

WILEY

INTERNATIONAL
TRANSACTIONS
IN OPERATIONAL
RESEARCHIntl. Trans. in Op. Res. 27 (2020) 458–479
DOI: 10.1111/itor.12579

Particle therapy patient scheduling with limited starting time variations of daily treatments

Johannes Maschler and Günther R. Raidl

Institute of Logic and Computation, TU Wien, Favoritenstraße 9–11/186-1, 1040 Vienna, Austria
E-mail: maschler@ac.tuwien.ac.at [Maschler]; raidl@ac.tuwien.ac.at [Raidl]

Received 15 November 2017; received in revised form 7 May 2018; accepted 15 July 2018

Abstract

The particle therapy patient scheduling problem (PTPSP) arises in modern cancer treatment facilities that provide particle therapy. It consists of scheduling a set of therapies within a planning horizon of several months. A particularity of PTSP compared with classical radiotherapy scheduling is that therapies need not only be assigned to days but also scheduled within each day to account for the more complicated operational scenario. In an earlier work, we introduced this novel problem setting and provided first algorithms including an iterated greedy (IG) metaheuristic. In this work, we consider an important extension to the PTPSP emerging from practice in which the therapies should be provided on treatment days roughly at the same time. To be more specific, the variation between the starting times of the therapies' individual treatments should not exceed the given limits, and needs otherwise to be minimized. This additional constraint implies that the sequencing parts within each day can no longer be treated independently. To tackle this variant of PTPSP, we revise our previous IG and exchange its main components: the part of the applied construction heuristic for scheduling within the days and the local search algorithm. The resulting metaheuristic provides promising results for the proposed extension of the PTPSP and further enhances the existing approach for the original problem.

Keywords: cancer treatment; particle therapy patient scheduling; iterated greedy metaheuristic; hybrid metaheuristic

1. Introduction

Particle therapy is a relatively novel and highly promising option to provide cancer treatments. A proton or carbon beam is produced by either a cyclotron or a synchrotron and is directed into one of up to five treatment rooms, where patients are irradiated. Since several tasks have to be completed in a treatment room before and after an actual irradiation, the usually single available beam is switched between the available treatment rooms to maximize the throughput of the facility. Consequently,

The copyright line for this article was changed on November 20, 2018 after original online publication.

© 2018 The Authors.

International Transactions in Operational Research published by John Wiley & Sons Ltd on behalf of International Federation of Operational Research Societies

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

the main challenge is to arrange the individual treatments in such a way that idle times on the particle beam are minimized. We consider here specifically the particle therapy treatment center MedAustron in Wiener Neustadt, Austria, which offers three treatment rooms.

The particle therapy patient scheduling problem (PTPSP) addresses the midterm planning part of such a particle therapy treatment center and was first introduced in our recent work (Maschler et al., 2016). In PTPSP an effective plan has to be found for performing numerous therapies, each consisting of daily treatments (DTs) provided on 8–35 subsequent days. Therapies have to start on Mondays or Tuesdays between an earliest and a latest allowed starting day. After a therapy is started, the number of DTs that are provided each week has to stay between a lower and an upper bound. Moreover, there is a minimal and a maximal number of days that are allowed to pass between two subsequent DTs, and there has to be a break from the treatment of at least two consecutive days each week. The DTs have resource requirements that vary with time, but each specific resource is required at most once for a consecutive time period. These varying requirements originate from the different tasks involved in providing the treatments. Each resource can only be used by one DT at a time. Among others, the considered resources involve the particle beam, treatment rooms, radio oncologists, and an anesthetist. In terms of the resource-constrained project scheduling literature (see, e.g., Hartmann and Briskorn, 2010), DTs would be called activities with resource requests varying with time. Typically, the facility is open from Mondays to Fridays, but after recurring maintenance tasks DTs are also performed on Saturdays. Whenever the treatment center is open, resources have a regular availability period followed by an extended availability period in which they can be used, where the use of the latter induces (additional) costs. Furthermore, the availability of resources can be interrupted by so-called unavailability periods. The aim of the PTPSP is to schedule a given set of therapies by determining days and times for all corresponding DTs while considering all operational constraints. The objective is to minimize the use of extended availability periods, while the therapies have to be completed as early as possible. In addition, the DTs belonging to the same therapy should be planned roughly at the same time, in order to provide a consistent schedule for the patients. Ideally, the starting times of a therapy's DTs should not differ more than a half an hour within each week. Between two weeks, the starting times are allowed to differ by two hours. However, consistent starting times for DTs are not of direct medical relevance and, consequently, should not induce additional use of extended service periods or delay therapies. This consideration of similar DT starting times is a practically highly relevant extension of the PTPSP as introduced in Maschler et al. (2016).

The aim of this work is twofold. On the one hand, PTPSP is extended to cover the aspect that the variation among the times at which therapies' DTs are provided is minimized. On the other hand, we study an improved variant of the iterated greedy (IG) metaheuristic from Maschler et al. (2016). Preliminary results of this effort for the original problem formulation have been published in Maschler et al. (2017). We propose a novel construction heuristic that assigns DTs to days as the previous construction heuristic but replaces the part for determining starting times with one that is more appropriate for the extended problem variant. This new heuristic applied within the IG's construction phase is able to keep relative timing characteristics of untouched DTs to a larger extent. Furthermore, we replace the so far rather simple local improvement operator by a local search method that alternately considers a DT exchange neighborhood and solves a linear programming (LP) model for determining updated nominal starting times. The IG is compared with our previous metaheuristic and shows significant

improvements on all considered benchmark instances. Moreover, we assess the qualities of the individual components of the metaheuristic by gradually transforming the previous approach into the proposed one.

The remainder of this work is structured as follows. After giving an overview of related literature in the next section, a formal problem definition is provided in Section 3. We present the enhanced IG metaheuristic in Section 4. The conducted experiments are then discussed in Section 5. Finally, Section 6 concludes this article with an outlook on future work.

2. Related work

Midterm planning for classical radiotherapy has attracted the focus of the scheduling community starting with the works from Kapamara et al. (2006) and Petrovic et al. (2006). Several further heuristic as well as exact approaches followed. Heuristic techniques range from a greedy randomized adaptive search procedure (GRASP) (Petrovic and Leite-Rocha, 2008) and steepest hill climbing methods (Kapamara and Petrovic, 2009; Riff et al., 2016) to more advanced techniques using genetic algorithms (GAs) (Petrovic et al., 2009, 2011). Exact methods are based on mixed integer linear programming (MILP) models and consider different levels of granularity (Conforti et al., 2008; Burke et al., 2011). All these works have in common that they assign treatments only to days, but do not sequence the treatments within each day. The reason is that in the considered scenarios linear accelerators are used, which serve single treatment rooms exclusively. Hence, only a sequential processing of treatments is possible. This stands in contrast to the PTPSP, where the particle beam is shared between multiple treatment rooms and a finer grained scheduling is necessary to maximize the throughput of the facility.

In Maschler et al. (2016), we formalized the PTPSP via an MILP model. However, even solving a strongly simplified version of the model turned out to be practically intractable. Therefore, we proposed the therapy-wise construction heuristic (TWCH), which acts in two phases by assigning first all DTs to days (day assignment) and then scheduling the DTs on each day (time assignment). Moreover, a GRASP and an IG metaheuristic, which are based on this construction heuristic, were developed. Experiments indicated that the IG yields superior results in comparison to the Greedy Randomized Adaptive Search Procedure (GRASP). This is mainly due to the fact that the IG preserves substantial parts of the solution from one iteration to the next, and consequently, poor decisions made especially in the first phase of TWCH can be corrected in the course of the iterations. However, the IG proposed in Maschler et al. (2016) does not exhaust its full potential: Moving DTs between days might require to reevaluate the start times of all DTs, and this is done by simply dropping all start times of the considered days. In addition, we used a local improvement operator within the IG that is based on applying a randomized version of the time assignment phase of TWCH iteratively many times. Even though this local improvement operator is able to enhance solutions rather quickly, this approach has the drawback that partly redundant work is repeatedly done, still yielding relatively similar solutions for the time assignments.

Our subsequent work (Maschler et al., 2018) focuses on the decomposition of the PTPSP into a day assignment part and a sequencing part. This decomposition makes the problem computationally more manageable as it allows us to separate the allocation of DTs to days with determining the DTs' starting times. However, both levels are dependent on a large degree. Especially, on the day

assignment level we have to be aware of the behavior of the time assignment part. Hence, we provide a surrogate model that predicts the use of extended service windows given a set of DTs and a specific day. Experiments showed that the application within our IG metaheuristic allowed us to improve our previous results significantly.

Another scheduling problem related to particle therapy is considered in Vogl et al. (2018). Although their problem is from a setting similar to ours, it differs in many details. While our emphasis is mainly on the throughput of the facility, that is, on the scheduling of DTs under limited resource availabilities, the authors shift the subject more to the aspect of planning therapies including activities surrounding the core therapy. In particular, they have additional appointments that need to be provided either before or after a DT once a week. These appointments distinguish themselves from DTs as they can be supplied by different resources. In comparison to the PTPSP as described here, Vogl et al. (2018) assume that the resources are available on all days without any further restrictions. Moreover, their objective function differs substantially from ours: the aim is on minimizing the total idle time of the beam resource and the violations of time windows. Vogl et al. (2018) propose a multiencoded GA and compare two solution decoding approaches.

The works by Riedler et al. (2020) and Horn et al. (2017) deal with strongly simplified variants of the time assignment part for a single day. Motivated by the fact that irradiation times are known exactly beforehand, Riedler et al. (2020) consider a resource-constrained project scheduling problem for high time resolutions. Their solution approach is to relax the problem by partitioning time into so-called time buckets, which are then iteratively refined until an optimal solution is found. The problem variant considered by Horn et al. (2017) is even closer to our scenario. Each of their jobs requires one common resource during a part of its processing time and one of several secondary resources for the entire processing time. Such jobs model the essential aspect of our DTs, where the common resource corresponds to the beam and the secondary resources correlate with the rooms. They consider the minimization of the makespan as objective and provide an exact A* algorithm, a heuristic beam search, and a hybrid thereof.

3. Problem definition

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

Each therapy $t \in T$ consists of a set of DTs $U_t = \{1, \dots, \tau_t\}$. In the course of a therapy, the number of DTs applied per week has to be in the range from n_t^{twmin} to n_t^{twmax} and DTs have to be performed at most every $\delta_t^{min} \geq 1$ and at least every $\delta_t^{max} \geq \delta_t^{min}$ days. Between two weeks, there have to be at least two days where no DT is performed. In addition, the maximum intended time difference of the starting times of the DTs within the same week and between two consecutive weeks is denoted with $\delta^{intra w}$ and $\delta^{inter w}$, respectively. The set of possible start days for each DT $u \in U_t$ is given by the subset $\{d_{t,u}^{min}, \dots, d_{t,u}^{max}\} \subseteq D$ of days. For each DT $u \in U_t$ we are given a processing time $p_{t,u} \geq 0$ and a set of required resources $Q_{t,u} \subseteq R$. In the execution of a DT, each resource $r \in Q_{t,u}$ is in general required during a part of the whole processing time specified by the time interval $P_{t,u,r} = [P_{t,u,r}^{start}, P_{t,u,r}^{end}] \subseteq [0, p_{t,u})$.

The planning horizon is structured into a subset $D' \subseteq D$ of working days on which the treatment center is actually open and DTs can be scheduled on. The weeks covered by D are denoted by $V = \{1, \dots, n_V\}$. Furthermore, let $\bigcup_{v \in V} D'_v$ be the partitioning of D' into n_V subsets corresponding to the weeks. For each working day $d \in D'$, we have a fundamental opening time $\widetilde{W}_d = [\widetilde{W}_d^{\text{start}}, \widetilde{W}_d^{\text{end}})$ that limits the availability of all resources on the considered day.

Each resource $r \in R$ is available on a subset $D_r^{\text{res}} \subseteq D'$ of the working days. On such days, the availability of each resource is defined by a regular service time window $W_{r,d} = [W_{r,d}^{\text{start}}, W_{r,d}^{\text{end}}) \subseteq \widetilde{W}_d$ that is immediately followed by an extended service time window $\widehat{W}_{r,d} = [W_{r,d}^{\text{end}}, \widetilde{W}_d^{\text{end}}) \subseteq \widetilde{W}_d$. Moreover, for each resource $r \in R$ and each day $d \in D_r^{\text{res}}$, the availability of resource r may be interrupted by a set of unavailability intervals $\overline{W}_{r,d} = \bigcup_{w=1, \dots, \omega_{r,d}} \overline{W}_{r,d,w}$ with $\overline{W}_{r,d,w} = [\overline{W}_{r,d,w}^{\text{start}}, \overline{W}_{r,d,w}^{\text{end}}] \subset W_{r,d} \cup \widehat{W}_{r,d}$.

We represent a solution for the PTPSP as a triple (Z, S, \widetilde{S}) , where $Z = \{Z_{t,u} \in D' \mid t \in T, u \in U_t\}$ denotes the days at which the DTs are planned, $S = \{S_{t,u} \geq 0 \mid t \in T, u \in U_t\}$ is the set of start times for all the DTs on the respective days, and $\widetilde{S} = \{\widetilde{S}_{t,v} \mid \forall t \in T, v \in V\}$ corresponds to the set of nominal starting times of the therapies' DTs within the weeks. A solution is feasible if all resource availabilities, precedence relations, and the remaining operational constraints are respected. The objective is to minimize primarily the use of extended time over all resources R while finishing each therapy as early as possible. In addition, the goal is to minimize the deviation of the DTs' starting times from the corresponding nominal starting times and to minimize the difference of the nominal starting times between weeks. More formally, we aim at minimizing

$$\begin{aligned} & \gamma^{\text{ext}} \sum_{r \in R} \sum_{d \in D_r^{\text{res}}} \eta_{r,d} + \gamma^{\text{finish}} \sum_{t \in T} (Z_{t,\tau_t} - Z_{t,\tau_t}^{\text{earliest}}) + \gamma^{\text{intra}} \sum_{t \in T} \sum_{u \in U_t \setminus \{1\}} \sigma_{t,u}^{\text{intra}} \\ & + \gamma^{\text{interw}} \sum_{t \in T} \sum_{v \in V \setminus \{1\}} \sigma_{t,v}^{\text{interw}}, \end{aligned} \quad (1)$$

where γ^{ext} , γ^{finish} , γ^{intra} , and γ^{interw} are scalar weights. The first term of the objective function gives the total time used of the extended service time windows, where $\eta_{r,d} = \max(\{S_{t,u} + P_{t,u,r}^{\text{end}} - W_{r,d}^{\text{end}} \mid t \in T, u \in U_t, r \in Q_{t,u}, Z_{t,u} = d\} \cup \{0\})$ for resource r and day d . The second term computes for each therapy t the deviation of the last treatment day from a lower bound $Z_{t,\tau_t}^{\text{earliest}}$ (see Maschler et al., 2016). The third objective term gives the total excess of the allowed deviation of the DTs' starting times to their respective nominal starting times, where $\sigma_{t,u}^{\text{intra}} = \max(|S_{t,u} - \widetilde{S}_{t,v}| - \delta^{\text{intra}}, 0)$ and $Z_{t,u} \in D'_v$. Each therapy's first DT is excluded from the above calculation since those are regarded in the specific situation at MedAustron as special. Finally, the last term computes the excess of the maximum intended time difference of the nominal starting times between two weeks and is calculated by $\sigma_{t,v}^{\text{interw}} = \max(|\widetilde{S}_{t,v} - \widetilde{S}_{t,v-1}| - \delta^{\text{interw}}, 0)$.

Note that the definition of DTs stated here differs from the one given in Maschler et al. (2016), where DTs are composed of consecutively executed activities that are related with minimum and maximum time lags. The simplification here is motivated by the fact that in practice the possibility to have different minimum and maximum time lags between two activities is not expected to be exploited in midterm planning. Consequently, time lags may either be replaced by “dummy” activities of fixed length or, as we do here, the subdivision of DTs into activities can be replaced by the time intervals $P_{t,u,r}$ specifying at which times which resources are needed.

4. Iterated greedy approach

Iterated greedy (IG) algorithms (Jacobs and Brusco, 1995) usually start with an initial solution and then repeatedly apply a destruction phase annulling part of the solution, followed by a construction phase that completes the solution again, until a termination criterion is reached. The initial solution is usually obtained by applying a construction heuristic. The destruction phase removes randomly selected components from the incumbent solution, which are then reinserted by a greedy reconstruction method in the construction phase. Afterward, an acceptance criterion is evaluated to determine whether the newly generated solution replaces the incumbent solution. Frequently, a local search algorithm is applied to the initial solution and after the construction phase to further boost the performance. In the following sections, we discuss the components of the proposed IG.

4.1. Initial solution

We presented in Maschler et al. (2016) the therapy-wise construction heuristic (TWCH) for the PTPSP without the extension that the starting times should be close to their weekly nominal starting times and that the nominal starting times belonging to the same therapy should not differ too much. Basically, this construction heuristic acts in two phases, first assigning all DTs to days (day assignment) and afterward determining the actual starting times of the DTs (time assignment). While the day assignment phase can be adopted unchanged, the time assignment phase has to be altered such that also the two new objective terms regarding the variation of the starting times are considered.

TWCH starts with the day assignment phase in which therapies are processed in the order of the latest possible starting day of their first DT. For a selected therapy, the corresponding DTs are then allocated sequentially to days, starting with the first DT. To this end, for the current DT all feasible days between the earliest and latest starting day with respect to the constraints imposed by the DT's predecessors are evaluated. The DT is assigned to the candidate day that minimizes on the one hand, the expected additional use of extended service windows by the current DT and all subsequent DTs and on the other hand the day at which the therapy is completed. A crucial aspect for the overall performance of TWCH is the estimation of the total time required from extended service windows. While an underestimation yields in general overfull days with avoidable use of extended service windows, an overestimation results in underused days and delayed ends of therapies. We studied the impact of this estimation in Maschler et al. (2018) and provide a surrogate model that exploits the structure of DTs and further problem knowledge. Experiments show that the surrogate model predicts the use of extended service windows quite accurately and improves our IG approach from Maschler et al. (2017) substantially. Consequently, we adopt this scheme here.

During the time assignment phase, the scheduling within the days has to be done. In other words, we have to find for all DTs starting times with as little use of extended service windows as possible that allow in addition nominal starting times that minimize the respective intraweek and interweek objective terms. In contrast to our earlier work, the schedules for the individual days cannot be regarded as independent but are coupled through the nominal starting times. A further changed property resulting from the extension of the PTPSP is that scheduling DTs in close succession might be suboptimal with respect to the two new objective terms. In fact, it might be necessary to have

breaks between two consecutive DTs. In principle, our approach to consider the DTs in a certain order and schedule each DT as early as possible has to change to the one where we have to decide in addition the duration of an optional gap between each pair of successive DTs. However, in practice the facility should be used at full capacity and, thus, it can be assumed that in general adding significant gaps between DTs immediately yields additional use of extended service time windows, which results in a worse objective value. Therefore, we restrict the construction heuristic and our overall approach to solutions without gaps that are not induced by resource availabilities.

Our approach for the time assignment part consists of two components executed in an interleaved way, one for scheduling the DTs on a considered day and one for setting and adapting the nominal starting times. To this end, the working days are processed in chronological order, starting with scheduling the DTs assigned to the first working day. After all starting times for a day have been determined, the nominal starting times of every considered therapy t in the current week v are updated as follows. Therapies' first DTs are ignored as they are excluded in the intraweek objective term. For each therapy's second DT, we set $\tilde{S}_{t,v}$ to $S_{t,2}$. For subsequent DTs u' assigned to the therapy's first week, the nominal starting time is set to the value that minimizes $\sum_{u=2}^{u'} \sigma_{t,u}^{\text{intra}w}$. Determining this value corresponds to finding the minimal value of a continuous piecewise linear function where the slope of the segments are multiples of $\gamma^{\text{intra}w}$. For DTs u' belonging to a therapy's second week and onward, the nominal starting time $\tilde{S}_{t,v}$ of the current week v is set to $\arg \min_{\tilde{S}_{t,v}} (\sum_{u \in \{D'_v | u \leq u'\}} \sigma_{t,u}^{\text{intra}w} + \sigma_{t,v}^{\text{inter}w})$. To this end, the nominal starting time of the previous week is considered as fixed. Note that for this reason the determined nominal starting times might not be optimal.

TWCH's component for scheduling DTs within a day as presented in Maschler et al. (2016) repeatedly places a not yet scheduled DT, selected using a priority function, as early as possible in the schedule until all DTs have been planned. The priority of the DTs is determined by a lexicographic combination of three criteria that consider the idle time that emerges on the beam resource, the earliest end of a regular service window from a required resource, and the ratio between the time the beam is required and the total processing time of the respective DT. These greedy criteria provide in practice reasonable performance with respect to minimizing the use of extended service windows, while yielding short processing times. Extending this lexicographic combination of criteria to respect also the intraweek and interweek objective terms is, however, not promising. The main difficulty is to balance between generating a tight schedule and prioritizing DTs that are close to their nominal starting time. While concentrating too much on the former aspect causes many deviations to the respective nominal starting times, focusing too much on the latter results often in extensive use of extended service windows.

To obtain a heuristic for scheduling DTs within a day that performs well on the extended problem formulation we shift the focus from which DT to schedule next to inserting DTs within the order of already scheduled DTs. A straightforward way to this would be to process the DTs assigned to a day in a particular order and test for each DT all positions of the already scheduled DTs. Finally, the DT is inserted in the position yielding the smallest objective value. This technique is analogous to the classic NEH algorithm by Nawaz et al. (1983) for the permutation flow shop problem (PFSP). However, preliminary experiments have shown that the performance of this simple insertion heuristic is worse compared to our original approach. The main reason is that the insertion of DTs postpones already inserted DTs, which might end up at a quite different time they have been

originally inserted. Hence, it makes sense to reevaluate the positions of already inserted DTs. In the PFSP literature, several heuristics have been proposed that extend NEH with reinsertions. We adapt here the FRB3 heuristic from Rad et al. (2009), which reevaluates and possibly reinserts after each insertion all already scheduled jobs.

Algorithm 1: FRB3 for the PTPSP

```

Input: A day  $d$ 
1  $\pi \leftarrow ()$ ;
2 let  $(\beta_1, \dots, \beta_n)$  be the DTs to schedule in decreasing order of  $p_{t,u}$  with
    $\beta_i \in \{(t, u) \mid t \in T, u \in U_t, Z_{t,u} = d\}$  for all  $i = 1, \dots, n$ ;
3 for  $i \leftarrow 1$  to  $n$  do
4   insert DT  $\beta_i$  in  $\pi$  resulting in the smallest objective value and in case of ties the smallest
   makespan;
5   for  $j \leftarrow 1$  to  $i$  do
6     extract and reinsert DT  $\pi_j$  at the position that results in the smallest objective value and in
     case of ties in the smallest makespan;
7   end
8 end

```

Algorithm 1 shows FRB3 adapted for scheduling DTs on a considered day d . It starts with an empty sequence π and considers then the DTs assigned to day d in the order of the largest processing times. We use here the same ordering as Rad et al. (2009) applied for FRB3 on the Permutation Flow Shop Problem (PFSP) with the C_{\max} (makespan) objective. Although our objective is quite different, preliminary experiments have shown that sorting according to the largest processing times is indeed effective. At Line 4, we test scheduling the current DT before each already scheduled DT and after the last one. The current DT is then inserted in the most promising position. In general, inserting a DT at position $l < i$ delays the starting times of the DTs π_{l+1}, \dots, π_i , while the starting times of the DTs π_1, \dots, π_{l-1} remain unchanged. Afterward, the position of all already scheduled DTs within π are reevaluated and possibly reinserted to accommodate the newly inserted DT in a better way.

DTs are inserted at the best position with respect to our objective function. However, as function (1) assumes a complete solution, we evaluate an objective function tailored for the time assignment part, which is

$$\begin{aligned}
 & \gamma^{\text{ext}} \sum_{r \in R} \max \left(\{S_{t,u} + P_{t,u,r}^{\text{end}} - W_{r,d}^{\text{end}} \mid i \in \{1, \dots, n\}, (t, u) = \pi_i, r \in Q_{t,u}\} \cup \{0\} \right) \\
 & + \sum_{\substack{i \in \{1, \dots, n\}, \\ (t,u) = \pi_i, u > 2}} \gamma^{\text{intra}} \cdot \sigma_{t,u}^{\text{intra}}.
 \end{aligned} \tag{2}$$

This function determines for the current sequence π the use of extended service windows and the excess of the allowed deviation from the nominal starting times. While the former term is analogous to (1), the latter term requires further considerations. As mentioned earlier, the nominal starting

times are updated after all starting times have been determined for a considered day. Consequently, for a therapy's second DT (i.e., the first for which the time difference to the nominal starting time is relevant) and every therapy's subsequent first DT of a week, we have not yet determined the corresponding nominal starting time. In the case where a DT is a therapy's second we can set $\sigma_{t,u}^{\text{intra}}$ to 0, as the respective nominal starting time can be set to the same value as the starting time of the current DT without inducing any cost. For a therapy's first DT within a week, we suppose that the nominal starting time will not differ more than δ^{interw} compared with the nominal starting time of the previous week. Hence, we regard in such cases starting times that differ more as $\delta^{\text{interw}} + \delta^{\text{intra}}$ to their respective nominal starting time of the previous week as excess. For all other cases, $\sigma_{t,u}^{\text{intra}}$ is computed as in the problem definition. To summarize, we calculate $\sigma_{t,u}^{\text{intra}}$ by

$$\sigma_{t,u}^{\text{intra}} = \begin{cases} 0 & \text{if } u = 2 \\ \max(|S_{t,u} - \tilde{S}_{t,v-1}| - \delta^{\text{interw}} - \delta^{\text{intra}}, 0) & \text{if } Z_{t,u} \in D'_v \wedge Z_{t,u-1} \in D'_{v-1} \\ \max(|S_{t,u} - \tilde{S}_{t,v}| - \delta^{\text{intra}}, 0) & \text{otherwise.} \end{cases} \quad (3)$$

In case more than one position evaluate to the same value by (2), we insert the DT that has the smaller makespan. The rationale behind the latter criterion is that in particular at the beginning of the algorithm many insertion points allow scheduling the sequence without use of extended service windows. Preferring a smaller makespan typically results in a tighter packed schedule and hopefully retains better options for the still to be inserted DTs.

4.2. Local search

The design of the neighborhood used within the IG's local search component depends on several factors. As real-world instances are expected to be quite large, the main challenge is to find neighborhoods that can be searched rather fast, still allowing a reasonable number of iterations of the IG to be completed, while improving the solution significantly in most cases. Due to typically small flexibility within the therapy process, assigning one DT to another day usually entails that also the therapy's preceding and succeeding DTs have to be reassigned to new days. Consequently, changing the day assignment of DTs affects the time assignment of several days, which results in a local search operator that is computationally too expensive to be applied within an iterated approach. Therefore, we restrict ourselves to a local search method that focuses on the time assignment part, that is, the allocation of the therapies' DTs on days is assumed to be fixed. In Maschler et al. (2017), this restriction allowed us to apply the local search for each day independently. The much smaller neighborhoods have shown to be crucial to receive adequate computation times for the local search component to be integrated within the IG. The presence of the therapies' nominal starting times and the extended objective function, however, links the starting times of the therapies' DTs: If we find a better starting time for a DT on a certain day, then we might also find a better nominal starting time, which again induces improvements on days that have been at a local optimum. To obtain a fast local search component, we consider first the nominal starting times as fixed and optimize the DTs' starting times until a local optimum with respect to a neighborhood for each individual day is reached. Afterward, we update the nominal starting times and repeat the first step until no further improvements are achieved. In the following, we explain both components in more detail.

In our scenario the DTs are heterogeneous regarding their time and resource requirements. Thus, moving DTs or exchanging the starting times of two DTs in a tightly scheduled day will lead in most cases to an infeasible solution. However, we can exploit the fact that each DT requires the beam resource exactly once to define a unique sequence of the DTs scheduled on a particular day. A solution is encoded by the sequence (π_1, \dots, π_n) resulting from sorting the DTs assigned to the currently considered day d , given by the set $\{(t, u) \mid t \in T, u \in U, Z_{t,u} = d\}$, in ascending order of the times from which on they use the beam resource B , that is, according to $S_{t,u} + P_{t,u,B}^{\text{start}}$. On the encoded days we can then define classical neighborhoods for sequencing problems. To evaluate neighbors we have to decode the corresponding sequences of DTs to obtain actual starting times. Algorithm 2 shows this decoding for a given sequence (π_1, \dots, π_n) of DTs and a working day $d \in D'$. The procedure starts by initializing each time marker C_r to the time the corresponding resource r becomes available. In the main loop, each DT is assigned in the order of the given sequence to the earliest possible start time at which all resources are available. First, at Line 3 the DT's start time $S_{t,u}$ is set to the earliest possible time such that all required resources are used after their corresponding time marker. At this time, the considered DT might still overlap with unavailability periods. If this is the case, the DT is delayed in the inner while loop until all required resources become available. At Line 7 the time markers C_r are set to the times when the corresponding resources become free after the just scheduled DT.

Algorithm 2: Procedure for determining starting times for a given sequence of DTs

```

Input: A day  $d$  and a sequence of DTs  $(\pi_1, \dots, \pi_n)$ 
1  $C_r \leftarrow W_{r,d}^{\text{start}} \quad \forall r \in R, d \in D_r^{\text{res}};$ 
2 for  $(t, u) \leftarrow \pi_1$  to  $\pi_n$  do
3    $S_{t,u} \leftarrow \max_{r \in Q_{t,u}} (C_r - P_{t,u,r}^{\text{start}});$ 
4   while  $\exists r \in Q_{t,u} \exists \overline{W}_{r,d,w} \in \overline{W}_{r,d} ([S_{t,u} + P_{t,u,r}^{\text{start}}, S_{t,u} + P_{t,u,r}^{\text{end}}] \cap \overline{W}_{r,d,w} \neq \emptyset)$  do
5      $S_{t,u} := \overline{W}_{r,d,w}^{\text{end}} - P_{t,u,r}^{\text{start}};$ 
6   end
7    $C_r \leftarrow S_{t,u} + P_{t,u,r}^{\text{end}} \quad \forall r \in Q_{t,u};$ 
8 end

```

To obtain an effective neighborhood, we have to take the problem structure into account. A fundamental property is that all DTs require the beam and one of the room resources. Moreover, the beam resource is used only during a part of the time the respective room resources are required. In a tight schedule, the beam usually cycles between the three treatment rooms as in this way the emerging idle time on the beam resource is minimal and the throughput of the facility is maximized. If we interrupt this interleaved execution of DTs by removing a single DT from the sequence, then in general the resulting gap on beam resource cannot be fully closed by decoding the remaining DTs. Consequently, classic insertion moves in which a single DTs is removed and reinserted in another position of the encoded sequence will rarely improve already tight schedules. Exchanging the position of two DTs, however, circumvents this situation. Therefore, we consider a neighborhood structure based on such exchanges.

The DT exchange neighborhood is defined for a day d on a sequence (π_1, \dots, π_n) of DTs by considering all pairs of DTs π_i and π_j , where $1 \leq i < j \leq n$. A move in this neighborhood results in a new sequence $(\pi_1, \dots, \pi_{i-1}, \pi_j, \pi_{i+1}, \dots, \pi_{j-1}, \pi_i, \pi_{j+1}, \dots, \pi_n)$ and is accepted if the decoded time assignment has a better objective value. The size of the neighborhood is $n(n-1)/2$.

The examination of the neighborhood is computationally costly due to the fact that decoding sequences of DTs as well as evaluating the objective function are both expensive operations. Hence, we exploit several aspects in order to accelerate the search for improvements. The first speedup is based on the observation that the starting times of the DTs $(\pi_1, \dots, \pi_{i-1})$ are not affected from the move with respect to the original sequence. Consequently, Algorithm 2 is modified to consider for each move only the DTs following π_{i-1} . This requires, however, to store all possible states of the time markers C_r for each position of the original sequence. The next acceleration aborts the decoding of a neighbor within Algorithm 2 if the considered move yields no improvement with a high probability. In particular, this is the case if a move worsens the interleaving of the DTs or forces a DT to be placed after an unavailability period. The resulting delay produces as a consequence usually an additional use of extended service periods and in general an increased objective value. Therefore, we prematurely terminate the main loop at Line 2 after processing DT π_{j+k} if its newly determined starting time is larger than in the original sequence. To this end, offset k is chosen such that the corresponding DT is the first that requires the same room resources as π_i , thus, $k \in \{1, 2, 3\}$. We use this offset to assess whether DT π_i interleaves well with its successors. As in the FRB3 algorithm for the initial solution, it suffices also here to consider an objective function tailored to the considered day. Thus, we evaluate equation (2), where $\sigma_{t,u}^{\text{intra}}$ can now be calculated as described in Section 3. Computing equation (2) from scratch can be done in $O(n_R + n \cdot n_Q + n)$ time, where n_Q is the maximal number of required resources by a DT. However, if we reuse the time markers C_r from Algorithm 2, the number of required steps decreases to $O(n_R + n)$.

After we reached a local optimum with respect to the considered neighborhood on each day, it might be that better suiting nominal starting times exist. Therefore, we assume now the set of all starting times S to be fixed and solve the following LP model to obtain new optimal nominal starting times. In the model, we use $\tilde{S}_{t,v}$ variables for the nominal starting times in \tilde{S} and nonnegative variables $\sigma_{t,u}^{\text{intra}}$ and $\sigma_{t,v}^{\text{inter}}$ to state the intraweek and interweek objective terms.

$$\min \gamma^{\text{intra}} \sum_{t \in T} \sum_{u \in U_t \setminus \{1\}} \sigma_{t,u}^{\text{intra}} + \gamma^{\text{inter}} \sum_{t \in T} \sum_{v \in V \setminus \{1\}} \sigma_{t,v}^{\text{inter}} \quad (4)$$

$$|S_{t,u} - \tilde{S}_{t,v}| - \delta^{\text{intra}} \leq \sigma_{t,u}^{\text{intra}} \quad \forall t \in T, \forall u \in U_t \setminus \{1\}, \forall v \in V : Z_{t,u} \in D'_v \quad (5)$$

$$|\tilde{S}_{t,v} - \tilde{S}_{t,v-1}| - \delta^{\text{inter}} \leq \sigma_{t,v}^{\text{inter}} \quad \forall t \in T, \forall v \in V \setminus \{1\} : \exists u \in U_t (Z_{t,u} \in D'_v) \quad (6)$$

$$\sigma_{t,u}^{\text{intra}} \geq 0 \quad \forall t \in T, \forall u \in U_t \quad (7)$$

$$\sigma_{t,v}^{\text{inter}} \geq 0 \quad \forall t \in T, \forall v \in V \quad (8)$$

$$\tilde{S}_{t,v} \in \mathbb{R} \quad \forall t \in T, \forall v \in V. \quad (9)$$

The model's objective function (4) corresponds to the objective function of the overall problem (1) restricted to the terms directly affected by the nominal starting times. Inequalities (5) enforce that the $\sigma_{l,u}^{\text{intra}w}$ variables are set to the excess of the maximum intended difference of the DTs' starting times to their respective nominal starting times. Finally, constraints (6) ensure that the $\sigma_{l,v}^{\text{inter}w}$ variables attain the time difference that exceed the maximum intended time difference of the nominal starting times between consecutive weeks.

4.3. Destruction and construction

In the destruction phase, the DTs of randomly selected therapies are removed from the schedule. In the subsequent construction phase, the DTs of the removed therapies are first assigned to days and afterward suitable starting times are determined. To this end, we used in Maschler et al. (2016) the proposed TWCH, which involved the time assignment part to be applied from scratch, completely ignoring the existing starting times of the kept DTs. In particular, with respect to the local search algorithm from Section 4.2, discarding the whole time assignment during destruction and construction is disadvantageous since much previous effort is wasted. Instead, we should try to transfer meaningful starting times as far as possible. To overcome this drawback, we adopt in Maschler et al. (2017) the NEH insertion heuristic of Nawaz et al. (1983) which preserves the sequence of unchanged DTs and inserts the removed ones greedily. Here we incorporate the extensions for NEH presented by Rad et al. (2009).

In principle there are many options for the destruction phase. Since we keep the assignment of DTs to days fixed within the local search, it is especially important to allow such modifications during the destruction and construction phase. Due to the in general tight constraints on the day assignment level, it can be expected that the therapies' DTs are planned in close succession. Consequently, removing the assignments of single DTs within a therapy will have in general only limited effect as their assignments are usually determined by the remaining ones. In contrast, removing all assignments of some therapy allows much more flexibility like moving the therapy's start to another week. A further aspect is the greedy behavior of the TWCH's day assignment phase that acts like the first fit heuristic until it detects that another first fit assignment will induce use of extended service windows. At this point, TWCH starts to actively delay and stretch the day assignments of the whole therapy as long it is beneficial with respect to the objective function. Thus, the destruction of the assignments of entire therapies allows TWCH revising poor decisions. Although one could think of different goal-driven selection strategies for therapies to remove, we choose to select random ones, since the destruction phase is our main source of diversification. To summarize, the deconstruction phase invalidates the day and time assignments of $n^{\text{ig-dest}}$ randomly selected therapies. To increase the robustness of the algorithm we do not keep $n^{\text{ig-dest}}$ fixed for all iterations, but sample a new value for $n^{\text{ig-dest}}$ from a discrete uniform distribution $\beta^{\text{ig-dest}}$ at the beginning of every destruction phase.

The construction phase starts with an application of TWCH's day assignment on the destroyed therapies. Afterward, the respective DTs are inserted in a randomized order into the schedule. In the time assignment phase for the initial solution, we use FRB3 for scheduling the DTs. After each insertion, all already scheduled DTs are reevaluated and possibly reinserted. In contrast to the initial solution, the construction phase is executed for many times. Hence, the IG's construction

phase is much more time critical and compared to FRB3 a less exhaustive approach is needed. Rad et al. (2009) proposed, among others, the FRB 4_k algorithm, which is conceptually between NEH and FRB3 in that it reconsiders only the k DTs around each inserted DT. FRB 4_k is based on the assumption that reevaluating the immediate neighbors has the largest effect. To be precise, we receive FRB 4_k if we modify the inner loop of Algorithm 1 such that it reevaluates the DTs at the positions $\max(1, l - k)$ to $\min(n, l + k)$, where l is the position of the previously inserted DT at Line 4. Moreover, unlike Algorithm 1 we start FRB 4_k within the construction phase not with an empty sequence π , but with the sequence resulting from sorting the not removed DTs according to the first time they use the beam resource (see Section 4.2). The nominal starting times are set as described in Section 4.1, followed by solving models (4)–(9) after all starting times have been determined.

5. Computational study

In this section, we perform an experimental evaluation of the proposed enhanced IG approach, which we call from here on EIG. As a reference method, we use a variant of the IG from Maschler et al. (2016), denoted as IG-LI, for the extended problem formulation. In the experiments, we start with IG-LI and replace its components step by step with the ones from EIG. The aim is to investigate the properties and impacts of the individual proposed enhancements and to demonstrate that their combination yields indeed an improved metaheuristic.

The used artificial benchmark instances are related to the expected situation at MedAustron and are available at <http://www.ac.tuwien.ac.at/research/problem-instances>. We consider instances with 50, 70, 100, 150, 200, and 300 therapies. The used naming schema encodes first the number of therapies followed by a consecutive number. The length of the planning horizon is derived from the number of therapies considered in the instance by roughly estimating the number days required to perform all DTs. Afterward, windows of 14 days are sampled from the planning horizon in which the therapies have to start. A characteristic of the instances is that there is a ramp-up phase of a few weeks after which the facility is used at full capacity followed by a wind-down phase near the end of the planning horizon. Our central resources are the beam and the rooms as they are required by each DT. They have regular service time windows on each weekday of 14 hours starting from $\widetilde{W}_d^{\text{start}}$ followed by extended service time windows of another 10 hours. There are further resources, like the personnel, that are regularly available only for a part of these 14 hours. All those further resources are sufficiently dimensioned to be not the primary source of substantial use of extended service time. At full capacity, the facility is able to perform around 60 DTs on each working day. To keep the instances with 50 and 70 therapies challenging, we halve the working days, that is, the extended service time windows starts for the resources at latest seven hours after $\widetilde{W}_d^{\text{start}}$. For more details on the instance generation, see Maschler et al. (2016). We use here the benchmark instances generated for Maschler et al. (2017), which we created for two reasons. On the one hand, Maschler et al. (2016) considered DTs composed of activities associated with minimum and maximum time lags. However, as already mentioned, this feature is not considered as relevant in our real-world midterm planning application. On the other hand, many of the instances from Maschler et al. (2016) had rather unrealistically strict constraints concerning the starting days of therapies, which made an extensive use of extended service time windows unavoidable.

All algorithms have been implemented in C++14 and compiled with G++ 6.3.0, and all experiments were carried out using a single core of an Intel Xeon E5-2640 v4 CPU with 2.40 GHz. The LP models for determining the nominal starting times have been solved with Gurobi 7.5. We adopt the acceptance criterion and the termination condition from Maschler et al. (2016): The incumbent solution is replaced by a current new solution iff the latter has a smaller objective value, and the total CPU time is limited to 20 minutes, respectively. In the objective function, we use the weights $\gamma^{\text{ext}} = 1/60$, $\gamma^{\text{finish}} = 1/100$, $\gamma^{\text{intra}} = 1/600$, and $\gamma^{\text{inter}} = 1/600$. The intuition behind these values is as follows. The time resolution of the instances is in minutes. Weight γ^{ext} is set such that the use of one hour from an extended service time window corresponds to one unit in the objective function (1). The weight for the second objective term γ^{finish} reflects the fact that the completion of a therapy should usually be delayed if performing a DT entirely within extended time can be avoided. As already mentioned, providing the therapies' DTs within the allowed variance is not of medical relevance and is consequently a subordinate goal. Therefore, we set both γ^{intra} and γ^{inter} to a 10th of γ^{ext} . A used hour of extended service windows is defined to be equally bad as 10 hours of excess from the allowed variance between the starting times of the therapies.

To use IG-LI as a reference algorithm, we have to provide an extension that determines in addition the nominal starting times. This can be done by solving the LP model presented in Section 4.2 for the initial solution provided by TWCH and at the end of each construction phase. Moreover, like in the local search procedure presented in Section 4.2, we repeatedly apply the local improvement method followed by a recalculation of the nominal starting times until no improvements can be found. As already stated in Section 4.1, it is not straightforward how to extend the time assignment part of TWCH such that it explicitly respects the additional objective terms. However, due to the behavior of the time assignment to prioritize DTs with certain properties, we can observe quite frequently that DTs belonging to the same therapy are scheduled roughly at the same time anyway.

To show that the IG approach presented in this work indeed enhances IG-LI, we will gradually exchange components of IG-LI with the ones presented here. In this way, we assess the properties of the individual components and their interplay. We start by exchanging in IG-LI the local improvement component with the local search procedure from Section 4.2. We call the resulting algorithm IG-LS. Afterward, we interchange in addition the destruction and construction phase, which is named IG-DRLS. Finally, we swap in IG-DRLS the construction heuristic for the initial solution and obtain the final EIG. Note that we receive all meaningful variants between IG-LI and EIG by exchanging the metaheuristic's components in this order. Starting with applying the proposed local search method instead of the local improvement makes sense since it is performed last within each iteration. In contrast, starting with replacing the destruction and construction component with the one from Section 4.3 yields a conceptually flawed variant. While the construction phase aims at keeping relative timing characteristics of not removed DTs, the local improvement operator ignores the time assignment provided by the construction phase. Hence, either the construction phase or the local improvement becomes dominant and ignores the influence of the other. Exchanging the construction heuristic for the initial solution after the construction phase is also reasonable for the same argument: In an IG variant using the newly assembled construction heuristic combined with the construction phase of IG-LI, the first successful iteration would completely revoke the time assignment for the initial solution.

In a preliminary study, we experimented with several neighborhoods in place of the exchange neighborhood within the EIG. We considered two variants of insertion neighborhoods. The moves

of the first consist of removing and reinserting a DT in another position in the sequence obtained by decoding a considered day. Besides being larger than the exchange neighborhood, insertion moves are less likely to improve the solution due to the interleaving of DTs with respect to the used rooms in good solutions. The second considered insertion neighborhood removes and reinserts DTs, but changes only the positions of the DTs requiring the same room resource. The exchange neighborhood performs better for two reasons. On the one hand, DTs have in general different timing characteristics and resource requirements. Thus, keeping the positions of the DTs requiring other room resources fixed can cause disruptions in the schedule as well. On the other hand, if we only modify the positions of the DTs requiring the same room, then the starting times of the affected DTs change typically to larger degree. This may cause frequently an increase in the objective terms regarding the nominal starting times. Moreover, we also tried a neighborhood based on inversions of parts of the DT sequence on a considered day. Clearly, inversions of sequences of two or three subsequent DTs are already covered by exchanges. Inversion of very long sequences of DTs are very likely to accumulate large costs from the deviations of nominal starting times. Therefore, we considered inverting sequences up to a given maximal length. Although being smaller than the exchange neighborhood and, hence, allowing more iterations of the IG, it turned out that the overall performance is weaker compared to the presented approach. Most likely many improving exchanges of more distant DTs with respect to the job sequence cannot be replicated by inversions. Finally, we observed for the presented local search method that the next improvement strategy converges, in general, significantly faster than the best improvement strategy, while yielding similarly good solutions.

The preliminary results of this work presented in Maschler et al. (2017) focused on the first two steps. The experiments indicate that exchanging just the local improvement operator of the IG-LI with the local search component does not yield a substantial improvement. If, however, both local search and new construction phase are used together, then the resulting approach clearly dominates IG-LI. The main reason for this performance improvement is the interplay between the construction phase and the local search procedure. On the one hand, the local search operator is, in general, able to provide better results than IG-LI's local improvement operator. However, encoding, decoding, and evaluating the solution is computationally demanding and, hence, converging to a local optimum is time-consuming, especially on strongly perturbed solutions. On the other hand, the construction phase is designed in such a way that large parts of the sequence of the DTs are preserved while introducing the removed DTs in a sensible but randomized way. Starting with a solution close to a local optimum with respect to the DT exchange neighborhood allows to reduce the time spent in the local search procedure and, consequently, increases the total number of iterations. Although these experiments have been conducted on the original version of the PTPSP and with a slightly simplified algorithm, these results can be replicated also for the problem at hand. Therefore, we consider here only IG-DRLS further.

The metaheuristics' strategy parameters were tuned using the automatic parameter configuration tool *irace* (López-Ibáñez et al., 2016) in version 2.4. In detail, *irace* was applied in two rounds for each instance size separately. To this end, we used an independent set of instances designated for tuning and a computational budget of 2000 experiments for each application of *irace*. In the first round, we used *irace* with a larger amount of parameters for determining an effective design of the algorithm. During this configuration phase, we employed $FRB4_k$ instead of $FRB3$ within the construction heuristic for the initial solution. The values obtained for parameter k for the initial

Table 1
Parameter settings for IG-LI, IG-DRLS, and EIG determined by irace

Instance size n_T	IG-LI			IG-DRLS		EIG	
	$\beta^{\text{ig-dest}}$	$n^{\text{rta-noimp}}$	$k^{\text{rta-rand}}$	$\beta^{\text{ig-dest}}$	k^{frb4}	$\beta^{\text{ig-dest}}$	k^{frb4}
50	$\mathcal{U}\{2, 18\}$	1.881	2	$\mathcal{U}\{2, 2\}$	67	$\mathcal{U}\{2, 4\}$	24
70	$\mathcal{U}\{6, 7\}$	2.300	2	$\mathcal{U}\{2, 3\}$	46	$\mathcal{U}\{2, 9\}$	16
100	$\mathcal{U}\{6, 10\}$	0.513	7	$\mathcal{U}\{3, 3\}$	2	$\mathcal{U}\{3, 3\}$	1
150	$\mathcal{U}\{3, 8\}$	0.660	9	$\mathcal{U}\{2, 2\}$	12	$\mathcal{U}\{2, 2\}$	12
200	$\mathcal{U}\{6, 9\}$	0.598	6	$\mathcal{U}\{2, 3\}$	17	$\mathcal{U}\{2, 2\}$	14
300	$\mathcal{U}\{29, 45\}$	2.420	2	$\mathcal{U}\{2, 5\}$	25	$\mathcal{U}\{2, 2\}$	44

solution corresponded with the number of DTs assigned to a fully utilized day, that is, $k \approx n$, and thus FRB4_k degenerates to FRB3 . This suggests that the comprehensiveness and the implied computational costs of FRB3 are justified and beneficial for the overall approach. For the local search method, we tested randomizing the order in which the local search examines the neighboring solutions. It turned out that this randomization is favorable over considering moves in the order of the starting times of respective DTs. From a theoretical point of view, the randomization of the order of the considered moves removes a bias toward exchanges at the beginning of days. Moreover, we raced whether to activate the accelerations in the local search that prematurely terminate the evaluation of a neighbor. While the technique described in Section 4.2 has been activated on all instance sizes, a complementary method considering increased variations to the nominal starting times has been rejected. In the second round of the algorithm configuration, we kept the assessed design choices fixed to focus on the central parameters: $\beta^{\text{ig-dest}}$, $n^{\text{rta-noimp}}$, $k^{\text{rta-rand}}$ for IG-LI and $\beta^{\text{ig-dest}}$, k^{frb4} for IG-DRLS and EIG. As described in Section 4.3, $\beta^{\text{ig-dest}}$ specifies a discrete uniform distribution $\mathcal{U}(a, a + b)$ from which in each iteration a random number of therapies to destroy is sampled. The parameter configuration determined values for a and b , where $a \in \{1, \dots, 40\}$ and $b \in \{0, \dots, 40\}$. For $n^{\text{rta-noimp}}$ and $k^{\text{rta-rand}}$ we used the value ranges $[0.5, 2.5]$ and $\{1, \dots, 10\}$, respectively. We denote with k^{frb4} the parameter k of the FRB4_k algorithm used in the construction phase and consider values from $\{0, \dots, 70\}$. The resulting parameter configurations are shown in Table 1.

Table 2 depicts for IG-LI, IG-DRLS, and EIG averages of the final objective values \overline{obj} and the corresponding standard deviation $\sigma(obj)$ over 30 runs for each of the 30 benchmark instances. Moreover, Table 2 gives the probability values obtained by an application of an one-tailed Wilcoxon rank sum test on the objective values of two methods at a time. We start by comparing IG-LI with IG-DRLS. The average objective values of IG-DRLS are on 29 of 30 benchmark instances smaller than those obtained by IG-LI. The Wilcoxon rank sum test indicated with a confidence level of 95% that IG-DRLS yields significantly better solutions than IG-LI on 29 benchmark instances, the only exception is instance 050-05. In fact, on 21 instances, the average objective values are halved compared with the ones from IG-LI, and on eight instances the average objective values of IG-DRLS are even 75% smaller compared with those of IG-LI. These results confirm the outcome of the experiments conducted for the original variant of the PTPSP in Maschler et al. (2017), which showed the advantage of applying the new construction and local search components.

Table 2
Comparison of IG-LI, IG-DRLS, and EIG

Instance	IG-LI (1)		IG-DRLS (2)		EIG (3)		Wilcoxon rank sum test		
	\overline{obj}	$\sigma(obj)$	\overline{obj}	$\sigma(obj)$	\overline{obj}	$\sigma(obj)$	$p_{1 \leq 2}$	$p_{1 \leq 3}$	$p_{2 \leq 3}$
050-01	8.570	1.323	2.784	0.562	2.608	0.215	0.000	0.000	0.027
050-02	52.830	2.228	40.005	3.006	39.812	2.398	0.000	0.000	0.339
050-03	47.600	2.022	37.281	3.821	37.660	3.252	0.000	0.000	0.733
050-04	29.561	1.564	26.419	3.743	25.795	3.152	0.000	0.000	0.261
050-05	53.363	1.996	54.089	2.761	49.848	2.725	0.893	0.000	0.000
070-01	28.600	2.826	13.242	1.742	14.015	1.787	0.000	0.000	0.945
070-02	48.346	2.861	44.871	5.588	41.540	4.556	0.003	0.000	0.007
070-03	74.141	3.957	70.376	4.897	70.227	5.833	0.000	0.001	0.524
070-04	13.826	1.794	7.710	1.907	7.313	1.556	0.000	0.000	0.282
070-05	56.838	4.016	49.328	4.593	49.316	3.518	0.000	0.000	0.371
100-01	81.405	5.025	18.383	2.296	17.840	2.524	0.000	0.000	0.075
100-02	108.194	5.006	24.366	2.041	23.552	2.179	0.000	0.000	0.099
100-03	53.978	3.251	15.564	1.592	15.306	1.457	0.000	0.000	0.275
100-04	88.931	5.892	19.703	1.478	20.865	1.694	0.000	0.000	0.992
100-05	77.063	3.895	14.762	1.256	16.089	1.235	0.000	0.000	1.000
150-01	128.564	7.308	40.590	5.421	35.605	5.399	0.000	0.000	0.001
150-02	273.450	10.896	80.328	4.877	72.008	5.785	0.000	0.000	0.000
150-03	127.236	5.736	72.212	6.622	62.368	5.999	0.000	0.000	0.000
150-04	116.294	5.290	32.255	2.126	27.787	1.744	0.000	0.000	0.000
150-05	96.049	6.785	22.889	2.019	20.093	1.619	0.000	0.000	0.000
200-01	194.440	9.204	76.943	5.658	61.244	5.783	0.000	0.000	0.000
200-02	223.821	7.241	88.747	5.988	78.073	6.780	0.000	0.000	0.000
200-03	165.977	8.092	57.843	4.214	49.197	3.277	0.000	0.000	0.000
200-04	212.980	9.355	63.085	7.408	51.859	6.435	0.000	0.000	0.000
200-05	188.749	7.701	44.143	3.253	33.294	2.572	0.000	0.000	0.000
300-01	239.586	5.072	52.349	4.503	35.082	3.149	0.000	0.000	0.000
300-02	344.637	9.689	133.092	7.858	128.118	10.560	0.000	0.000	0.028
300-03	249.621	10.444	64.612	5.839	48.084	3.599	0.000	0.000	0.000
300-04	370.461	9.514	162.315	8.896	115.586	8.354	0.000	0.000	0.000
300-05	251.234	7.381	48.194	2.936	35.885	2.312	0.000	0.000	0.000

We consider for each instance and every approach average objective values \overline{obj} of 30 runs and corresponding standard deviations $\sigma(obj)$. Moreover, the p -values originating from an application of the Wilcoxon rank sum test on pairs of algorithms are given. To this end, we denote with $p_{A \leq B}$ the p -values under the null hypothesis that approach A performs better than or equal to method B . We mark for each instance the best average objective value and p -values smaller than 0.05 in bold.

In fact, the superiority of IG-DRLS over IG-LS is here even more predominant. This can be explained by the fact that we use here in contrast to Maschler et al. (2017) FRB4_k instead of NEH (i.e., FRB4₀). Moreover, the new construction operator as well as the local search operator are able to handle the more complicated objective much better than their counterparts within IG-LI.

We continue by taking our main approach into consideration. Table 2 clearly shows that EIG outperforms the two other metaheuristics, and provides the best average objective values on 26 of 30 benchmark instances. As before, we applied a Wilcoxon rank sum test on the objective values

Table 3

The breakdown of the average objective values of 30 runs for IG-LI, IG-DRLS, and EIG presented in Table 2 into the first two and the last two terms of our objective function (1) denoted as $\overline{ext + \overline{fin}}$ and $\overline{iaw + \overline{iew}}$, respectively. Best values for each objective term are marked in bold

Instance	IG-LI		IG-DRLS		EIG	
	$\overline{ext + \overline{fin}}$	$\overline{iaw + \overline{iew}}$	$\overline{ext + \overline{fin}}$	$\overline{iaw + \overline{iew}}$	$\overline{ext + \overline{fin}}$	$\overline{iaw + \overline{iew}}$
050-01	2.834	5.735	2.770	0.014	2.602	0.006
050-02	37.729	15.100	37.613	2.392	36.908	2.903
050-03	33.642	13.958	34.373	2.908	34.533	3.127
050-04	20.834	8.726	24.393	2.026	23.694	2.100
050-05	40.899	12.464	50.447	3.642	46.279	3.569
070-01	16.812	11.788	12.934	0.308	13.615	0.400
070-02	32.592	15.754	42.868	2.003	38.572	2.968
070-03	55.566	18.574	67.096	3.280	65.863	4.364
070-04	6.325	7.501	7.461	0.249	6.769	0.544
070-05	41.663	15.175	47.482	1.846	46.451	2.866
100-01	23.018	58.387	13.869	4.514	13.422	4.418
100-02	31.548	76.646	17.166	7.199	16.142	7.410
100-03	10.263	43.715	10.894	4.670	10.413	4.893
100-04	26.416	62.514	13.340	6.364	14.018	6.847
100-05	24.704	52.359	9.398	5.364	11.085	5.004
150-01	34.650	93.914	28.387	12.203	26.102	9.504
150-02	134.003	139.447	59.588	20.740	52.431	19.577
150-03	51.618	75.619	59.815	12.396	48.742	13.626
150-04	24.568	91.727	20.950	11.305	17.758	10.028
150-05	24.652	71.396	14.471	8.417	13.901	6.192
200-01	66.198	128.243	52.634	24.309	39.416	21.828
200-02	86.416	137.405	67.024	21.723	56.513	21.561
200-03	36.428	129.549	35.604	22.239	29.719	19.478
200-04	67.216	145.764	45.907	17.178	34.686	17.174
200-05	46.356	142.393	28.765	15.378	20.421	12.873
300-01	34.580	205.006	29.479	22.870	20.496	14.586
300-02	131.749	212.888	96.046	37.046	90.977	37.141
300-03	42.013	207.609	39.849	24.764	30.603	17.481
300-04	120.678	249.783	115.095	47.220	76.253	39.334
300-05	32.446	218.789	24.558	23.636	21.373	14.512

with a confidence level of 95% for each instance to compare EIG with IG-LI and EIG with IG-DRLS. The former tests showed that EIG performs significantly better on all benchmark instances. The latter series of statistical tests evinced that the EIG outperforms IG-DRLS on 18 benchmark instances significantly, while the observed better average objective values of IG-DRLS have been two times significant. Furthermore, there is clear tendency that EIG significantly performs better than IG-DRLS on larger benchmark instances. This indicates that exchanging the construction heuristic for the initial solution with the one described in Section 4.1 becomes of greater importance with the increasing size of the instances. The reason for this is that with larger instance sizes also the computational cost for each iteration of the IG increases and, consequently, the number of executed iterations decreases. Hence, quality of the initial solution becomes with larger instance sizes more

and more important. This argument is further supported by the fact that the construction heuristic presented here requires substantially more of the total time budget of 20 CPU minutes than the application of TWCH: The average computation time for the construction heuristic based on FRB3 is on the largest instances 29.8 seconds, while the computation of the initial solution for IG-LI and IG-DRLS takes on average only 0.25 seconds. On this account, it becomes more evident that the combination of TWCH with FRB3 is indeed advantageous and that the large processing time is well spent.

Table 3 gives a more detailed breakdown of the average objective values presented in Table 2. To this end, we denote with $\overline{ext} + \overline{fin}$ the part of the average objective values originating from the use of extended service time windows and from the delayed completion of therapies. The sum of these two objective terms correspond to the objective function used in the original variant of the PTPSP. Moreover, $\overline{iaw} + \overline{iaw}$ stands for the part of the average objective values that arise from the intraweek and interweek objective terms. For a well-performing method, it is clearly not sufficient to focus on only one part of the objective function while neglecting the other aspects of the problem. This is especially visible for the smaller benchmark instances. On the instances with 50 and 70 therapies, IG-LI frequently provides solutions with the smallest costs with respect to the first two terms of objective function (1). This comes, however, with a much higher cost on the intraweek and interweek objective terms compared with the other two approaches. Furthermore, IG-DRLS is on several occasions able to provide solutions with less excess on the allowed variations of the DTs' starting times. Nevertheless, EIG outperforms the other two approaches on almost all cases due to the better balance between the objective parts. On the larger benchmark instances, the superiority of EIG over IG-LI and IG-DRLS becomes more evident. For most benchmark instances with more than 100 therapies, the EIG metaheuristic is able to provide the best results on both parts of the objective function.

6. Conclusions

In this paper, we presented an extended version of the PTPSP in which the starting time variations of DTs belonging to the same therapy should not exceed specified thresholds. To this end, we introduced nominal starting times that serve as a reference point for computing the variation within and between weeks. From a practical viewpoint, this extension increases the difficulty of the problem substantially since on the one hand the calculation of the nominal starting times requires the DTs' starting times, while on the other hand finding good starting times involves knowing the nominal starting times. Moreover, minimizing the variation of starting times is frequently contrary to our main objective, which is to minimize the use of extended service time windows. Consequently, also the design of an effective metaheuristic is more involved as it requires to balance the different aspects of the objective function.

To tackle the problem at hand, we provide an IG metaheuristic for the PTPSP that enhances the IG from our previous work (Maschler et al., 2016). The approach features a construction heuristic that combines parts of the TWCHs from Maschler et al. (2016) and the FRB heuristics from Rad et al. (2009). Moreover, a local search technique is applied that alternately examines an exchange neighborhood to improve the DTs starting times and solves an LP model that computes nominal starting times. In contrast to our previous IG, the presented approach is able to preserve the order of

the not removed DTs on the individual days. The resulting advantage is that far more information from the incumbent solution is retrieved.

To evaluate the performance of the proposed IG, we started by extending the IG from our previous work to cover the new PTPSP variant. Afterward, we replaced components of this reference algorithm step by step with the newly described ones to assess their properties. The enhancements are twofold. First, the interplay between its construction phase and the applied local search method: The local search procedure yields, in general, better results than applying TWCH's randomized time assignment iterated for many times. However, due to the required encoding and decoding steps, evaluating neighbors is time-consuming. Hence, to ensure that the metaheuristic is able to perform sufficiently many iterations, it is required that the neighborhood requires on average only a few steps until it reaches a local optimum. To this end, we apply in the construction phase an insertion heuristic based on FRB4_k that iteratively places the removed DTs into the permutation resulting from sorting the DTs according to the times at which they use the beam. Second, although being computationally expensive, the newly presented construction heuristic for the initial solution based on TWCH and FRB3 gives the IG a superior starting point. The resulting approach outperforms our reference method on all benchmark instances significantly.

A remaining challenge is to determine suitable parameter configurations for real-world instances. Although the considered benchmark instances aim at modeling the expected situation at MedAustron, they contain assumptions and simplifications which might differ in the future practice. The main characteristic of the used benchmark instances is the number of therapies that have to be scheduled. Our experiments showed that the values of some parameters are highly dependent on the instance size. For real-world instances, it is likely that there are also other aspects that have to be considered for obtaining good parameter configurations. Hence, the next step to improve the applicability of the presented approach is to define a parameter model that determines values for the IG's strategy parameters on the basis of the observed instance characteristics.

PTPSP, as defined here, still simplifies the midterm planning part arising in practice. In particular, therapies are restricted here to consist only of DTs. In the general case, however, there is a treatment planning phase preceding all DTs in which the DTs are prepared. Since other constraints have to be enforced for these tasks, they cannot be modeled as DTs. There are further activities (e.g., control examinations) complementing the core therapies that have to be provided once a week before or after one of the DTs. However, from a practical viewpoint these additional activities should never influence the throughput of the facility and can be sufficiently well handled in a postprocessing step.

We use in this work an acceptance criterion that accepts a current solution if it has a better objective value. An obvious next step is to also consider acceptance criteria that allow selecting suboptimal solutions, for example, in a simulated annealing like fashion (see Ruiz and Stützle, 2007). Moreover, the proposed local search procedure exchanges the DTs only within days. Neighborhoods that can alter the days on which a therapy is applied seem promising. The main challenge here is to design and restrict the neighborhoods such that the resulting local search method is still fast enough to allow a sufficiently large number of IG iterations. We formulated PTPSPs as a single-objective optimization problem using a linear combination of the objective goals. A further natural next step would be to consider PTPSP as a multiobjective optimization problem.

Acknowledgments

We thank EBG MedAustron GmbH, Wiener Neustadt, Austria, for the collaboration on particle therapy patient scheduling and partially funding this work.

References

- Burke, E.K., Rocha, P.L., Petrovic, S., 2011. An integer linear programming model for the radiotherapy treatment scheduling problem. Available at <http://arxiv.org/pdf/1103.3391.pdf>.
- Conforti, D., Guerriero, F., Guido, R., 2008. Optimization models for radiotherapy patient scheduling. *4OR* 6, 3, 263–278.
- Hartmann, S., Briskorn, D., 2010. A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research* 207, 1, 1–14.
- Horn, M., Raidl, G.R., Blum, C., 2017. Job sequencing with one common and multiple secondary resources: a problem motivated from particle therapy for cancer treatment. In Giuffrida, G., Nicosia, G., Pardalos, P., Umeton, R. (eds) *The Third International Conference on Machine Learning, Optimization and Big Data—MOD 2017*, Springer International Publishing, Basel.
- Jacobs, L.W., Brusco, M.J., 1995. A local-search heuristic for large set-covering problems. *Naval Research Logistics* 42, 7, 1129–1140.
- Kapamara, T., Petrovic, D., 2009. A heuristics and steepest hill climbing method to scheduling radiotherapy patients. Proceedings of the International Conference on Operational Research Applied to Health Services (ORAHs), Catholic University of Leuven, Leuven, Belgium.
- Kapamara, T., Sheibani, K., Haas, O., Petrovic, D., Reeves, C., 2006. A review of scheduling problems in radiotherapy. Proceedings of the International Control Systems Engineering Conference (ICSE), Coventry University Publishing, Coventry, pp. 207–211.
- López-Ibáñez, M., Dubois-Lacoste, J., Cáceres, L.P., Birattari, M., Stützle, T., 2016. The irace package: iterated racing for automatic algorithm configuration. *Operations Research Perspectives* 3, 43–58.
- Maschler, J., Hackl, T., Riedler, M., Raidl, G.R., 2017. An enhanced iterated greedy metaheuristic for the particle therapy patient scheduling problem. Proceedings of the 12th Metaheuristics International Conference, Barcelona, Spain, pp. 465–474.
- Maschler, J., Riedler, M., Raidl, G.R., 2018. Particle therapy patient scheduling: Time estimation for scheduling sets of treatments. Computer Aided Systems Theory—EUROCAST 2017, Springer International Publishing, Basel, pp. 364–372.
- Maschler, J., Riedler, M., Stock, M., Raidl, G.R., 2016. Particle therapy patient scheduling: first heuristic approaches. Proceedings of the 11th International Conference on the Practice and Theory of Automated Timetabling, Udine, Italy, pp. 223–244.
- Nawaz, M., Enscore, E.E., Ham, I., 1983. A heuristic algorithm for the m -machine, n -job flow-shop sequencing problem. *Omega* 11, 1, 91–95.
- Petrovic, D., Morshed, M., Petrovic, S., 2009. Genetic algorithm based scheduling of radiotherapy treatments for cancer patients. *Proceedings of the Conference on Artificial Intelligence in Medicine (AIME)*, Vol. 5651, Springer, Berlin, 101–105.
- Petrovic, D., Morshed, M., Petrovic, S., 2011. Multi-objective genetic algorithms for scheduling of radiotherapy treatments for categorised cancer patients. *Expert Systems with Applications* 38, 6, 6994–7002.
- Petrovic, S., Leite-Rocha, P., 2008. Constructive and GRASP approaches to radiotherapy treatment scheduling. Advances in Electrical and Electronics Engineering—IAENG Special Edition of the World Congress on Engineering and Computer Science 2008, IEEE, Los Alamitos, CA, pp. 192–200.
- Petrovic, S., Leung, W., Song, X., Sundar, S., 2006. Algorithms for radiotherapy treatment booking. 25th Workshop of the UK Planning and Scheduling Special Interest Group, Nottingham, pp. 105–112.
- Rad, S.F., Ruiz, R., Boroojerdian, N., 2009. New high performing heuristics for minimizing makespan in permutation flowshops. *Omega* 37, 2, 331–345.

- Riedler, M., Jatschka, T., Maschler, J., Raidl, G.R., 2020. An iterative time-bucket refinement algorithm for a high-resolution resource-constrained project scheduling problem. *International Transactions in Operational Research* 27, 1, 573–613.
- Riff, M.C., Cares, J.P., Neveu, B., 2016. RASON: a new approach to the scheduling radiotherapy problem that considers the current waiting times. *Expert Systems with Applications* 64, 287–295.
- Ruiz, R., Stützle, T., 2007. A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research* 177, 3, 2033–2049.
- Vogl, P., Braune, R., Doerner, K.F., 2018. A multi-encoded genetic algorithm approach to scheduling recurring radiotherapy treatment activities with alternative resources, optional activities, and time window constraints. *Computer Aided Systems Theory—EUROCAST 2017*, Springer International Publishing, Basel, pp. 373–382.