla C, Storer R (2012) A sample average approximation approach to stochastic appointment sequencing and scheduling. IIE Trans 44(8):655–670. https://doi.org/10.1080/0740817X.2011.635174

Maschler J, Hackl T, Riedler M, Raidl GR (2017a) An enhanced iterated greedy metaheuristic for the particle therapy patient scheduling problem. In: Duarte A, Viana A, Angel J, Mélian B, Ramalhinho H (eds.), Proceedings of the MIC and MAEB 2017 conferences, Barcelona, pp 465–474

Maschler J, Riedler M, Raidl GR (2017b) Particle therapy patient scheduling: Time estimation for scheduling sets of treatments. In: Moreno-Díaz R, Pichler F, Quesada-Arencibia A (eds.), Computer Aided Systems Theory—EUROCAST 2017, Springer, Switzerland, pp 364–372. https://doi.org/10.1007/978-3-319-74718-7_44

Möhring RH, Stork F (2000) Linear preselective policies for stochastic project scheduling. Math Methods Oper Res 52(3):501–515. https://doi.org/10.1007/s001860000095

Petrovic D, Morshed M, Petrovic S (2009) Genetic algorithm based scheduling of radiotherapy treatments for cancer patients. Proc Conf Artif Intell Med 5651:101–105

Petrovic S, Castro E (2011) A genetic algorithm for radiotherapy pre-treatment scheduling. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 6625 LNCS(PART 2):454–463. https://doi.org/10.1007/978-3-642-20520-0_46

Petrovic S, Leite-Rocha P (2008) Constructive and GRASP approaches to radiotherapy treatment scheduling. In: Proceedings of the advances in electrical and electronics engineering—IAENG special edition of the world congress on engineering and computer science 2008, WCECS 2008, pp 192–200. https://doi.org/10.1109/WCECS.2008.31

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

Samudra M, Van Riet C, Demeulemeester E, Vansteenkiste N, Rademakers F (2016) Scheduling operating rooms: achievements, challenges and pitfalls. J Sched 5:493–525. https://doi.org/10.1007/s10951-016-0489-6

Sauré A, Patrick J, Tyldesley S, Puterman ML (2012) Dynamic multi-appointment patient scheduling for radiation therapy. Eur J Oper Res 223:573–584. https://doi.org/10.1016/j.ejor.2012.06.046

Shylo OV, Prokopyev OA, Schaefer AJ (2013) Stochastic operating room scheduling for high-volume specialties under block booking. INFORMS J Comput 25(4):682–692. https://doi.org/10.1287/ijoc.1120.0530

Tancrez JS, Roland B, Cordier JP, Riane F (2013) Assessing the impact of stochasticity for operating theater sizing. Decis Support Syst 55:616–628. https://doi.org/10.1016/j.dss.2012.10.021

Van De Vonder S, Demeulemeester E, Herroelen W, Leus R (2005) The use of buffers in project management: the trade-off between stability and makespan. Int J Prod Econ 97(2):227–240. https://doi.org/10.1016/j.ijpe.2004.08.004

Van De Vonder S, Demeulemeester E, Leus R, Herroelen W (2006) Proactive-reactive project scheduling—trade-offs and procedures. In: Jozefowska J, Weglarz J (eds) Perspectives in modern project scheduling, chap 2. Springer, New York, pp 25–53. https://doi.org/10.1007/978-0-387-33768-5_2

Van De Vonder S, Demeulemeester E, Herroelen W (2007) A classification of predictive–reactive project scheduling procedures. J Sched 10(3):195–207. https://doi.org/10.1007/s10951-007-0011-2

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):1–11. https://doi.org/10.1186/s12911-016-0390-4

Vogl P, Braune R, Doerner KF (2018a) A multi-encoded genetic algorithm approach to scheduling recurring radiotherapy treatment activities with alternative resources, optional activities, and time window constraints. Comput Aided Syst Theory EUROCAST 2017 10671(1):373–382. https://doi.org/10.1007/978-3-319-74718-7_45

Vogl P, Braune R, Doerner KF (2018b) Scheduling recurring radiotherapy appointments in an ion beam facility: considering optional activities and time window constraints. J Sched. https://doi.org/10.1007/s10951-018-0574-0

Wild C, Weiderpass E, Stewart B (eds.) (2020) World Cancer Report: Cancer Research for Cancer Prevention. http://publications.iarc.fr/586