Particle Therapy Patient Scheduling: First Heuristic Approaches

**Johannes Maschler · Martin Riedler ·
Markus Stock · Günther R. Raidl**

Abstract The Particle Therapy Patient Scheduling Problem arises in radiotherapy used for cancer treatment. Previous contributions in the existing literature primarily dealt with photon and electron therapy with a one-to-one correspondence of treatment rooms and accelerators. In particle therapy, however, a single accelerator serves multiple rooms in an interleaved way. This leads to a novel scenario in which the main challenge is to utilize the particle beam as well as possible. Switching between rooms allows to reduce idle time of the beam that emerges as a consequence of preparation steps.

In this work we present first algorithms for solving this problem. In particular, we address the midterm planning variant which involves a time horizon of a few months but also requires detailed scheduling within each day. We formalize the problem via a mixed integer linear programming model, which, however, turns out to be intractable in practice. Consequently, we start with a construction heuristic featuring a forward-looking mechanism. Based upon this fast method we further study a Greedy Randomized Adaptive Search Procedure as well as an Iterated Greedy metaheuristic. A computational comparison of these algorithms is performed on benchmark instances created in a way to reflect the most important aspects of a real-world scenario.

Keywords radiotherapy · particle therapy · patient scheduling · metaheuristic · Iterated Greedy · GRASP

J. Maschler, M. Riedler, G. R. Raidl
Institute of Computer Graphics and Algorithms, TU Wien, Vienna, Austria
E-mail: {maschler|riedler|raidl}@ac.tuwien.ac.at

M. Stock
EBG MedAustron GmbH, Wiener Neustadt, Austria
E-mail: markus.stock@medaustron.at

1 Introduction

The number of new cancer cases in 2012 amounted to about 14.1 Million worldwide (not including skin cancer other than melanoma). In the same year cancer caused about 14.6% of all human deaths [21], making it one of the ten most common causes of death [22]. Accordingly, a serious amount of research and investments has been contributed to provide and develop treatments. In 2010 the costs of cancer have been estimated at around USD 1.16 trillion per year [21]. A widely applied treatment option is radiotherapy. Usually linear accelerators (LINACs) are used to provide treatment in conventional external beam therapy (electron or photon therapy). More recently the option to treat with particle beams (like protons or carbons) has shown promising results and the number of patients treated worldwide increases rapidly. Nevertheless, the investment costs of such centers is high compared to conventional centers. It is therefore especially important to utilize available resources for particle therapy centers as efficiently as possible and to maximize the number of treated patients over time.

In typical photon and electron radiotherapy it is common that a single LINAC serves a dedicated room exclusively. In contrast to LINACs, particle beams are produced by either cyclotrons or synchrotrons which can serve two to five treatment rooms sequentially. Therefore, in particle therapy the beam can be considered as the bottleneck. Due to the time required to prepare patients in one room until their irradiation starts and the time needed until other patients exit the room after irradiations, undesirable idle times on the beam emerge. We consider here more specifically the radiotherapy treatment center MedAustron[1] located in Wiener Neustadt, Austria. This emerging facility is currently one of the most modern of its kind. In Wiener Neustadt proton and carbon beams are being produced by a synchrotron serving four different rooms, three of which are used for treatment. For every treatment preparation of the patient is necessary. To avoid undesirable idle times this preparation is performed in parallel in all available rooms. This helps to increase the treatment throughput. However, accomplishing this goal is far from trivial and requires elaborate scheduling techniques. The tasks executed in the context of irradiation need to be arranged in the most efficient manner while respecting several particular side-constraints and resource dependencies.

We consider here a first, simplified problem formulation for addressing the midterm planning part of the Particle Therapy Patient Scheduling Problem (PTPSP), in which an effective plan has to be found for performing a larger number of therapies over the next few months. A particle therapy at MedAustron will typically involve from 8 up to 35 treatments provided on subsequent days. We refer to them as daily treatments (DTs). Each therapy is specified by an earliest as well as a latest day at which it can be started. The number of DTs provided each week needs to stay within a lower and an upper bound. Similarly, the number of days that passes between two consecutive

[1] `http://www.medaustron.at/`

treatments is also restricted. Between two weeks each patient requires a break of at least two days. Each DT involves several steps such as the preparation, the actual irradiation, and the follow-up. We refer to these individual steps as activities. A DT consists of either five or seven activities depending on whether sedation of the patient is required. All activities belonging to the same DT are executed consecutively. Activities have associated (expected) processing times. Additionally, lower and upper time limits on the duration from the end of an activity to the start of some following activities may be given. Moreover, activities require for their execution several resources (e.g., particle beam, room, oncologist, anesthetist). Each resource has a regular availability period on each day it can be used. Beyond each regular availability period, a resource can be used for a not further restricted *extended time* at additional costs. Generally, all resources might be unavailable for smaller time intervals specified by a list of unavailability periods.

The objective we consider in this work is to schedule a given set of therapies s.t. all operational constraints are fulfilled, costs for used extended times of resources are minimized, and all treatments are finished as soon as possible.

1.1 Our contribution

After giving an overview on related work in the next section, we formally define the problem in Section 3 by means of a Mixed Integer Linear Programming (MILP) model. This model, however, is intractable in practice, and consequently we consider heuristics. Section 4 proposes a therapy-wise construction heuristic based on greedy principles and featuring a forward-looking mechanism to avoid too naive decisions. In Section 5 we further build upon this construction heuristic, proposing a Greedy Randomized Adaptive Search Procedure (GRASP) and an Iterated Greedy (IG) metaheuristic. Computational results are presented in Section 6. While the construction heuristic on its own already produces reasonable solutions, the IG method yields substantially better results, also outperforms GRASP, and scales well to instances of practically relevant size. Section 7 concludes this work with remarks on future work.

2 Related Work

Compared to other scheduling problems there is not much material available on radiotherapy scheduling. A first attempt at automating this task has been made in 1993 by Larsson [9]. Then, for more than a decade no further contributions appeared. In 2006 interest in the topic grew again starting with the works of Kapamara et al. [8] and Petrovic et al. [17]. Afterwards, several heuristic as well as exact approaches followed. Heuristic techniques reach from GRASP [16] and steepest hill climbing methods [7] to more advanced techniques using Genetic Algorithms (GAs) [14,15]. Exact approaches consider different levels of granularity and rely on MILP models [3,4,5,2]. Additionally, there are two theses available dealing with the topic [13,11].

The latest contributions focus on dynamic scenarios. Sauré et al. [20] consider a discounted infinite-horizon Markov decision process and solve it via Linear Programming with Column Generation to identify good policies for allocating available treatment capacity to incoming demand. Legrain et al. [10] introduce a hybrid method combining stochastic optimization and online optimization to determine better planning strategies. They also consider information on the future arrivals of patients to better estimate the expected resource utilization.

All the references mentioned above consider a coarse scheduling scenario, i.e., they only assign treatments to days, but do not deal with sequencing within a day. This is due to the fact that the addressed practical applications feature radiotherapy with photons or electrons[2]. In these scenarios multiple LINACs are available, but for each of them only sequential processing is possible. Thus, the main issue is to assign treatments to appropriate machines. Therefore, there is no immediate need for fine grained scheduling in these scenarios. The application we consider substantially differs in this respect since the availability of just a single accelerator whose beam can be directed to only one room at a time demands much more detailed planning to reach maximum throughput. Moreover, in the more widely applied photon and electron radiotherapy it is common to have long waiting lists associated with priorities determined by oncologists and algorithms need to select from that list when inserting new patients. In our scenario this is not required since the accepted patients must always be determined by physicians.

The Resource-Constrained Project Scheduling Problem (RCPSP) is a vast research area with many variants [6]. In principle the PTPSP can be viewed as highly specialized case of an RCPSP. To consider all our requirements we need – in project scheduling terminology – release times and deadlines, minimum and maximum time lags, disjunctive resources with availability varying over time, and overflow periods; for an explanation of these aspects see [1]. Although each of these facets has been addressed in the literature, to our best knowledge no work exists considering all of them together. Moreover, project scheduling focuses primarily on activities, i.e., there is no further level of granularity comparable to DTs. Of course, our objective also differs significantly due to the different domain. Research on the RCPSP can provide ideas for dealing with certain aspects in radiotherapy patient scheduling, but unfortunately none of the existing RCPSP variants is close enough to directly build upon it.

[2] Men [13] considers proton therapy with a single particle beam and several rooms but also schedules only on the coarse level, i.e., entire DTs and not individual activities.

3 Problem Formalization

In the PTPSP a set of therapies $T = \{1, \ldots, n_T\}$ needs to be scheduled on consecutive days $D = \{1, \ldots, n_D\}$ considering a set of renewable resources $R = \{1, \ldots, n_R\}$.

Each therapy $t \in T$ is associated with a set of DTs $U_t = \{1, \ldots, \tau_t\}$, and each DT $u \in U_t$ is associated with a set of sequential activities $A_{t,u} = \{1, \ldots, \alpha_{t,u}\}$. For each therapy $t \in T$ we are given a minimal number n_t^{twmin} and a maximal number n_t^{twmax} of DTs that need to be performed per week, as well as a minimal number $\delta_t^{\min} \geq 1$ and a maximal number $\delta_t^{\max} \geq \delta_t^{\min}$ of days that must separate two consecutive DTs. For each DT $u \in U_t$, we are further given earliest and latest starting days $d_{t,u}^{\min}$, $d_{t,u}^{\max} \in D$, respectively. Each activity $a \in A_{t,u}$ is associated with a processing time $p_{t,u,a} \geq 0$ and requires a set of resources $Q_{t,u,a} \subseteq R$ for its execution. Note that for each DT, its activities must be scheduled strictly sequentially in the given order, but not necessarily without breaks in-between consecutive activities. Each activity is non-preemptive. Moreover, a set of end-to-start (EtS) precedence constraints $P_{t,u}^{\min} \subseteq A_{t,u} \times A_{t,u}$ among activities of DTs $u \in U_t$ with associated minimum time lags $L_{t,u,a,a'}^{\min} \geq 0$ is given for each DT. One such constraint enforces that activity a fully precedes activity a' and the starting time of activity a' and the ending time of activity a differ at least by $L_{t,u,a,a'}^{\min}$. Correspondingly, there is also a set $P_{t,u}^{\max} \subseteq A_{t,u} \times A_{t,u}$ among activities of DTs $u \in U_t$ with associated maximum time lags $L_{t,u,a,a'}^{\max} \geq 0$ given for each DT.

Let $D' \subseteq D$ denote the subset of working days on which the treatment center is actually open and DTs can be scheduled on. By $V = \{1, \ldots, n_V\}$ we refer to the set of weeks w.r.t. the considered planning horizon, and let $\bigcup_{v \in V} D'_v$ be the partitioning of D' into n_V subsets corresponding to the n_V weeks. For $d \in D'$, let $\widetilde{W}_d = [\widetilde{W}_d^{\text{start}}, \widetilde{W}_d^{\text{end}})$ be the fundamental opening time, i.e., the time window in which any activity must be scheduled – including enough practically unrestricted extended time outside of the regular business hours.

Each resource $r \in R$ is available on a subset $D_r^{\text{res}} \subseteq D'$ of the working days. On each of these days a resource is associated with a single regular service window (time interval) $W_{r,d} = [W_{r,d}^{\text{start}}, W_{r,d}^{\text{end}}] \subseteq \widetilde{W}_d$ where $W_{r,d}^{\text{start}} \leq W_{r,d}^{\text{end}}$ are the start and end times, respectively, and an immediately following extended service window $\widehat{W}_{r,d} = [W_{r,d}^{\text{end}}, \widetilde{W}_d] \subseteq \widetilde{W}_d$ running until the end of the day's fundamental opening time. Furthermore, for each resource $r \in R$ and each day $d \in D_r^{\text{res}}$, we are given unavailability periods $\overline{W}_{r,d} = \bigcup_{w=1,\ldots,\omega_{r,d}} \overline{W}_{r,d,w}$ with $\overline{W}_{r,d,w} = [\overline{W}_{r,d,w}^{\text{start}}, \overline{W}_{r,d,w}^{\text{end}}] \subset W_{r,d} \cup \widehat{W}_{r,d},\ w = 1, \ldots, \omega_{r,d}$, where $\overline{W}_{r,d,w}^{\text{start}}$ and $\overline{W}_{r,d,w}^{\text{end}}$ denote the start and end times of the w-th unavailability period. All these periods are assumed to be non-overlapping, and sorted according to increasing time.

A solution to the PTPSP is a tuple (Z, S), with $Z = \{Z_{t,u} \in D : t \in T,\ u \in U_t\}$ denoting the days at which the DTs are scheduled and $S = \{S_{t,u,a} \geq 0 : t \in T,\ u \in U_t,\ a \in A_{t,u}\}$ denoting the starting times of the DT's individual activities at the respective days. Such a solution is feasible if all domain restrictions, resource availabilities, precedence relations, lag constraints, and the remaining operational constraints are respected.

A fundamental assumption is that it is in practice not difficult to find any feasible solution – we expect "enough" extended time of all the resources to be available. The background here is that once a patient is accepted for treatment it is ensured by all possible means that his or her treatment will take place according to all the defined requirements. What we aim for is to minimize the required extended time over all resources R while finishing each treatment as early as possible. These two goals are considered in a single objective function by linearly combining corresponding terms using scalar weights γ^{ext} and γ^{finish}, respectively.

In the following we give a formal definition of the PTPSP by providing a MILP model. To this end, we first focus on the upper level of assigning DTs to days, and only afterwards consider the assignment of times to the individual activities by an extension. Note, however, that these two levels are not independent.

3.1 Day Assignment

The MILP model for the day assignment uses binary variables $x_{t,u,d}$ that are one if DT u of therapy t is to be performed on day d, i.e., $Z_{t,u} = d$, and zero otherwise, binary variables $y_{t,v}$ that are one if at least one of the DTs of therapy t is provided in week v and zero otherwise, and variables $\eta_{r,d}$ that give the amount of extended time used from resource r on day d. The latter clearly depends on the solution of the timing subproblems.

Here, at the day assignment level, we relax the specificities of the activity timings and consider only aggregated resource consumptions for each DT as well as the total resource availabilities on each day (both per resource). We denote by $q_{t,u,r} = \sum_{a \in \{A_{t,u} : r \in Q_{t,u,a}\}} p_{t,u,a}$ the total time that DT u of therapy t requires resource r. Let $h_{r,d}$ denote the aggregated regular availability of resource r on day d and $\widehat{h}_{r,d}$ the available extended time, i.e., the total time available on that day is $h_{r,d} + \widehat{h}_{r,d}$.

$$\begin{aligned}
\min\ & \gamma^{\text{ext}} \sum_{r \in R} \sum_{d \in D_r^{\text{res}}} \eta_{r,d} + \\
& \gamma^{\text{finish}} \sum_{t \in T} \left(\sum_{d \in D'} d x_{t,\tau_t,d} - Z^{\text{earliest}}_{t,\tau_t} \right) && && (1) \\
\text{s.t.}\ & \sum_{d \in D'} d x_{t,u+1,d} - \sum_{d \in D'} d x_{t,u,d} \geq \delta_t^{\min} && \forall t \in T,\ \forall u \in U_t \setminus \{\tau_t\} && (2) \\
& \sum_{d \in D'} d x_{t,u+1,d} - \sum_{d \in D'} d x_{t,u,d} \leq \delta_t^{\max} && \forall t \in T,\ \forall u \in U_t \setminus \{\tau_t\} && (3) \\
& \sum_{d \in D'_v} \sum_{u \in U_t} x_{t,u,d} \geq \min(n_t^{\text{twmin}}, |D'_v|) \cdot && \forall t \in T,\ \forall v \in V \setminus \{n_V\} && (4) \\
& \qquad (y_{t,v} + y_{t,v+1} - 1)
\end{aligned}$$

$$\sum_{d \in D'_v} \sum_{u \in U_t} x_{t,u,d} \leq n_t^{\text{twmax}} y_{t,v} \qquad \forall t \in T,\ \forall v \in V \tag{5}$$

$$x_{t,u,d} + x_{t,u,d'} \leq 1 \qquad \begin{aligned} &\forall t \in T,\ \forall v \in V \setminus \{n_V\}, \\ &\forall d, d' \in D' : d \in \max\{D'_v\}, \\ &d' \in \min\{D'_{v+1}\}, d' - d = 2 \end{aligned} \tag{6}$$

$$\sum_{t \in T} \sum_{u \in U_t} q_{t,u,r} x_{t,u,d} \leq h_{r,d} + \eta_{r,d} \qquad \forall r \in R,\ \forall d \in D_r^{\text{res}} \tag{7}$$

$$\sum_{d \in D'} x_{t,u,d} = 1 \qquad \forall t \in T,\ \forall u \in U_t \tag{8}$$

$$x_{t,u,d} \leq y_{t,v} \qquad \begin{aligned} &\forall t \in T,\ \forall u \in U_t, \\ &\forall v \in V,\ \forall d \in D'_v \end{aligned} \tag{9}$$

$$x_{t,u,d} = 0 \qquad \begin{aligned} &\forall t \in T,\ \forall u \in U_t, \\ &\forall d \in D' : d \notin [d_{t,u}^{\min}, d_{t,u}^{\max}] \end{aligned} \tag{10}$$

$$0 \leq \eta_{r,d} \leq \widehat{h}_{r,d} \qquad \forall r \in R,\ \forall d \in D_r^{\text{res}} \tag{11}$$

$$x_{t,u,d} \in \{0,1\} \qquad \forall t \in T,\ \forall u \in U_t,\ \forall d \in D' \tag{12}$$

$$y_{t,v} \in \{0,1\} \qquad \forall t \in T,\ \forall v \in V \tag{13}$$

Objective function (1) minimizes the use of extended time and prioritizes early finishing days. The formula involves a bound on the earliest possible finishing day:

$$Z_{t,\tau_t}^{\text{earliest}} = \begin{cases} d_{t,1}^{\min} + \left(\left\lceil \frac{\tau_t}{n_t^{\text{twmax}}} \right\rceil - 1 \right) (7 - n_t^{\text{twmax}}) + (\tau_t - 1) & \text{if } \delta_t^{\min} = 1 \\ d_{t,1}^{\min} + (\tau_t - 1)\delta_t^{\min} & \text{otherwise.} \end{cases} \tag{14}$$

We then only consider the number of days a treatment is finished later than $Z_{t,\tau_t}^{\text{earliest}}$. Inequalities (2) enforce that all the DTs of a therapy t are scheduled in the correct order and that the minimal number of required days $\delta_t^{\min}$ between two consecutive DTs is adhered. Similarly, inequalities (3) take care that the consecutive DTs of a therapy are scheduled no more than $\delta_t^{\max}$ days apart. The following two sets of constraints ensure for each therapy that the number of planned DTs stays within n_t^{twmin} and n_t^{twmax} per week, except for the last week where the number might be lower. Inequalities (6) require that if on a Saturday and on the following Monday DTs can be scheduled then treatments of the same therapy may be scheduled on at most one of these days. This guarantees the required break of at least two days between weeks[3]. Subsequent inequalities (7) enforce that the amount of consumed resources does not exceed the amount of available resources on any day. Equations (8) assure that each DT is performed exactly once. Inequalities (9) link the x variables with the y variables. The domains of the DTs' days are restricted by (10), simply fixing invalid assignments to zero. Inequalities (11) restrict the extended time that might be used. (12) specifies the domain of the x variables and (13) that of the y variables.

[3] The facility is assumed to be always closed on Sundays.

Again, be reminded that we relaxed all activity timing aspects in the above model and so far only considered aggregated resource consumptions and availabilities. Thus, this model on its own only provides a lower bound for the optimal solution value for the whole PTPSP and day assignments that might be useful in a heuristic context.

3.2 Time Assignment

Let us now extend the above model by exactly modeling also the time assignments for the DT's activities and calculating $\eta_{r,d}$ respectively. We formulate an individual set of additional constraints for each working day $d \in D'$. To ease the notation, let $U'_d = \{(t,u) : t \in T, u \in U_t, x_{t,u,d} = 1\}$ be the set of DTs and $A'_d = \{(t,u,a) : (t,u) \in U'_d, a \in A_{t,u}\}$ the set of activities that have been assigned to day d. Moreover, we define $S^{\mathrm{D}}_{t,u,a,d}$ to be the set of all (integral) feasible starting times on day d allowing the corresponding activity to be executed without overlapping with one of the unavailability periods of its required resources.

The extension is stated in terms of binary variables $x_{t,u,a,k} \in S^D_{t,u,a,d}$ that are one if activity $a \in A_{t,u}$ starts at time point k, i.e., $S_{t,u,a} = k$, and zero otherwise.

For all $d \in D'$:

$$\sum_{k \in S^D_{t,u,a,d}} x_{t,u,a,k} = 1 \qquad \forall (t,u,a) \in A'_d \quad (15)$$

$$\sum_{k \in S^D_{t,u,a,d}} k x_{t,u,a,k} + p_{t,u,a} \leq \sum_{k \in S^D_{t,u,a+1,d}} k x_{t,u,a+1,k'} \qquad \forall (t,u) \in U'_d,\ \forall a \in A_{t,u} \setminus \{\alpha_{t,u}\} \quad (16)$$

$$\sum_{k \in S^D_{t,u,a,d}} k x_{t,u,a,k} + p_{t,u,a} + L^{\min}_{t,u,a,a'} \leq \sum_{k \in S^D_{t,u,a',d}} k x_{t,u,a',k'} \qquad \forall (t,u) \in U'_d,\ \forall (a,a') \in P^{\min}_{t,u} \quad (17)$$

$$\sum_{k \in S^D_{t,u,a,d}} k x_{t,u,a,k} + p_{t,u,a} + L^{\max}_{t,u,a,a'} \geq \sum_{k \in S^D_{t,u,a',d}} k x_{t,u,a',k'} \qquad \forall (t,u) \in U'_d,\ \forall (a,a') \in P^{\max}_{t,u} \quad (18)$$

$$\sum_{(t,u,a) \in A'_d : r \in Q_{t,u,a}} \ \sum_{\substack{k \in [b - p_{t,u,a}, b] \\ \cap S^D_{t,u,a,d}}} x_{t,u,a,k} \leq 1 \qquad \forall r \in R,\ \forall b \in (W_{r,d} \cup \widehat{W}_{r,d} \setminus \overline{W}_{r,d}) \cap \mathbb{Z}_{\geq 0} \quad (19)$$

$$\eta_{r,d} \geq \sum_{k \in S^D_{t,u,a,d}} (k\, x_{t,u,a,k} + p_{t,u,a}) - W^{\mathrm{end}}_{r,d} \qquad \forall r \in R,\ \forall (t,u,a) \in A'_d : r \in Q_{t,u,a} \quad (20)$$

$$x_{t,u,a,k} \in \{0,1\} \qquad \forall (t,u,a) \in A'_d,\ \forall k \in S^D_{t,u,a,d} \quad (21)$$

Inequalities (15) ensure that each activity is scheduled at exactly one point in time. Inequalities (16) guarantee that activities belonging to the same DT are scheduled in the correct order. The next two sets of inequalities enforce the minimum and maximum EtS time lags. Constraints (19) take care that the amounts of consumed resources never exceed the amount of available resources. Finally, inequalities (20) are used to calculate $\eta_{r,d}$, the required extended times of each resource $r \in R$, from the latest time it is in use by any activity. Remember that the sum over all $\eta_{r,d}$ appears in the objective function (1) and is to be minimized. The domains of the starting times $x_{t,u,a,k}$ of the activities are given in (21).

While we will see practical results for the MILP model of the day assignment relaxation (1)–(13), the time assignment extension was provided here only to specify the PTPSP in an exact way. Experiments very soon indicated that already solving the day assignment relaxation is computationally challenging. Solving the whole MILP model including the time assignments is clearly out of reach for the instances of practically relevant size. This would even hold already for a very crude time discretization. We will therefore focus on heuristic methods in this work.

4 Therapy-Wise Construction Heuristic

As a first, rather fast method to obtain a heuristic solution for PTPSP, we propose the Therapy-Wise Construction Heuristic (TWCH) in the following. It acts in two phases, first assigning all DTs to days and afterwards scheduling the activities on each day. The heuristic follows simple greedy principles but also uses a forward-looking mechanism to avoid getting trapped by making obviously poor decisions.

4.1 Day Assignment Phase

The heuristic iteratively selects one yet unconsidered therapy and assigns days to its DTs in a sequential manner. For each DT, all days are considered that allow a feasible allocation of the DT's activities w.r.t. aggregated resource demands and still available capacities and also admit the scheduling of the subsequent DTs at later days. A DT is then always assigned to the day with the lowest estimated cost increase w.r.t. the objective function (1).

Algorithm 1 shows this procedure in detail. It starts with a set of therapies T' to be scheduled, which initially corresponds to T, and an empty set G_d, $\forall d \in D'$, which contains the DTs scheduled on each day d. In each iteration of the while-loop a therapy t is selected and removed from T' according to a priority value determined by function therapy_priority. We will consider the calculation of this priority value later. The DTs $u \in U_t$ of the selected therapy t are then processed sequentially in the for-loop from line 6 on.

For each DT a range of feasible days $\{d^{\text{earliest}}, \dots, d^{\text{latest}}\}$ is determined first. This range is for all DTs at most $\{d^{\min}_{t,u}, \dots, d^{\max}_{t,u}\}$. For all DTs for which the predecessor has been already assigned, i.e., all DTs except the first, the range of possible days can be restricted further. Line 10 and line 12 exclude days from the range that are either to close or too far from the predecessor. Lines 11 and 13 exclude days staying in conflict with the requirement that at least n^{twmin}_t and at most n^{twmax}_t DTs need to be provided per week. Each working day in this range is then further evaluated in the inner for-loop from line 15 on. Days at which at least one resource required by the DT is not

```
1  $G_d := \emptyset \quad \forall d \in D'$;
2  while $T' \neq \emptyset$ do
3      $t := \arg\max_{t \in T'}$ therapy_priority($t$); ties are broken randomly;
4      $T' := T' \setminus \{t\}$;
5      $d^{\text{last}} := -1$; $v^{\text{last}} := -1$; $n^{\text{tw}} := 0$;
6      for $u := 1$ to $\tau_t$ do
7          $d^{\text{best}} := -1$; bestCost $:= \infty$;
8          $d^{\text{earliest}} := d^{\min}_{t,u}$; $d^{\text{latest}} := d^{\max}_{t,u}$;
9          if $u > 0$ then
10             $d^{\text{earliest}} := \max\{d^{\text{earliest}}, d^{\text{last}} + \delta^{\min}_t\}$;
11             if $n^{\text{tw}} = n^{\text{twmax}}_t$ then $d^{\text{earliest}} := \max(d^{\text{earliest}}, \max\{D'_{v^{\text{last}}}\} + 1)$;
12             $d^{\text{latest}} := \min(d^{\text{latest}}, d^{\text{last}} + \delta^{\max}_t)$;
13             if $n^{\text{tw}} < \min(n^{\text{twmin}}_t, |D'_{v^{\text{last}}}|)$ then
               $d^{\text{latest}} := \min(d^{\text{latest}}, \max\{D'_{v^{\text{last}}}\})$;
14         end
15         foreach $d \in \{d^{\text{earliest}}, \dots, d^{\text{latest}}\} \cap D'$ do
16             if $\exists r \in R : q_{t,u,r} + \sum_{(t',u') \in G_d} q_{t',u',r} > h_{r,d} + \widehat{h}_{r,d}$ then continue;
17             lookaheadCost := lookahead(); continue if infeasible;
18             extCost $:= \sum_{r \in Q_{t,u,1} \cup \dots \cup Q_{t,u,\alpha_{t,u}}} (\gamma^{\text{ext}} \cdot (\max(q_{t,u,r} +$
               $\sum_{(t',u') \in G_d} q_{t',u',r} - \ h_{r,d}, 0)) - \max(\sum_{(t',u') \in G_d} q_{t',u',r} - h_{r,d}, 0)))$;
19             finishCost $:= \gamma^{\text{finish}} \cdot (d - d^{\text{earliest}})$;
20             if lookaheadCost + extCost + finishCost < bestCost then
21                 $d^{\text{best}} := d$;
22                 bestCost := lookaheadCost + extCost + finishCost;
23             end
24         end
25         if $d^{\text{best}} = -1$ then infeasible;
26         let $v \in V$ be the week index s.t. $d \in D'_v$;
27         if $v^{\text{last}} = v$ then $n^{\text{tw}} := n^{\text{tw}} + 1$ ;
28         else $v^{\text{last}} := v$; $n^{\text{tw}} := 1$ ;
29         $d^{\text{last}} := d^{\text{best}}$;
30         $G_{d^{\text{best}}} := G_{d^{\text{best}}} \cup \{(t, u)\}$;
31     end
32 end
```

Algorithm 1: TWCH DayAssignment(T')

available in enough quantity anymore are skipped by line 16. Variables $h_{r,d}$ and $\widehat{h}_{r,d}$ denote here again the total available regular and extended time of resource r on day d.

For each remaining day the cost of assigning $u \in U_t$ to d are then estimated by calculating the sum of the costs arising from using extended service windows, a penalty cost for using later days and an estimation of the costs for assigning all successive DTs $u + 1, \dots, \tau_t$. The cost that originates from a single DT w.r.t. resource availabilities is the time the DT's activities use from extended service windows as calculated at line 18. Line 19 computes how much selecting a specific day delays the whole therapy in relation to the second term of objective function (1). Line 30 finally assigns DT u to the day with the lowest estimated cost, which is recorded in d^{best}.

To estimate the costs for assigning the successive DTs $u+1,\ldots,\tau_t$ we use a forward-looking mechanism that works almost analogously to the main algorithm's part from line 6 to line 29. The essential difference to Algorithm 1 is that in the lookahead we use for $u+1,\ldots,\tau_t$ the costs only from the first feasible day instead of considering the costs of all feasible days. This can be accomplished by omitting line 17 and terminating the for-loop from line 15 if line 22 is reached. The lookahead returns either the cost of assigning the DTs $u+1,\ldots,\tau_t$ to their first possible days, respectively, or the result that for at least one of the DTs no feasible day could be found. In the latter case we can remove the considered day from further consideration.

The performance of Algorithm 1 depends to a large extent on the function therapy_priority. In preliminary tests we considered as this priority (a) the number of DTs τ_t, (b) the latest starting day for the first DT $-d_{t,1}^{\max}$ (with negative sign to give a DT with an earlier day higher priority), and (c) the first DT's total time required over all resources $\sum_{a\in A_{t,1}} p_{t,u,a}\cdot|Q_{t,u,a}|$, breaking ties randomly. With respect to our benchmark instances, see Section 6, our experiments clearly show that (b), the latest starting day for the first DT $-d_{t,1}^{\max}$, provides the best guidance for the heuristic. It appears reasonable that therapies having less flexibility w.r.t. their start should be given higher priority. Consequently, we will use this function in all our further investigations.

4.2 Time Assignment Phase

In the time assignment phase the planning for each working day is done independently. For a considered day $d \in D'$, the heuristic selects a not yet scheduled DT from G_d and sets the start times of its corresponding activities as early as possible respecting their sequential order, the availabilities of all required resources, and the minimum and maximum time lags.

Algorithm 2 shows this procedure in detail. The input is the considered day and its set of DTs G_d as computed by Algorithm 1. For each resource r available on day d, a time marker C_r is initialized with the start of the service window. This variable in general refers to the most recent point in time resource r has been used. All not yet scheduled activities requiring resource r will be assigned a start time of at least C_r. In each iteration of the while-loop from line 2 on, a DT $u \in U_t$ is selected and removed from G_d according to a priority function DT_priority.

In the for-loop from line 5 on, the start time of each activity $a \in A_{t,u}$ is set to the earliest time at which all required resources are available and a placement of the corresponding activity also does not violate any other constraint. To this end, we first set $S_{t,u,a}$ to the minimum time where all required resources are available, i.e., to $\max_{r\in Q_{t,u,a}} C_r$. At this point unavailability periods and the end of extended service window may not allow a placement of activity a at $S_{t,u,a}$. Hence, we call the function resource_availability that checks if from $S_{t,u,a}$ on all resources in $Q_{t,u,a}$ are available for a duration of at least $p_{t,u,a}$. If this is the case, $S_{t,u,a}$ is returned, otherwise the earliest point in time after

```
1  $C_r \coloneqq W^{\text{start}}_{r,d} \quad \forall r \in R : d \in D^{\text{res}}_r$;
2  while $G_d \neq \emptyset$ do
3      $U_{t,u} \coloneqq \arg\max_{(t,u)\in G_d}$ DT_priority$(t,u)$; ties are broken randomly;
4      $G_d \coloneqq G_d \setminus \{(t,u)\}$;
5      for $a \coloneqq 1$ to $\alpha_{t,u}$ do
6          $S_{t,u,a} \coloneqq \max_{r\in Q_{t,u,a}} C_r$;
7          $S_{t,u,a} \coloneqq$ resource_availability$(t,u,a,S_{t,u,a})$;
8      end
9      $a \coloneqq 1$;
10     while $a \leq \alpha_{t,u}$ do
11         if $a > 1 \wedge S_{t,u,a} < S_{t,u,a-1} + p_{t,u,a-1}$ then
12             $S_{t,u,a} \coloneqq S_{t,u,a-1} + p_{t,u,a-1}$;
13             $S_{t,u,a} \coloneqq$ resource_availability$(t,u,a,S_{t,u,a})$;
14         end
15         for $(a',a) \in P^{\min}_{t,u}$ do
16             if $S_{t,u,a} < S_{t,u,a'} + p_{t,u,a'} + L^{\min}_{t,u,a',a}$ then
17                 $S_{t,u,a} \coloneqq S_{t,u,a'} + p_{t,u,a'} + L^{\min}_{t,u,a',a}$;
18                 $S_{t,u,a} \coloneqq$ resource_availability$(t,u,a,S_{t,u,a})$;
19             end
20         end
21         $a^{\min} \coloneqq a$;
22         for $(a',a) \in P^{\max}_{t,u}$ do
23             if $S_{t,u,a} > S_{t,u,a'} + p_{t,u,a'} + L^{\max}_{t,u,a',a}$ then
24                 $S_{t,u,a'} \coloneqq S_{t,u,a} - p_{t,u,a'} - L^{\max}_{t,u,a',a}$;
25                 $S_{t,u,a'} \coloneqq$ resource_availability$(t,u,a,S_{t,u,a})$;
26                 $a^{\min} \coloneqq \min(a^{\min}, a')$;
27             end
28         end
29         if $a^{\min} = a$ then $a \coloneqq a+1$ ;
30         else $a \coloneqq a^{\min}$ ;
31     end
32     $C_r \coloneqq \max(C_r, S_{t,u,a} + p_{t,u,a}) \quad \forall a = 1,\dots,\alpha_{t,u},\ \forall r \in Q_{t,u,a}$;
33 end
```

Algorithm 2: TWCH TimeAssignment(d, G_d)

$S_{t,u,a}$ allowing a feasible placement, possibly using extended time, is used. Remember that we assume that there always exists sufficient extended time for scheduling any activity, so satisfiability is not a practical issue here.

In the following while-loop from line 10 on we enforce that activities have to be scheduled strictly sequentially in the given order and that the minimum and maximum time lags are adhered. After each modification of a start time the resource_availability function is called to ensure that all resources are available during the whole processing time. First, a is initialized with 1 (line 9), i.e., to the first activity of DT $u \in U_t$. The if-statement in line 11 checks whether activity a starts strictly after $a-1$ and adjusts $S_{t,u,a}$ if necessary. For activity a lines 15 to 20 enforce that for all minimum time lags $(a',a) \in P^{\min}_{t,u}$ inequality $S_{t,u,a} \geq S_{t,u,a'} + p_{t,u,a'} + L^{\min}_{t,u,a',a}$ holds. Afterwards $a^{\min}$ is initialized to a and line 22 to line 28 accomplish for all maximum time lags $(a',a) \in P^{\max}_{t,u}$ that $S_{t,u,a'}$ is greater than or equal to $S_{t,u,a} - p_{t,u,a'} - L^{\max}_{t,u,a',a}$. Moreover,

$a^{\min}$ is set to the activity with the smallest index a' for which $S_{t,u,a'}$ has been changed. This change might result in a violation of the sequential order or the minimum and maximum time lags of the activities $a^{\min}, \ldots, a-1$. Hence, those activities have to be reconsidered. Consequently, if $a^{\min} < a$ then a is set to $a^{\min}$, otherwise a is increased by one. Finally, at line 32 all activities of the considered DT have valid start times and all relevant C_r variables are updated.

The solution quality of Algorithm 2 is influenced to a high degree by the greedy function DT_priority used for selecting the next DT to be considered. To this end we consider three different criteria, with smaller values always indicating higher priorities:

(a) The idle time that emerges on the beam resource if DT $(t, u) \in G_d$ is considered next, i.e., the value $C''_r - C'_r - q_{t,u,r}$ for the beam resource, where C'_r and C''_r are the values of the corresponding C_r variable before and after scheduling $u \in U_t$.

(b) The minimum time a resource required by the considered DT $(t, u) \in G_d$ leaves its regular service window, i.e., $\min_{r \in Q_{t,u,1} \cup \cdots \cup Q_{t,u,\alpha_{t,u}}} W^{\text{end}}_{r,d}$.

(c) The ratio between the processing time of the activity requiring the beam and the total processing time of all activities of a DT $(t, u) \in G_d$, i.e., $\frac{p_{t,u,b}}{\sum_{a \in A_{t,u}} p_{t,u,a}}$, where $b \in A_{t,u}$ is the activity that requires the beam resource.

Preliminary tests indicated that all the above criteria provide reasonable guidance with criterion (a) tending to yield on average better solutions than (b) and (c), but no single criterion dominates the others clearly. A problem, at least with respect to our benchmark instances, is that ties frequently happen, i.e., different DTs sometimes evaluate to the same priority criterion value. To counteract these ties, we define our actual DT_priority function via a lexicographic combination of all three above criteria: First, the value of criterion (a) is considered as priority. In case of a tie, criterion (b) is used, and if a tie happens again, the last criterion (c) is considered. Remaining ties are broken randomly. Note that the sign of all criteria values are inverted in order to obtain large priority values for DTs that should be preferred.

TWCH can be implemented in time $O(n_T \cdot \tau^2_{\max} \cdot d' + |D'| \cdot n_T \cdot \alpha_{\max})$, where $\tau_{\max} = \max_{t \in T} \tau_t$, the maximum number of DTs of any therapy, $d' = \max_{t \in T}(\max_{u \in U_t}(d^{\max}_{t,u} - d^{\min}_{t,u}))$, the maximum number of days within which any DT must start, and $\alpha_{\max} = \max_{t \in T}(\max_{u \in U_t} \alpha_{t,u})$, the maximum number of activites a DT has. Hereby, we neglect the number of resources, their non-availability periods, and the number of time lag constraints.

5 GRASP and Iterated Greedy Metaheuristics

As will be seen in the experimental results TWCH is relatively fast also on instances of practically relevant size. However, it obviously leaves room for improvements regarding solution quality, as some greedy decisions will in general not lead to an overall optimal solution. We therefore consider here two metaheuristic approaches that build upon TWCH: A GRASP and an Iterated Greedy metaheuristic.

5.1 GRASP

Greedy Randomized Adaptive Search Procedure (GRASP) is a prominent metaheuristic building upon a construction heuristic and usually a local search component [18]. The basic idea is to apply a randomized variant of the construction heuristic independently many times, to locally improve each obtained solution, and to select an overall best solution as final one.

In our context, we randomize TWCH's DayAssignment by changing the way the next therapy to be scheduled is selected at line 3 of Algorithm 1 in a GRASP-typical fashion. Let p^{tmin} and p^{tmax} be the minimal and maximal priority value received from therapy_priority(t), $\forall t \in T'$, respectively. The next considered DT is chosen uniformly at random from the subset of T' with therapy_priority$(t) \geq p^{\mathrm{tmax}} - \beta^{\text{gr-rand}} \cdot (p^{\mathrm{tmax}} - p^{\mathrm{tmin}})$. Parameter $\beta^{\text{gr-rand}} \in (0, 1)$ determines the allowed deviation from the highest priority and thus controls the strength of the randomization.

TWCH's original TimeAssignment algorithm is then applied to all days to which DTs are assigned to, determining activity starting times S. Afterwards, we apply as local improvement the following randomized multi-start heuristic individually to each day:

TWCH's TimeAssignment procedure is randomized by modifying line 3 to choose from the $k^{\text{rta-rand}}$ best DTs uniformly at random, with parameter $k^{\text{rta-rand}}$ controlling the strength of the randomization. This randomized construction is iteratively applied until a schedule, not requiring extended times at the respective day, is found or no improvement has been achieved over the last $n^{\text{rta-noimp}}$ iterations. A starting time configuration inducing the smallest cost is finally kept.

5.2 Iterated Greedy

In a nutshell, Iterated Greedy (IG) generates a sequence of solutions by iterating over greedy constructive heuristics using two main phases: destruction and construction. In the destruction phase some solution components are removed from a previously constructed complete candidate solution. The construction procedure then applies a greedy constructive heuristic to reconstruct a complete solution. An acceptance criterion is applied to decide whether or not to continue with this new solution. IG iterates over these steps until a stopping criterion is met. IG has many successful applications, in particular also in the domain of scheduling. See for example [19] where one of the first applications of IG to a permutation flowshop scheduling problem is described.

Our IG works as follows. TWCH as described in Section 4 is again used to create an initial solution. The destruction operator drops for $\beta^{\text{ig-dest}} \cdot n_T$ randomly selected therapies the assignment of all their DTs, i.e., removing them from the sets G_d and invalidating their assigned days $Z_{t,u}$. Hereby, $\beta^{\text{ig-dest}} \in (0, 1)$ is an exogenous parameter, the destruction rate. The construction step is then performed in a straight-forward way by reapplying TWCH's DayAssignment for the set of removed therapies, warm-starting with the current sets G_d. Finally, TWCH's time assignment is applied from scratch to all working days for which G_d has changed in comparison to the original solution.

In addition, we also try to locally improve the obtained solution at each iteration by performing the day-wise multi-start randomized TimeAssignment as described above for the GRASP. Our IG always accepts a new solution as new incumbent if it improves upon the previously best solution.

6 Computational Study

We start this section by describing the used test instances and the used generation approach. Afterwards, we give details on the computational experiments w.r.t. these instances.

6.1 Test Instances

We created artificial benchmark instances related to the expected situation at MedAustron and real particle treatments. These instances are available at `http://www.ac.tuwien.ac.at/research/problem-instances`. The main characteristic of an instance is its number of therapies n_T. We consider 5 instances for 10, 20, 50, 70, 100, 150, 200, and 300 therapies. In the used naming schema we encode first the number of therapies followed by a consecutive number. Depending on the number of treatments, we determine regular opening hours. Below 100 therapies we consider 7 hours per day and above 14 hours per day. This helps to keep smaller instances challenging. The time horizon n_D (considered days) is then derived from the number of considered therapies.

For each of the therapies the number of DTs is chosen uniformly at random from the interval $\{8, \ldots, 35\}$, reflecting the duration of real particle therapies. The period in which a therapy might start is assumed to have a fixed length of two weeks. We set for all therapies $n_t^{\text{twmin}} = 4$, $n_t^{\text{twmax}} = 5$, $\delta_t^{\min} = 1$ and $\delta_t^{\max} = 5$. Based on these values we calculate $d^{\text{buffer}} = 13$ that serves as an upper bound on the difference (in days) between the fastest and the slowest completion time of a therapy. Using this bound we determine the latest day by which the first DT of a therapy t might be provided: $d_{t,1}^{\text{latest}} := n_D - d^{\text{buffer}} - \lceil \tau_t / n_t^{\text{twmax}} \rceil \cdot 7 + 1$. Therefore, the earliest day by which the therapy can start is chosen from $\{1, \ldots, d_{t,1}^{\text{latest}} - 14\}$.

Most DTs have exactly five activities. With a probability of 5% a therapy requires sedation and in that case its DTs consist of seven activities each.

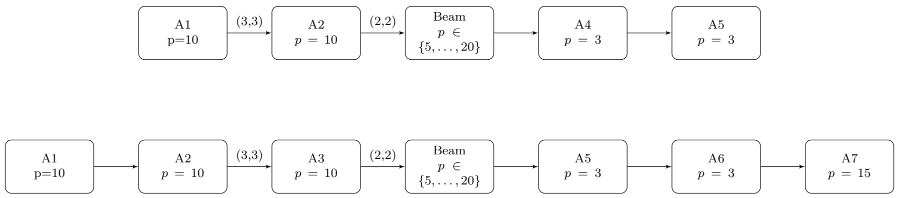

Fig. 1: Activity structure of the daily treatments.

Figure 1 shows the two types of daily treatments together with the associated processing times in minutes. Tuples above the arcs provide minimum and maximum time lags. The processing times of the beam activities are drawn uniformly at random from $\{5, \ldots, 20\}$ as shown in the figure. For the processing times of the other activities we consider a spread of ±20%, i.e., we choose from $\{\lfloor 0.8p \rfloor, \ldots, \lfloor 1.2p \rfloor\}$ uniformly at random. These choices are made per therapy and kept the same for all its DTs.

In the following we describe the resources associated with the activities. The main bottleneck in the considered application scenario is the beam resource. Its regular availability corresponds to the regular opening time on each working day. The availability of the three irradiation rooms is set correspondingly. As mentioned above some treatments require sedation which means that an anesthetist is needed. The corresponding resource has a regular service window spanning the first half of the regular working day. Some activities require the attendance of the oncologist responsible for the associated patient. Radio oncologists work in two shifts, the first shift spans the first two thirds of the regular opening time and the second shift spans the last two thirds of the regular opening time. In each shift five oncologists are available. For each therapy the associated oncologist is selected uniformly at random from both shifts. The first treatment of each therapy needs to be provided before noon. To model this we introduce an additional resource that spans half of the regular opening time. On each day there is a lunch break in the middle of the day modeled by unavailability periods. For the instances featuring only seven operating hours all resources are available during the total regular working day and the lunch break is placed at the end of the regular opening time.

For all instances the weights $\gamma^{\text{ext}} = 1/60$ and $\gamma^{\text{finish}} = 1/100$ have been used for the two terms of the objective function. The intuition for the γ^{ext} value is that the use of one hour of extended time of a resource corresponds to one unit in the objective function (1). Weight γ^{finish} has been chosen in such a way that finishing a therapy earlier by one day by performing a DT entirely in the extended time is typically not justified.

6.2 Computational Results

All algorithms were implemented in C++11 and compiled with G++ 4.8.4, GUROBI 6.5 was used for solving the MILP models, and all experiments were

Instance	TWCH's DayAssignment			Relaxed MILP		
	$\overline{da\text{-}obj}$	$\sigma(da\text{-}obj)$	$time[s]$	$da\text{-}obj$	lb	$time[s]$
ptpsp_010-01	**0.220**	0.000	0.001	**0.220**	0.220	3.4
ptpsp_010-02	**0.160**	0.000	0.001	**0.160**	0.160	4.6
ptpsp_010-03	**0.160**	0.000	0.000	**0.160**	0.160	3.0
ptpsp_010-04	**0.180**	0.000	0.001	**0.180**	0.180	4.0
ptpsp_010-05	**0.180**	0.000	0.001	**0.180**	0.180	4.2
ptpsp_020-01	**0.400**	0.000	0.001	**0.400**	0.400	6.9
ptpsp_020-02	**0.450**	0.000	0.001	**0.450**	0.450	7.8
ptpsp_020-03	**0.460**	0.000	0.001	**0.460**	0.460	10.4
ptpsp_020-04	**0.320**	0.000	0.001	**0.320**	0.320	9.2
ptpsp_020-05	**0.320**	0.000	0.001	**0.320**	0.320	7.9
ptpsp_050-01	7.532	0.037	0.003	**1.770**	1.601	7200.0
ptpsp_050-02	4.005	0.007	0.002	**1.480**	1.443	7200.0
ptpsp_050-03	11.679	0.582	0.003	**2.390**	2.258	7200.0
ptpsp_050-04	2.597	0.000	0.003	**1.470**	1.376	7200.0
ptpsp_050-05	7.325	0.434	0.003	**2.317**	2.142	7200.0
ptpsp_070-01	37.256	1.804	0.005	**10.073**	8.362	7200.0
ptpsp_070-02	43.895	0.258	0.004	**14.283**	13.933	7200.0
ptpsp_070-03	**3.665**	0.029	0.004	4.963	2.257	7200.0
ptpsp_070-04	**12.187**	0.481	0.004	NA	3.934	7200.0
ptpsp_070-05	5.165	0.068	0.004	**2.780**	2.657	7200.0
ptpsp_100-01	5.110	0.000	0.005	**3.953**	3.117	7200.0
ptpsp_100-02	4.719	0.066	0.005	**2.970**	2.900	7200.0
ptpsp_100-03	8.083	0.626	0.006	**3.710**	3.592	7200.0
ptpsp_100-04	9.966	0.190	0.006	**4.340**	4.189	7200.0
ptpsp_100-05	5.713	0.162	0.006	**2.860**	2.825	7200.0
ptpsp_150-01	**46.317**	0.733	0.010	NA	11.528	7200.0
ptpsp_150-02	**30.367**	0.233	0.010	NA	7.639	7200.0
ptpsp_150-03	**13.787**	0.171	0.008	NA	7.176	7200.0
ptpsp_150-04	10.541	0.347	0.008	**5.950**	5.811	7200.0
ptpsp_150-05	26.764	0.499	0.009	**9.067**	8.858	7200.0
ptpsp_200-01	17.611	0.711	0.012	**12.167**	7.500	7200.0
ptpsp_200-02	**53.440**	0.649	0.012	NA	11.068	7200.0
ptpsp_200-03	70.021	1.166	0.013	**13.940**	13.640	7200.0
ptpsp_200-04	**89.349**	2.343	0.014	NA	13.035	7200.0
ptpsp_200-05	27.785	0.166	0.013	**11.253**	10.883	7200.0
ptpsp_300-01	**56.725**	1.122	0.020	NA	0.000	7200.0
ptpsp_300-02	**68.653**	1.221	0.020	NA	17.455	7200.0
ptpsp_300-03	**60.787**	0.646	0.019	NA	19.660	7200.0
ptpsp_300-04	10.645	0.127	0.018	**8.717**	7.631	7200.0
ptpsp_300-05	**69.533**	0.606	0.020	NA	16.850	7200.0

Table 1: Results of TWCH's DayAssignment and the relaxed MILP, which also considers only day assignments.

carried out on a single core of an Intel Xeon E5-2630v2 processor with 2.6 GHz and about 4 GB RAM per core.

In the first series of conducted experiments we focus on the day assignment level only and assess the performance of TWCH's DayAssignment in comparison to the relaxed MILP model (1) to (13) from Section 3.1. Thus, resource consumptions are only considered at the aggregated level and no detailed time planning is done. For each of our benchmark instances, the MILP was solved

Instance size		GRASP			IG	
n_T	$\beta^{\text{gr-rand}}$	$n^{\text{rta-noimp}}$	$k^{\text{rta-rand}}$	$\beta^{\text{ig-dest}}$	$n^{\text{rta-noimp}}$	$k^{\text{rta-rand}}$
10, 20, 50, 70	0.310	$\lfloor 1.25 \cdot \lvert G_d \rvert \rfloor$	2	0.095	$\lfloor 1.85 \cdot \lvert G_d \rvert \rfloor$	2
100, 150	0.155	$\lfloor 1.19 \cdot \lvert G_d \rvert \rfloor$	2	0.090	$\lfloor 1.50 \cdot \lvert G_d \rvert \rfloor$	2
200, 300	0.090	$\lfloor 1.28 \cdot \lvert G_d \rvert \rfloor$	2	0.110	$\lfloor 1.60 \cdot \lvert G_d \rvert \rfloor$	2

Table 2: Parameter settings for GRASP and IG.

using a CPU-time limit of 2 hours, while TWCH's DayAssignment was applied 30 times because of its stochastic nature. Table 1 shows the obtained results for each instance. For TWCH average objective values $\overline{\textit{da-obj}}$ are listed together with the corresponding standard deviations $\sigma(\textit{da-obj})$ and the median computation times *time*. For the MILP approach, *da-obj* indicates the objective value of the best feasible solution, *lb* the final lower bound, and *time* the CPU-time when the algorithm terminated, either with proven optimality or when the time limit had been reached.

TWCH yielded reasonable solutions for each test instance quickly, but improvement potential can also clearly be seen. The MILP approach could not find a provably optimal solution for any instance with $n_T \geq 50$. Only instances that allowed solutions with none or very little extended time could be reasonably solved via the MILP. A reason for the rather poor performance of the MILP seem to be substantial symmetries in the model. Due to the poor performance of the MILP for the day assignment only, we can conclude that solving the full MILP including the detailed time planning unfortunately is in practice impossible for any instance of realistic size.

In the next series of experiments we consider the complete TWCH as well as GRASP and IG. In a preliminary study we determined 20 CPU-minutes to be a reasonable time limit for the metaheuristics after which only minor further improvements can be expected also on the largest instances with 300 therapies. We therefore used this time limit as termination criterion in all following metaheuristic runs. The automatic parameter configuration tool irace [12] was applied for tuning IG's and GRASP's strategy parameters on three instance sets with a budget of 2000 runs. The instance sets used for tuning consisted of six new training instances with 50 and 70, 100 and 150, 200 and 300 therapies. The tuned parameter settings are depicted in Table 2. Note that we apply the settings for instances with 50 and 70 therapies also on the instances with 10 and 20 therapies.

Table 3 shows for each of the approaches and each of the benchmark instances the average final objective values *obj* and corresponding standard deviations $\sigma(\textit{obj})$ over 30 runs. For TWCH median runtimes are also listed. For the benchmark instances with 10 and 20 therapies all three approaches always yielded the same objective values, which also coincide with the objective values from Table 1. This implies that in these cases the time assignment can be done without additional extended time, the instances are thus relatively easy. Since the MILP has shown that these objectives are optimal for the day

Instance	TWCH			GRASP		IG	
	obj	$\sigma(obj)$	*time*[s]	*obj*	$\sigma(obj)$	*obj*	$\sigma(obj)$
ptpsp_010-01	**0.220**	0.000	0.001	**0.220**	0.000	**0.220**	0.000
ptpsp_010-02	**0.160**	0.000	0.001	**0.160**	0.000	**0.160**	0.000
ptpsp_010-03	**0.160**	0.000	0.001	**0.160**	0.000	**0.160**	0.000
ptpsp_010-04	**0.180**	0.000	0.001	**0.180**	0.000	**0.180**	0.000
ptpsp_010-05	**0.180**	0.000	0.001	**0.180**	0.000	**0.180**	0.000
ptpsp_020-01	**0.400**	0.000	0.002	**0.400**	0.000	**0.400**	0.000
ptpsp_020-02	**0.450**	0.000	0.002	**0.450**	0.000	**0.450**	0.000
ptpsp_020-03	**0.460**	0.000	0.003	**0.460**	0.000	**0.460**	0.000
ptpsp_020-04	**0.320**	0.000	0.002	**0.320**	0.000	**0.320**	0.000
ptpsp_020-05	**0.320**	0.000	0.002	**0.320**	0.000	**0.320**	0.000
ptpsp_050-01	159.842	0.248	0.009	120.234	0.966	**105.837**	4.452
ptpsp_050-02	128.793	0.320	0.007	88.034	0.921	**80.724**	1.908
ptpsp_050-03	161.722	2.831	0.010	117.866	1.380	**93.362**	3.907
ptpsp_050-04	161.795	0.221	0.008	**114.689**	1.421	122.618	2.596
ptpsp_050-05	177.635	0.748	0.009	131.223	0.874	**113.455**	6.263
ptpsp_070-01	304.241	4.370	0.012	232.000	2.282	**192.295**	3.092
ptpsp_070-02	278.451	2.322	0.012	211.597	2.993	**169.409**	5.157
ptpsp_070-03	165.936	3.618	0.011	**118.942**	1.941	121.089	4.439
ptpsp_070-04	194.032	3.012	0.011	148.013	1.470	**116.831**	3.644
ptpsp_070-05	162.713	4.288	0.010	118.022	1.985	**107.238**	4.433
ptpsp_100-01	183.740	1.799	0.025	**138.086**	2.319	149.066	3.388
ptpsp_100-02	136.303	3.488	0.028	**105.593**	0.910	106.684	2.306
ptpsp_100-03	245.927	4.125	0.030	185.778	1.452	**185.629**	4.320
ptpsp_100-04	162.602	1.583	0.030	133.788	1.505	**122.859**	2.448
ptpsp_100-05	247.242	4.015	0.028	179.523	2.051	**177.468**	3.556
ptpsp_150-01	320.521	5.625	0.049	265.495	1.779	**186.702**	3.610
ptpsp_150-02	372.983	4.612	0.047	300.542	2.412	**252.423**	5.615
ptpsp_150-03	273.096	6.973	0.041	207.086	2.536	**195.565**	4.880
ptpsp_150-04	182.204	4.230	0.040	131.184	2.602	**126.098**	4.576
ptpsp_150-05	263.687	5.103	0.045	210.231	2.104	**168.895**	3.903
ptpsp_200-01	340.069	7.659	0.057	255.235	2.926	**233.247**	5.986
ptpsp_200-02	439.731	5.956	0.058	350.984	3.355	**292.811**	6.179
ptpsp_200-03	487.131	4.096	0.066	409.564	1.389	**335.429**	4.576
ptpsp_200-04	548.790	6.364	0.066	457.902	3.994	**352.461**	8.034
ptpsp_200-05	317.170	2.407	0.060	263.558	1.708	**230.248**	3.667
ptpsp_300-01	708.705	11.009	0.098	565.907	3.573	**512.875**	4.269
ptpsp_300-02	727.669	13.390	0.099	579.483	3.602	**519.220**	6.955
ptpsp_300-03	706.027	10.762	0.098	539.983	3.464	**521.847**	6.672
ptpsp_300-04	527.563	7.071	0.096	**370.891**	2.556	375.673	4.558
ptpsp_300-05	689.882	10.095	0.099	566.615	5.095	**509.551**	4.808

Table 3: Average results of TWCH, GRASP and IG over 30 runs.

assignment, it allows to conclude that these instances could also be solved optimally when additionally considering the time assignment. For all other instances, this observation does not hold anymore. It can clearly be seen that the detailed consideration of scheduling the activities within the working days imposed significant additional costs. Both, GRASP as well as IG, could find substantially better solutions than TWCH in all those cases. The clear winner, however, is IG, which performed best on most instances. The major reason for its superiority seems to be the fact that on the one hand its iterations are less costly as only parts of a solution are affected by destruction and construction, and on the other hand information is in this way also kept over the iterations and further fine-tuned.

7 Conclusion

We have seen that the midterm patient scheduling problem arising in modern cancer treatment centers applying particle therapy is particularly challenging as it involves a planning at day level as well as a depending detailed scheduling of activities at each day. We presented a MILP model for the whole problem, but it became clear that already solving just the day planning part is practically intractable for most instances of realistic size. Therefore, we considered a therapy-wise construction heuristic based on greedy principles featuring a forward-looking mechanism to avoid too naive decisions. This heuristic is fast and provides already reasonable solutions. We further built upon this construction heuristic, proposing a GRASP and an IG metaheuristic. In our experiments on newly created benchmark instances based on properties of realistic scenarios, IG yielded the best results. Its superiority over GRASP can be explained by the computationally more efficient destruction and construction: Only parts of each solution are reconsidered from scratch, and substantial information survives from one iteration to the next and is further fine-tuned. All proposed metaheuristics scale well to instances of practically relevant size.

Nevertheless, the problem formulation considered here captures only a restricted variant of the challenges arising in practice. In real world applications further considerations need to be taken into account. To provide a well-structured treatment process for the patients, it is desirable to ensure that DTs are provided at roughly the same time on each day within a week. The variations between weeks can typically be larger but should also be kept reasonably small. For the staff it is preferable to avoid gaps within the workday. This can be accomplished by minimizing the time a resource stays idle between subsequent uses.

Currently, we are only dealing with the core treatment phase providing the actual irradiation sessions. In general, a particle therapy also involves a so-called pre-treatment or treatment planning phase. In this stage personalized equipment needed for irradiation is produced. Moreover, a substantial amount of planning has to be done to determine the detailed treatment strategy. This requires the work of professionals and the use of specialized equipment needed for examinations of the patients (such as CT, MR). Since the resources needed in this phase are limited (although not being as scarce as the beam) we plan on including this stage into the scheduling process.

Concerning more advanced algorithms, an obvious next step is to consider various kinds of local search neighborhoods, such as moving a single DT by one day or locally rearranging a small number of activities within one day, and to embed corresponding local search procedures within IG and/or GRASP. Furthermore, MILP-based large neighborhood search techniques appear particularly promising in the context of this problem: Despite that the MILP model turned out to not scale well enough to be directly applied to the whole problem, even not at the day assignment level alone, it can be used within a destroy-and-repair approach to reassign a larger number of treatments (or activities) in a locally optimal way.

References

1. Brucker, P., Knust, S.: Resource-constrained project scheduling and timetabling. In: Practice and Theory of Automated Timetabling III, LNCS, pp. 277–293. Springer (2001)
2. Burke, E.K., Leite-Rocha, P., Petrovic, S.: An integer linear programming model for the radiotherapy treatment scheduling problem (2011). ArXiv preprint arXiv:1103.3391
3. Conforti, D., Guerriero, F., Guido, R.: Optimization models for radiotherapy patient scheduling. 4OR **6**(3), 263–278 (2008)
4. Conforti, D., Guerriero, F., Guido, R.: Non-block scheduling with priority for radiotherapy treatments. European Journal of Operational Research **201**(1), 289–296 (2010)
5. Conforti, D., Guerriero, F., Guido, R., Veltri, M.: An optimal decision-making approach for the management of radiotherapy patients. OR Spectrum **33**(1), 123–148 (2011)
6. Hartmann, S., Briskorn, D.: A survey of variants and extensions of the resource-constrained project scheduling problem. European Journal of Operational Research **207**(1), 1 – 14 (2010)
7. Kapamara, T., Petrovic, D.: A heuristics and steepest hill climbing method to scheduling radiotherapy patients. In: Proceedings of the International Conference on Operational Research Applied to Health Services (ORAHS). Catholic University of Leuven, Leuven, Belgium (2009)
8. Kapamara, T., Sheibani, K., Haas, O., Petrovic, D., Reeves, C.: A review of scheduling problems in radiotherapy. In: Proceedings of the International Control Systems Engineering Conference (ICSE), pp. 207–211. Coventry University Publishing, Coventry, UK (2006)
9. Larsson, S.N.: Radiotherapy patient scheduling using a desktop personal computer. Clinical Oncology **5**(2), 98–101 (1993)
10. Legrain, A., Fortin, M.A., Lahrichi, N., Rousseau, L.M.: Online stochastic optimization of radiotherapy patient scheduling. Health Care Management Science **18**(2), 110–123 (2015)
11. Leite-Rocha, P.: Novel approaches to radiotherapy treatment scheduling. Ph.D. thesis, University of Nottingham (2011)
12. López-Ibáñez, M., Dubois-Lacoste, J., Stützle, T., Birattari, M.: The irace package, iterated race for automatic algorithm configuration. Tech. Rep. TR/IRIDIA/2011-004, IRIDIA, Université libre de Bruxelles, Belgium (2011)
13. Men, C.: Optimization models for radiation therapy: treatment planning and patient scheduling. Ph.D. thesis, University of Florida (2009)
14. Petrovic, D., Morshed, M., Petrovic, S.: Genetic algorithm based scheduling of radiotherapy treatments for cancer patients. Proceedings of the Conference on Artificial Intelligence in Medicine (AIME) **5651**, 101–105 (2009)
15. Petrovic, D., Morshed, M., Petrovic, S.: Multi-objective genetic algorithms for scheduling of radiotherapy treatments for categorised cancer patients. Expert Systems with Applications **38**(6), 6994–7002 (2011)
16. Petrovic, S., Leite-Rocha, P.: Constructive and GRASP approaches to radiotherapy treatment scheduling. In: Advances in Electrical and Electronics Engineering - IAENG Special Edition of the World Congress on Engineering and Computer Science 2008, pp. 192–200. IEEE (2008)
17. Petrovic, S., Leung, W., Song, X., Sundar, S.: Algorithms for radiotherapy treatment booking. In: 25th Workshop of the UK Planning and Scheduling Special Interest Group, pp. 105–112. Nottingham, UK (2006)

18. Resende, M., Ribeiro, C.: Greedy randomized adaptive search procedures. In: F. Glover, G. Kochenberger (eds.) Handbook of Metaheuristics, pp. 219–249. Kluwer Academic Publishers (2003)
19. Ruiz, R., Stützle, T.: A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. European Journal of Operational Research **177**(3), 2033–2049 (2007)
20. Sauré, A., Patrick, J., Tyldesley, S., Puterman, M.L.: Dynamic multi-appointment patient scheduling for radiation therapy. European Journal of Operational Research **223**(2), 573–584 (2012)
21. Stewart, B., Wild, C.P., et al.: World cancer report 2014. World (2015)
22. WHO: The top 10 causes of death. `http://www.who.int/mediacentre/factsheets/fs310/en/` (2014). Accessed: 2016-03-01