FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

# Local Search Methods for the Particle Therapy Patient Scheduling Problem

DIPLOMARBEIT

zur Erlangung des akademischen Grades

**Diplom-Ingenieur**

im Rahmen des Studiums

**Masterstudium Software Engineering/Internet Computing**

eingereicht von

**Thomas Hackl, BSc**
Matrikelnummer 0927710

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Günther Raidl
Mitwirkung: Projektass. Dipl.-Ing. Johannes Maschler, BSc
Univ.-Ass. Dipl.-Ing. Martin Riedler, BSc

Wien, 13. August 2018

Thomas Hackl | Günther Raidl

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

# Local Search Methods for the Particle Therapy Patient Scheduling Problem

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

**Diplom-Ingenieur**

in

**Software Engineering/Internet Computing**

by

**Thomas Hackl, BSc**
Registration Number 0927710

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Günther Raidl
Assistance: Projektass. Dipl.-Ing. Johannes Maschler, BSc
Univ.-Ass. Dipl.-Ing. Martin Riedler, BSc

Vienna, 13th August, 2018

Thomas Hackl | Günther Raidl

# Erklärung zur Verfassung der Arbeit

Thomas Hackl, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 13. August 2018

Thomas Hackl

# Danksagung

Ich möchte mich bei der EBG MedAustron GmbH[1], Marie Curie-Straße 5, 2700 Wiener Neustadt, Österreich, für die Zusammenarbeit und die finanzielle Unterstützung dieser Arbeit bedanken.

[1] https://www.medaustron.at

# Acknowledgements

I want to thank EBG MedAustron GmbH[2], Marie Curie-Straße 5, 2700 Wiener Neustadt, Österreich, for their cooperation and for partially funding this thesis.

[2] https://www.medaustron.at

# Kurzfassung

Das Partikeltherapienpatientenplanungsproblem (PTPSP) entsteht in modernen Krebstherapieeinrichtungen, die eine Partikeltherapie anbieten, und besteht aus der Planung von Therapien innerhalb eines Planungshorizonts von mehreren Monaten. Eine Besonderheit des PTPSP im Vergleich zur klassischen Strahlentherapieplanung besteht darin, dass die Therapien nicht nur auf Tagesebene, sondern auch innerhalb der Tage geplant werden müssen, da sich alle Therapien denselben Partikelstrahl teilen. In einer vorhergehenden Arbeit führten Maschler et al. diese neuartige Problemstellung ein und präsentierten erste Algorithmen, inklusive einer Iterated-Greedy-Metaheuristik (IG). In dieser Arbeit bauen wir auf dem IG auf und tauschen zwei Hauptkomponenten aus: die Konstruktionsphase und den lokalen Suchalgorithmus. Die resultierende Metaheuristik verbessert den bestehenden Ansatz und liefert für alle betrachteten Benchmark-Instanzen wesentlich bessere Ergebnisse. Außerdem präsentieren wir einen 2-Phasen-Ansatz, der mittels einer Variable-Neighbourhood-Descent-Methode (VND) die Tages- und Zeitzuordnungen nacheinander optimiert. Schlussendlich, verbessern wir unsere IG-Metaheuristik, indem wir die lokale Suche durch eine VND ersetzen. Diese Methode liefert für alle Benchmark-Instanzen noch bessere Ergebnisse.

Da die in der Praxis vorkommenden Probleminstanzen sehr groß sein können, ist eine möglichst effiziente Dursuchung der Nachbarschaften bei der lokalen Suche notwendig. Um den Aufwand der Suche zu reduzieren, definieren wir verschiedene Filter, die die Nachbarschaften auf die vielversprechendsten Lösungen einschränken, wodurch kostspielige Evaluierungen von wahrscheinlich schlechteren Lösungen vermieden werden. Die eigentliche Evaluierung wird inkrementell durchgeführt, indem nur jene Terme der Zielfunktion neu ausgewertet werden, deren Werte sich geändert haben. Eine Schwierigkeit bei diesem Ansatz besteht darin, dass alle Therapieeinheiten einer Therapie ungefähr zur selben Uhrzeit stattfinden müssen. Zu diesem Zweck hängt die Zielfunktion von Variablen ab, die für jede Therapie und Woche die sogenannte nominelle Startzeit repräsentieren. Die Berechnung dieser Variablen ist jedoch recht aufwendig. Daher führen die VNDs zuerst eine lokale Suche mit fixierten nominellen Startzeiten durch und berechnen im Anschluss die nominellen Startzeiten mittels linearer Programmierung.

Wir haben die einzelnen Nachbarschaften auf 40 verschiedenen Benchmark-Instanzen ausgewertet und mittels statistischer Methoden verglichen. Basierend auf den Ergebnissen zeigen wir, welche Nachbarschaften für die Verwendung in unseren Metaheuristiken

geeignet sind. Danach haben wir mit dem automatisierten Parameterkonfigurationsprogramm irace Nachbarschaftskombinationen und alle anderen Parameterwerte für unsere Metaheuristiken ausgewählt. Schließlich haben wir die Metaheuristiken auf den Benchmark-Instanzen ausgewertet und die Ergebnisse mit statistischen Tests verglichen.

Teile dieser Arbeit wurden bereits veröffentlicht.

# Abstract

The Particle Therapy Patient Scheduling Problem (PTPSP) arises in modern cancer treatment facilities that provide particle therapy and consists of scheduling a set of therapies within a planning horizon of several months. A particularity of PTPSP compared to classical radiotherapy scheduling is that therapies need not only be assigned to days but also scheduled within each day because all therapies share the same particle beam. In an earlier work Maschler et al. introduced this novel problem setting and provided first algorithms including an Iterated Greedy (IG) metaheuristic. In this work we build upon this IG and exchange two main components: the construction phase and the local search algorithm. The resulting metaheuristic enhances the existing approach and yields substantially better results for all of the considered benchmark instances. Moreover, we present a 2-Phase Approach (2PA) that uses a Variable Neighborhood Descent (VND) to first optimize the day assignments and then the time assignments. Finally, we improve our IG metaheuristic by replacing the local search algorithm with a VND. This method provides even better results on all benchmark instances.

Since the problem instances occurring in practice can be very large, an efficient exploration of the local search neighbourhoods is necessary. In order to reduce the computational effort, we define various filters that limit the neighbourhoods to the most promising solutions, thus, preventing expensive evaluations of solutions which are most likely worse. The actual evaluation is done incrementally by re-computing only those terms of the objective function whose values have changed. A difficulty with this approach is that all daily treatments of a therapy have to start approximately at the same time. To that end, the objective function depends on variables representing the so-called nominal starting time of each therapy and week. The computation of these variables, however, is quite costly. Therefore, the VNDs first perform a local search with fixed nominal starting times, and compute the nominal starting times afterwards using linear programming.

We evaluated the individual neighbourhoods on 40 different benchmark instances and compared them using statistical methods. Based on the results, we show which neighbourhoods are suitable for being used in our metaheuristics. We then used the automated parameter configuration tool irace to select neighbourhood combinations and all other parameter values for our metaheuristics. Finally, we evaluated the metaheuristics on the benchmark instances and compared the results with statistical tests.

Parts of this thesis were already published.

# Contents

CHAPTER 1

# Introduction

The Particle Therapy Patient Scheduling Problem (PTPSP) arises in radiotherapy used for cancer treatment. In classical radiotherapy cancer treatments are provided by linear particle accelerators that serve a dedicated treatment room exclusively. In contrast, particle therapy uses beams that are produced by either cyclotrons or synchrotrons that can serve up to five treatment rooms in an interleaved way. Several sequential activities like the stabilization of patients not requiring the beam have to be performed in the treatment room before and after each actual irradiation. Using several rooms and switching the beam between the rooms thus allows an effective utilization of the expensive particle accelerator and an increased throughput of the facility. We consider the situation at MedAustron,[1] a facility with three treatment rooms.

The goal of the PTPSP is to schedule several hundred patient therapies over the next few months. Each therapy consists of up to 35 daily treatments (DTs) that have to be assigned to different days, respecting a set of constraints: A therapy has to start in a given time window, a lower and upper bound of days are allowed to pass between two subsequent DTs, and a break of at least two consecutive days is required in each week. Additionally, DTs should start roughly at the same time within each week. Each DT requires different resources such as the particle beam, a room or an oncologist, where each resource can only be used by a single DT at any point in time. A resource is available only in predefined availability periods. A part of these time spans is considered extended service time, in which the usage of the resource leads to extra costs.

A schedule assigns all DTs of a given set of therapies to days and determines their starting times considering all operational constraints. A schedule's quality is determined by an objective function that is defined as a weighted sum of the therapies' finishing days, the amount of used extended availability time and the variation of the starting times of the DTs.

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

It turns out that instances of the PTPSP occurring in practice are so large that the time it takes to find a provably optimal schedule is not acceptable in general. Therefore, a construction heuristic and two metaheuristics have been proposed for this problem in a previous work [MRSR16]. The aim of this thesis is to study local search techniques and to apply them within three metaheuristic frameworks: a 2-Phase Approach (2PA) and two Iterated Greedy (IG) methods. Moreover, two arising subproblems are identified for further improving obtained solutions that can be modeled and solved efficiently with a linear programming approach.

The 2PA generates, on average, 25% better solutions than the reference metaheuristic from [MRSR16]. The first IG approach from this thesis was also published in [MHRR17]. Its solutions are, on average, again 25% better than the solutions of 2PA. With the second IG from this thesis we get a further improvement of 25% in solution quality compared to the first IG. It improves its initial solutions, on average, by 74%.

The thesis is organized as follows. In Chapter 2 we review the related literature including the metaheuristics from [MRSR16], which are used in the following chapters. Chapter 3 discusses the methods used in this work. In Chapter 4 a formal model for the PTPSP is presented. The main part of the thesis is split into Chapter 5, 7 and 6, where we describe the neighbourhood structures and two metaheuristics for solving the PTPSP. Chapter 8 introduces a Linear Programming (LP) model which is used to polish a schedule. In Chapter 9 we discuss the computational experiments conducted on a set of test instances. We conclude the thesis in Chapter 10 with a summary and an outlook on possible future research directions.

CHAPTER 2

# Related Work

A first attempt at automating the task of Radio Therapy Patient Scheduling (RTPS) has been made in 1993 by Larsson [Lar93].

In 2006 Kapamara et al. [KSH+06] formulate this task as a Job Shop Problem (JSP), where a number of patients having different priorities are to be assigned to a set of machines such that an objective function is minimized while respecting certain constraints. According to [KSH+06] JSPs are categorized along two dimensions: Firstly, a JSP can be static or dynamic. Static problems have the number of jobs and their ready times known in advance and fixed, whereas dynamic problems involve patients coming late to their appointments, machine breakdowns and other unforeseen occurrences which influence the schedule. Secondly, one can distinguish between a deterministic and stochastic variant of the JSP. In deterministic problems all parameter values of a job, like its processing times and due dates, are known beforehand, whereas in stochastic problems these values may vary. The authors stated that the RTPS is best described as a stochastic dynamic JSP due to the uncertainties and disturbances involved in the treatment process. In their study the authors identified several possible objective functions which can be minimized on their own or combined to a multi-objective function:

- the mean flow time of all jobs to the first definitive treatment,
- the mean flow time of all jobs,
- the difference between the above two objective functions and
- the number of jobs failing to meet the first due date.

Due to the fact that JSP is NP-hard, specialized exact approaches as well as heuristic methods are developed. The authors compared several exact methods, like Branch & Bound, as well as heuristic approaches, like Simulated Annealing, Tabu Search, Genetic

Algorithms (GAs) and Greedy Randomized Adaptive Search Procedures (GRASPs), and came to the conclusion that Tabu Search outperforms the others in the analysed experiments.

In [PLSS06] Petrovic et al. split the process of booking incoming patients into two phases: First the patients are prioritized according to the severity of the disease like in [KSH+06]. Afterwards the required number of treatment sessions are booked for each patient, starting from the patients with highest priority. Two greedy-like algorithms are presented for this task: one which books a treatment forward starting from the earliest feasible start date and another one that schedules a treatment backwards from the latest feasible start date. The objective function incorporates the number of patients, the total length of waiting time breaches of the patients and the number of interruptions. The researchers made experiments showing that the forward booking strategy is superior with palliative patients, while the backward booking method performs better with radical patients.

In [PLR08] Petrovic et al. evaluate four different variants of GRASP for RTPS. In a nutshell, this metaheuristc repeatedly constructs solutions using a randomized heuristic and locally improves each obtained solution [GP10]. All four developed GRASP methods have in common that the patients are first sorted by their due date, priority and the required number of sessions. Afterwards, one of the following four approaches is applied to schedule the patients from the ordered list. The *Target Approach* is similar to the algorithm in [PLSS06] as it tries to schedule a treatment session at a specified target day and moves it forward or backwards until all constraints are satisfied. In the *Utilisation Threshold Approach* a threshold is defined for each radiation machine and patient priority, s. t. no more patients of a particular priority can be scheduled on a certain machine if the machine's utilisation reaches the specified threshold. Experiments show that the best schedules are produced if the threshold for routine patients is 90%, thus reserving 10% of the time on the machine for urgent and emergency patients. The *Schedule Creation Day Approach* limits the set of days a treatment can start on by defining the weekdays that the first treatment session can be scheduled on for each patient priority. The best results are observed if urgent and routine patients can be scheduled only on 3 days in a week, while allowing emergent patients to be treated on any weekday. In the *Maximum Number of Days in Advance Approach* a schedule is created for a patient a specified maximum number of days before the patient's due date. If this number is smaller for routine patients than for emergent patients, then the latter ones have a better chance to be scheduled earlier.

In contrast to the above methods which construct schedules from scratch, the steepest hill climbing approach presented in [KP09] gets a complete, feasible schedule and optimises it in an iterative way until a stopping criterion is fulfilled. In each iteration neighbours of the current schedule are constructed by moving appointments to different days. A schedule is accepted if it is feasible and the best schedule found in the current iteration. A schedule's objective value to be minimized is computed as a weighted sum of the patients' lateness. The lateness of a patient is defined as the difference between the date the

patient's details are referred to the centre and the targeted start of his or her treatment. The weights depend on the patient's priority and are set to 10, 5 and 1 for emergency, palliative and radical patients, respectively. Applying the steepest hill climbing method to a (generated) data set of more than 2000 patients showed that the waiting time can be reduced considerably by combining a constructive heuristic and the presented steepest hill climbing method.

In [PMP09] and [PMP11] Petrovic et al. present a GA for optimizing radiotherapy schedules. A GA is a population-based metaheuristic that is inspired by natural selection and genetics [Mit98]. The GA selects good solutions in each iteration and applies one of two operators on them: The crossover operator combines two solutions by replacing a part of the first solution with a part of the other solution. The mutation operator modifies some solutions randomly. The authors encoded the schedules as strings of patient IDs, which define the order in which the patients are to be irradiated. Two different objectives are defined: minimisation of average waiting time and minimisation of average tardiness of the patients. The authors applied their algorithm to real life data and measured a reduction of the average waiting time and the tardiness by 35% and 20%, respectively.

Burke et al. [BLRP11] formulate the radiotherapy scheduling problem as a Mixed Integer Linear Programming (MILP) model. A MILP model is a mathematical model consisting of real or integer variables, linear equations and inequations constraining the variables' domains and a linear objective function which is to be minimized [CCZ14]. Although it seems too restrictive to model a problem as a set of linear relations, this approach has the advantage that an optimal solution can be found. The variables of the proposed model are integer variables defining whether or not a certain patient is scheduled on a particular machine. The constraints in the model either represent parameters originating directly from the problem instance like the number of sessions required for a particular patient, or define the relationship between two variables like the equation stating that two subsequent treatment sessions lie a certain number of days apart from each other.

The above approaches do not take into account the arrival distribution of the patients or future events. Legrain et al. [LFLR15] address this issue by developing an approach that combines stochastic optimization and online optimization.

The aforementioned contributions deal with constructing a schedule on a very coarse level, meaning that each treatment session is only assigned to a day but not a time of day. This simplification is reasonable only as long as every treatment room is served by an individual linear accelerator because in this case the treatments in different rooms are independent of each other and can be scheduled separately. However, a particle therapy centre usually contains several treatment rooms that are served by the same accelerator (a cyclotron or a synchrotron). In this scenario the start time of each treatment session must be carefully chosen, such that the radiation of a patient in one room ends just before the radiation of another patient in a different room is about to start and the beam can be switched to this room without a relevant idle period. Maschler et al. [MRSR16] propose a MILP model which can be used to model this problem. In theory, it is possible to find a provably optimal solution for this model, but it turned out [MRSR16] that

instances of the PTPSP occurring in practice are so large that the time it takes to do this is not acceptable in general. Therefore the authors developed several heuristic methods (TWCH, GRASP and IG) which will be explained until the end of this section.

The therapy-wise construction heuristic (TWCH) is a fast greedy heuristic to create a schedule from scratch. It operates in two phases: In the day assignment phase the heuristic selects one yet unconsidered therapy and assigns days to its DTs, i. e., treatment sessions, 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. See [MRSR16] for a detailed explanation and a pseudo code. The performance of the heuristic depends mostly on the order, in which the therapies are selected. Different strategies were evaluated:

1. Therapies with more DTs have a higher priority.
2. Therapies with an earlier latest starting day for the first DT have a higher priority.
3. Therapies with a higher resource consumption for the first DT have a higher priority.

Experiments showed that strategy 2 yields the best results. In the time assignment phase the working days are planned separately in a similar greedy-like fashion. To this end a not yet scheduled DT is selected which has the highest priority accoring to one of several priority functions. Then this DT is assigned to the earliest possible starting time after all already assigned DTs, respecting the availabilities of all required resources. The performance of this procedure depends to a high degree on the used priority function for selecting the next DT. The following criteria were evaluated:

1. A DT with minimum induced idle time for the beam resource is considered next.
2. A DT is preferred which requires the resource that leaves its regular service window first.
3. The ratio between the time the beam resource is required and the total processing time of a DT is considered. DTs with a smaller ratio are prioritized.

It was shown that criterion 1 is superior on average. However, it frequently happened that several DTs evaluate to the same priority value. In order to break such ties, a lexicographic combination of all three criteria is used in the final algorithm: First criterion 1 is applied. In case of a tie, criterion 2 is used, and if a tie happens again, the last criterion 3 is considered.

The second heuristic developed by Maschler et al. [MRSR16] is an implementation of GRASP. The first iteration schedules all therapies using TWCH. All subsequent iterations

construct new schedules using a randomized version of TWCH's day assignment, which selects suboptimal DTs too with a definable probability, and TWCH's time assignment. Each iteration ends with a local improvement which repeatedly assigns new times to the DTs by applying a randomized variant of TWCH's time assignment, where also suboptimal DTs are considered with a certain probability.

The third heuristic is called IG. In a nutshell, the general Iterated Greedy heuristic improves a solution by iteratively destroying and recreating parts of the solution [RS07]. The IG algorithm presented in [MRSR16] works as follows. TWCH is used to create an initial solution. The destruction operator removes a defineable amount of therapies from the schedule. The construction step is then performed by reapplying TWCH's day assignment for the set of removed therapies. Finally, TWCH's time assignment is applied from scratch to all working days which have been modified. Additionally, the randomized time assignment procedure from GRASP is used to further improve the obtained solution.

The three heuristics (TWCH, GRASP and IG) were tested on instance sets defining up to 300 therapies. A statistical comparison of the results showed that IG finds the best schedule on most instances.

Parts of this thesis were published in a recent publication by Maschler et al. [MHRR17].

CHAPTER 3

# Methods

A combinatorial optimization problem is the problem of finding a solution minimizing or maximizing a given objective function in a finite solution space. The PTPSP, which is formally defined in Chapter 4, belongs to this problem class. The solution space of a combinatorial optimization problem is given by a finite set $S$ containing all feasible solutions. The cost or quality of a solution in $S$ is defined by an objective function $f : S \to \mathbb{R}$. In the context of this thesis, a solution $x_1 \in S$ is considered better than a solution $x_2 \in S$ if $f(x_1) < f(x_2)$. Using this terminology, solving a combinatorial optimization problem means minimizing the objective function [AKM07].

One can distinguish between exact and heuristic approaches for solving combinatorial optimization problems. An exact approach finds a solution which is provably a global minimum w.r.t. to the objective function if there is one. However, on real-world problems exact approaches are often too slow, which limits their applicability. A LP model, for example, can be solved very efficiently in polynomial time, but many combinatorial problems are NP-hard meaning that they can in general not be solved exactly by a deterministic polynomial time procedure. Heuristic methods, on the other hand, do not guarantee to find the optimal solution. However, their strength is that they can find sufficiently good solutions for many real-world problems which are too complex to be exactly solved in a reasonable time.

Heuristic approaches can be classified, among others, into constructive and local search algorithms [AKM07]. Constructive algorithms generate a solution by iteratively extending a partial solution until a complete solution is obtained. Local search algorithms, on the other hand, start with a complete solution and try to find better solutions by making modifications to the current solution.

The next three sections in this chapter present three widely used heuristic approaches: local search, variable neighbourhood descent and iterated greedy. The forth section discusses linear programming, which is an exact approach. The last two sections are

dedicated to the statistical evaluation of optimization algorithms and to the process of finding good parameter values for a parameterized optimization method.

## 3.1 Local Search

A local search algorithm starts with a complete solution created by a construction method and tries to find better solutions by making modifications to the current solution. Algorithm 3.1 depicts the high-level structure of a local search algorithm.

**Algorithm 3.1:** Local Search

```
1 x ← initial solution;
2 repeat
3 |   choose an x' ∈ N(x);
4 |   if f(x') ≤ f(x) then
5 |   |   x ← x';
6 until stopping criteria satisfied;
7 return x;
```

Variable $x \in S$ holds a solution from the solution space $S$. How the solutions are represented, is an important design decision and cannot be defined in general because it highly depends on the concrete problem to be solved.

Function $N : S \to 2^S$ defines the *neighbourhood structure* that assigns a set of neighbours $N(x) \subseteq S$ to each solution $x \in S$. The set $N(x)$ is called the neighbourhood of $x$. Usually one specifies the neighbourhood structure not as a function but as a set of *move* operators which construct new solutions by modifying certain parts of the current solution, yielding a solution which has in general a slightly different objective value. The performance of a local search depends to a large extent on the concrete definition of the move operators. Ideally, the move operations construct only better solutions, s.t. no time is wasted by evaluating worse solutions. Note that the globally best solution is usually not reachable from any start solution. Hence, a local search, in general, finds only a local optimum w.r.t. the neighbourhood structure. Some techniques for escaping local optima are discussed in the remaining chapter.

There are different ways, called step functions, to choose $x' \in N(x)$:

**Random neighbour:** A random neighbour is selected.

**Next improvement:** $N(x)$ is searched in a specific order, taking the first solution that is better than $x$.

**Best improvement:** $N(x)$ is searched completely and the best neighbour is selected.

The step function has great influence on the performance but no strategy is always better than the other ones. At first glance, one might think that best improvement always

leads to the best objective value in the least number of iterations. However, it can be shown [HM06] that there are optimization problems where next improvement produces better solutions on some start solutions while best improvement performs better on other start solutions. The random neighbour strategy is the least targeted and is often used in more advanced algorithms to escape local optima. The local search method Simulated Annealing (SA), for example, is based on Algorithm 3.1 but accepts a randomly selected neighbour in Line 4 even if it is worse with a small probability. Another algorithm which makes use of this idea is General Variable Neighbourhood Search (GVNS) [GP10]. This method alternately finds a local optimum using next or best improvement, and selects a random neighbour (possibly in a different neighbourhood) to escape it.

The local search ends as soon as a stopping criterion or a combination of several criteria is fulfilled. The following stopping criteria are used in practice:

**Minimum reached:** If no better neighbour has been found using best or next Improvement, the search is aborted because the current solution must be locally optimal.

**Time limit:** The search is stopped if a given time limit is exceeded. For more complex heuristic search methods, one could define multiple time limits on different layers. If, for instance, the local search is embedded into another method, then two separate time limits could be defined for both methods. An example is the IG approach which is discussed in Chapter 7. This method executes a local search in every iteration before applying a destruction and construction operator. In order to ensure that the IG executes enough iterations, one could, for example, set the time limits of the local search and the whole IG to 10 seconds and 20 minutes, respectively.

**Solution quality:** The search is terminated if the current solution is good enough, e.g. its objective value is close enough to a known lower bound.

**Total number of iterations:** The procedure is stopped after a certain number of moves.

**Number of consecutive unsuccessful iterations:** The search is stopped after a certain number of consecutive moves that did not improve the solution. This criterion is especially useful for the random neighbour step function because the more consecutive unsuccessful iterations have passed the more likely it is to be already at a local optimum.

## 3.2 Variable Neighbourhood Descent

VND is a method which is based on the idea of systematically changing several neighbourhood structures $N_1, \ldots, N_{l_{\max}}$ during a local search [GP10]. Algorithm 3.2 illustrates this method.

**Algorithm 3.2:** Variable Neighbourhood Descent

```
1  x ← initial solution;
2  l ← 1;
3  repeat
4  |  find an x' with f(x') ≤ f(x''), ∀x'' ∈ N_l(x);
5  |  if f(x') ≤ f(x) then
6  |  |  x ← x';
7  |  |  l ← 1;
8  |  else
9  |  |  l ← l + 1;
10 until l > l_max;
11 return x;
```

The method starts with the first neighbourhood structure $N_1$. Line 4 finds the best solution $x'$ in the current neighbourhood $N_l(x)$ of $x$. If it is better than the current solution $x$, the first neighbourhood structure $N_1$ will be used in the next iteration again. If it is worse, then $x$ is already a local optimum with respect to the neighbourhood structure $N_l$. Hence, the algorithm switches to the next neighbourhood structure $N_{l+1}$. After termination the found solution is a local optimum with respect to all neighbourhood structures.

It can make sense to replace the best improvement step function by next improvement in line 4. This can speed up the convergence at the beginning of the search and lead to a better final solution [HM06] for some optimization problems. Additionally the same stopping criteria which are described for the local search can be applied here in order to abort the search before reaching an optimum.

## 3.3 Iterated Greedy

Iterated Greedy is a metaheuristic which is used to improve the performance of a given greedy construction heuristic [PP16]. As can be seen in Algorithm 3.3, IG consists of two phases which are executed repeatedly until a stopping criterion is fulfilled. The *destruction* phase removes random parts of the current solution yielding a partial solution. The *construction* phase completes the partial solution using the given greedy heuristic. Finally an *acceptance criterion* decides whether the new solution should become the next incumbent solution.

Possible choices for the acceptance criterion include the following [PP16]:

- Next Improvement: The constructed solution $x''$ is accepted if it is better than $x$.
- Random Walk: The constructed solution is always accepted unless it is infeasible.

**Algorithm 3.3:** Iterated Greedy

```
x ← initial solution;
repeat
    x' ← Destruction(x);
    x'' ← Construction(x');
    if acceptance criterion fulfilled then
        x ← x'';
until stopping criteria fulfilled;
```

- Simulated Annealing like: A solution $x''$ is always accepted if it is better than the current solution $x$. Otherwise it is accepted with probability $e^{-\frac{f(x'')-f(x)}{T}}$, where $T$ is a parameter called temperature.

## 3.4 Linear Programming

Many problems, which are solvable in polynomial time, can be modeled as LP problems in a natural way. They can then be solved using the Simplex Method developed by Dantzig [Dan16]. An explanation of this method would go beyond the scope of this thesis. Though it is worth mentioning that there are programming libraries, such as CPLEX[1] and GUROBI[2], which are highly optimized for solving such tasks.

Linear Programming, which is also known as Linear Optimization, deals with optimizing mathematical models composed solely of linear relations between the decision variables. To be more precise, an LP model consists of a cost vector $\mathbf{c} = (c_1, \ldots, c_n)$, a vector of unknowns $\mathbf{x} = (x_1, \ldots, x_n)$ and a linear cost function $\mathbf{c}^T\mathbf{x} = \sum_{i=1}^{n} c_i x_i$ that we seek to minimize over all vectors $\mathbf{x}$, subject to a set of linear equality and inequality constraints.

Let $M_1$, $M_2$ and $M_3$ be finite index sets for each of which we are given an n-dimensional vector $\mathbf{a_i}$ and a scalar $b_i$, used to form the $i$-th constriant. And let $N_1$ and $N_2$ be subsets of $\{1, \ldots, n\}$ indicating which variables $x_j$ are constraint to be nonnegative or nonpositive, respectively. Then the set of constraints is given as:

$$
\begin{aligned}
\mathbf{a_i}^T\mathbf{x} &\geq b_i && \forall i \in M_1,\\
\mathbf{a_i}^T\mathbf{x} &\leq b_i && \forall i \in M_2,\\
\mathbf{a_i}^T\mathbf{x} &= b_i && \forall i \in M_3,\\
x_j &\geq 0 && \forall j \in N_1,\\
x_j &\leq 0 && \forall j \in N_2.
\end{aligned}
$$

[1] https://www-01.ibm.com/software/commerce/optimization/cplex-optimizer
[2] http://www.gurobi.com

Even though LP is too restrictive for most problems, it is sometimes possible to find subproblems which can be modeled as an LP. Two important subproblems for the RTPS solvable using LP are presented in Chapter 4 and 8.

## 3.5 Statistical Evaluation

To check if a certain heuristic method produces better solutions than another method for the same optimization problem in a serious experimental way, one has to make use of a statistical hypothesis test. In this work we use the Wilcoxon rank-sum test [Wil45]. The null hypothesis of this test states that two independent samples were selected from populations having the same distribution. The alternative hypothesis says that the probability distributions of both populations are shifted against each other. The test assumes that the observations are independent of each other.

The comparison of two heuristic methods involves the following steps. First both heuristic methods have to be applied to a set of test instances a certain number of times, yielding two samples $X$ and $Y$ of objective values $x_1, \ldots, x_m$ and $y_1, \ldots, y_n$. Then all values from both samples are combined in a single list of $m+n$ values and sorted by size. Afterwards, each value is assigned a rank, which is just the value's position in the list if all values are different. Now the Wilcoxon rank sum statistic can be computed as

$$W = (\text{sum of all X ranks}) - (1 + 2 + \cdots + m).$$

This value $W$ is then compared to critical values $w_{\alpha/2}$ and $w_{1-\alpha/2}$, where $\alpha$ indicates the confidence level. The critical values for commonly used sample sizes are tabulated in [WKW70]. The null hypothesis is rejected if $W \leq w_{\alpha/2}$ or $W \geq w_{1-\alpha/2}$ in case a two-tailed test is used. For a one-tailed test, only one of the above conditions has to be used.

## 3.6 Parameter Configuration

After deciding which heuristic to use for solving a certain optimization task, one faces the problem of determining good parameter values for it. For example, the IG metaheuristic uses a parameter defining the destruction rate. A GA metaheuristic has several parameters like the population size, the crossover and the mutation rate. For a VND one has to decide which neighbourhoods to use in which order.

The task of finding a good parameter configuration for a given heuristic is called parameter tuning and follows the following steps:

1. The set of available problem instances is partitioned into a training set $T$ and a validation set $V$.
2. The heuristic's parameters are tuned on the set $T$, meaning that a parameter configuration $\theta$ from as set of available configurations $\Theta$ is found, with which the heuristic generates the best solutions on average.

3. The heuristic is applied on $V$ using $\theta$ to see whether $\theta$ is a good choice on a new data set too.

A brute-force approach for parameter tuning in Step 2 works by first executing the heuristic with each parameter configuration on every instance in $T$ and finding the best configuration by comparing their performance on all test runs afterwards. The main issue with this approach is that a lot of computational time is wasted with the worst parameter configurations even though it is obvious after few test instances that they cannot compete with other configurations.

A method which counteracts this problem is called Racing [MM97] [BSPV02]. The idea of Racing is to evaluate different parameter configurations in sequence and to discard a configuration as soon as enough statistical evidence is available that it is worse than the best configuration found so far. In case of F-Race this statistical proof is made using the Friedman test [BSPV02]. By excluding configurations early on, the other configurations can be evaluated more often given a certain time budget. Thus, more statistical evidence is gathered for the remaining configurations, which is needed in order to select the better of two configurations if both are very similar regarding their performance.

Balaprakash et al. [BBS07] proposed *iterated F-Race* which is an extension of F-Race that is suitable for very large configuration sets. It is an iterative procedure in which each iteration first defines a probability measure over the parameter space using the best configurations obtained from the previous iteration, then selects a subset of configurations that are distributed according to the newly defined probability measure, and finally applies F-Race on the selected configurations [BBS07]. The tool irace[3] implements this method and we used it to conduct the experiments in Section 9.

[3] `http://iridia.ulb.ac.be/irace/`

CHAPTER 4

# Problem Formalization

This chapter presents the formal model for the PTPSP, which is the basis for all following algorithms. Section 4.1 describes the given input data. For example, the set of all working days on which a therapy can be scheduled is given by the variable $D'$. Section 4.2 defines the solutions space by explaining how a solution is represented and what criteria a solution must fulfill in order to be feasible. The requirement that every DT of a therapy must be scheduled on a day in $D'$ is one of these criteria. Furthermore, several auxiliary variables are defined here, which help to formalize the problem. Finally, the objective of the PTPSP is described in an informal way in this section as well. The next section 4.3 defines the objective function to be minimized and lists all constraints for the variables it contains in a formal way. Both, the objective function and the constraints, use all three types of variables: variables defining a solution, auxiliary variables and input variables. The last section 4.4 is dedicated to an LP which is used to compute the optimal values of a particular subset of the auxiliary variables.

## 4.1 Given Input Data

We are given the following input data.

- Times are generally specified in $H^{\text{unit}}$ units of an hour.
- Let $D = \{0, \ldots, n_D - 1\}$ refer to the $n_D$ days that need to be considered within the planning period in the given order. Moreover, $D' \subseteq D$ denotes the subset of working days where the treatment centre is actually open. We refer to the weeks covered by $D$ by set $V = \{0, \ldots, n_V - 1\}$. Furthermore, let $\bigcup_{v \in V} D'_v$ be the partitioning of $D'$ into $n_V$ subsets corresponding to the $n_V$ weeks.

For day $d \in D'$, 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 anything must be scheduled, including extended times outside of the regular business hours.

- Let $R$, implemented by $R = \{0, \ldots, n_R - 1\}$, denote the set of all (renewable) resources. They include the following special ones:

  $r^{\text{B}}$: index of the beam resource
  $R^{\text{rooms}}$: set of indices of the room resources

  Each resource $r \in R$ is available on a subset of the working days $D_r^{\text{res}} \subseteq D'$. Moreover, each resource $r \in R$ is associated for each day $d \in D_r^{\text{res}}$ with a single 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. In addition, resources have defined extended service windows. For each $r \in R$ and $d \in D_r^{\text{res}}$ we are given $\widehat{W}_{r,d} = [\widehat{W}_{r,d}^{\text{start}}, \widehat{W}_{r,d}^{\text{end}}) \subseteq \widetilde{W}_d$, where $\widehat{W}_{r,d}^{\text{start}}$ and $\widehat{W}_{r,d}^{\text{end}}$ denote the extended start and end times, respectively, and $\widehat{W}_{r,d}^{\text{start}} \leq W_{r,d}^{\text{start}} \leq W_{r,d}^{\text{end}} \leq \widehat{W}_{r,d}^{\text{end}}$ holds. For some resources the extended service window might be the same as the regular one on all days. Therefore, we define the subset $\widehat{R} \subseteq R$ of resources with actual extended service windows, i.e., $\widehat{R} = \{r \in R \mid \exists d \in D_r^{\text{res}}(\widehat{W}_{r,d}^{\text{start}} < W_{r,d}^{\text{start}} \vee W_{r,d}^{\text{end}} < \widehat{W}_{r,d}^{\text{end}})\}$.

  Furthermore, for each resource $r \in R$ and each day $d \in D_r^{\text{res}}$, we are given unavailability time periods $\overline{W}_{r,d} = \bigcup_{w=0,\ldots,\omega_{r,d}-1} \overline{W}_{r,d,w}$ with $\overline{W}_{r,d,w} = [\overline{W}_{r,d,w}^{\text{start}}, \overline{W}_{r,d,w}^{\text{end}}) \subset \widehat{W}_{r,d},\ w = 0, \ldots, \omega_{r,d} - 1$, where $\overline{W}_{r,d,w}^{\text{start}}$ and $\overline{W}_{r,d,w}^{\text{end}}$ denote the start and end time of the $w$-th unavailability period. All these periods are non-overlapping, and sorted according to increasing time.

  We, thus, assume the service times of all resources to be cropped according to the general opening times $\widetilde{W}_d$. On the contrary, these general opening times are also tightened based on the resource availabilities as far as possible, considering only those time intervals in which any task might have a chance to be scheduled.

  Unavailability periods are expected to neither start at the beginning of extended resource availability periods nor to end directly at the end of extended resources availability periods since otherwise the resources extended service window (and possibly also the regular one) can be tightened accordingly.

- The set of therapies to be scheduled is given by $T$, implemented by the index set $T = \{0, \ldots, n_T - 1\}$. Each therapy $t \in T$ is associated with an ordered set of DTs $U_t$ implemented by the index set $U_t = \{0, \ldots, \tau_t - 1\}$. Last but not least, each DT $u \in U_t$ is associated with a sequence of activities. As all activities of each DT are always to be performed without any breaks in-between, we can ignore the activities here in our optimization, except that certain resources are only needed at certain times.

  For each therapy $t \in T$ we are given

  – a priority $\varphi_t^{\text{priority}} \geq 0$, which is typically 1,

  - a minimal number $n_t^{\text{twmin}}$ and a maximal number $n_t^{\text{twmax}}$ of DTs per week,
  - a minimal number $\delta_t^{\min} \geq 1$ and a maximal number $\delta_t^{\max}$ of days between two consecutive DTs.

  The subset $\widetilde{T} \subseteq T$ shall denote therapies which are actually remaining parts of larger therapies whose first parts have already been fixed or completed. For those, we are additionally given

  - $\widetilde{S}_{t,-1}$ the nominal starting time of the DT within the last already fixed week for therapy $t \in \widetilde{T}$.

  For each DT $u \in U_t$ we are given

  - an earliest starting day $d_{t,u}^{\min} \in D$ and a latest starting day $d_{t,u}^{\max} \in D$,
  - $p_{t,u} > 0$ denotes the processing time for performing the DT,
  - $Q_{t,u} \subseteq R$ denotes the set of resources required by the DT at some time,
  - for each required resource $r \in Q_{t,u}$, interval $P_{t,u,r} = [P_{t,u,r}^{\text{start}}, P_{t,u,r}^{\text{end}}) \cap \mathbb{Z} \subseteq [0, p_{t,u}) \cap \mathbb{Z}$ denotes the time relative to the DT's start in which resource $r$ is needed.

- Let $\delta^{\text{intraw}}$ denote a maximum intended time difference of the starting times of the first activities of the DTs within the same week.

- Let $\delta^{\text{interw}}$ denote a maximum intended time difference of the starting times of DTs between two consecutive weeks.

## 4.2 Solutions, Feasibility, and Objective

A *schedule* (solution) is described by a tuple $(Z, S)$, with

- $Z = \{Z_{t,u} \in D' \mid t \in T,\ u \in U_t\}$ denoting the days on which all the DTs are scheduled and

- $S = \{S_{t,u} \geq 0 \mid t \in T,\ u \in U_t\}$ denoting the starting times of the DTs at the respective days.

To aid modeling we use the following further variables:

- $Y_{t,v} \in \{0, 1\}$ for $t \in T,\ v \in V$ indicates with value one that therapy $t$ takes place, i.e., has at least one DT, in week $v$.

- $X_{t,d} \in \{0, 1\}$ for $t \in T,\ d \in D'$ indicates with value one that therapy $t$ has a DT at day $d$.

- $\widetilde{S}_{t,v}$ for each therapy $t \in T$ and each week $v \in V$ corresponds to the *nominal starting time* of the DT within the whole week $v$ when the treatment takes place in this week. The actual starting times within the week should not differ by more than a given tolerance $\delta^{\text{interw}}$ (soft constraint).

- $S^{\text{first}}_{r,d}$ and $S^{\text{last}}_{r,d}$ for $r \in \widehat{R},\ d \in D^{\text{res}}_r$ denote the first, respectively last, time resource $r$ is needed at day $d$.

- $\sigma^{\text{intraw}}_{t,u}$ corresponds for DTs $u \in U_t$ of therapies $t \in \widetilde{T}$ and DTs $u \in U_t \setminus \{0\}$ of therapies $t \in T \setminus \widetilde{T}$ to the violation of the maximum intended time difference of the starting times.

- $\sigma^{\text{interw}}_{t,v}$ corresponds for weeks $v \in V$ of therapies $t \in \widetilde{T}$ and $v \in V \setminus \{0\}$ of therapies $t \in T \setminus \widetilde{T}$ to the violation of the maximum intended time difference of the starting times of DTs between the two weeks $v-1$ and $v$.

To be *feasible*, a schedule must fulfill the following requirements.

- For each therapy, all its DTs must be scheduled sequentially at different days in the given order.

- For each therapy and for each week $D'_v$ of treatment except the last, the number of treatments has to be larger than or equal to $\min(n^{\text{twmin}}_t, |D'_v|)$.

- For each therapy, the number of treatments per week is not allowed to exceed $n^{\text{twmax}}_t$.

- Consecutive DTs of the same therapy have to be separated by at least $\delta^{\min}_t$ and at most $\delta^{\max}_t$ days. The times at these days are not considered hereby.

- For each DT $u \in U_t$, its resource requirements specified by $Q_{t,u}$ and $P_{t,u,r}$ must be fulfilled at the time the DT is scheduled.

The *objective* is to

- find a feasible schedule

- which minimizes usage of extended time outside of regular service windows of each resource in $\widehat{R}$ at each day,

- minimizes the finishing day $Z_{t,\tau_t}$ of each therapy $t \in T$, weighted by its priority $\varphi_t$, and

- minimizes the violation of the "intraweek" and "interweek" soft constraints.

The individual optimization objectives are roughly prioritized according to the order in which they are listed.

## 4.3 Mathematical Model

We now formulate a mathematical model that covers all aspects of the PTPSP using the variables introduced above. Auxiliary function $\text{used}(r, d, b)$ as well as constants $Z^{\text{earliest}}_{t,\tau_t}$, $S^{\text{L}}_{t,u}$, $S^{\text{U}}_{t,u}$, $\widetilde{S}^{\text{L}}_{t,v}$, and $\widetilde{S}^{\text{U}}_{t,v}$ are described below.

$$
\begin{aligned}
\min\ & \gamma^{\text{extfront}} \frac{1}{H^{\text{unit}}} \sum_{r \in \widehat{R}} \sum_{d \in D^{\text{res}}_r} \max\left(W^{\text{start}}_{r,d} - S^{\text{first}}_{r,d}, 0\right) + \\
& \gamma^{\text{extback}} \frac{1}{H^{\text{unit}}} \sum_{r \in \widehat{R}} \sum_{d \in D^{\text{res}}_r} \max\left(S^{\text{last}}_{r,d} - W^{\text{end}}_{r,d}, 0\right) + \\
& \gamma^{\text{finish}} \sum_{t \in T} \varphi^{\text{priority}}_t (Z_{t,\tau_t} - Z^{\text{earliest}}_{t,\tau_t}) + \\
& \gamma^{\text{intraw}} \frac{1}{H^{\text{unit}}} \sum_{t \in T} \sum_{u \in U_t \setminus \{0\}} \sigma^{\text{intraw}}_{t,u} + \\
& \gamma^{\text{interw}} \frac{1}{H^{\text{unit}}} \sum_{t \in T} \sum_{v \in V \setminus \{n_V - 1\}} \sigma^{\text{interw}}_{t,v} && (1) \\
\text{s.t. } & Z_{t,u} - Z_{t,u-1} \geq \delta^{\min}_t && \forall t \in T,\ \forall u \in U_t \setminus \{0\} \quad (2) \\
& Z_{t,u} - Z_{t,u-1} \leq \delta^{\max}_t && \forall t \in T,\ \forall u \in U_t \setminus \{0\} \quad (3) \\
& \sum_{d \in D'_v} X_{t,d} \geq \min(n^{\text{twmin}}_t, |D'_v|) \\
& \qquad \text{if } Y_{t,v} = Y_{t,v+1} = 1 && \forall t \in T,\ \forall v \in V \setminus \{n_V - 1\} \quad (4) \\
& \sum_{d \in D'_v} X_{t,d} \leq n^{\text{twmax}}_t \text{ if } Y_{t,v} = 1 && \forall t \in T,\ \forall v \in V \quad (5) \\
& X_{t,d} + X_{t,d'} \leq 1 && \forall t \in T,\ \forall v \in V, \\
& && \forall d, d' \in D' : d \in \max\{D'_v\}, \\
& && d' \in \min\{D'_{v+1}\},\ d' - d = 2 \quad (6) \\
& \text{used}(r, d, b) \leq 1_{b \notin \overline{W}_{r,d}} && \forall r \in R,\ \forall d \in D^{\text{res}}_r,\ \forall b \in \widehat{W}_{r,d} \quad (7) \\
& |S_{t,0} - \widetilde{S}_{t,0}| - \sigma^{\text{intraw}}_{t,0} \leq \delta^{\text{intraw}} \text{ if } Z_{t,0} \in D'_0 && \forall t \in \widetilde{T} \quad (8) \\
& |S_{t,u} - \widetilde{S}_{t,v}| - \sigma^{\text{intraw}}_{t,u} \leq \delta^{\text{intraw}} \text{ if } Z_{t,u} \in D'_v && \forall t \in T,\ \forall v \in V, \\
& && \forall u \in U_t \setminus \{0\} \quad (9) \\
& |\widetilde{S}_{t,0} - \widetilde{S}_{t,-1}| - \sigma^{\text{interw}}_{t,0} \leq \delta^{\text{interw}} \text{ if } Y_{t,0} = 1 && \forall t \in \widetilde{T} \quad (10) \\
& |\widetilde{S}_{t,v} - \widetilde{S}_{t,v-1}| - \sigma^{\text{interw}}_{t,v} \leq \delta^{\text{interw}} \text{ if } Y_{t,v} = Y_{t,v-1} = 1 && \forall t \in T,\ \forall v \in V \setminus \{0\} \quad (11) \\
& Z_{t,u} = d \rightarrow X_{t,d} = 1 && \forall t \in T,\ u \in U_t \quad (12) \\
& \sum_{d \in D'} X_{t,d} = \tau_t && \forall t \in T \quad (13) \\
& X_{t,d} \leq Y_{t,v} && \forall t \in T,\ \forall v \in V,\ \forall d \in D'_v \quad (14) \\
& S^{\text{first}}_{r,d} \leq S_{t,u} + P^{\text{start}}_{t,u,r} \text{ if } Z_{t,u} = d && \forall r \in \widehat{R},\ \forall d \in D^{\text{res}}_r,\ \forall t \in T, \\
& && \forall u \in U_t \mid r \in Q_{t,u} \quad (15)
\end{aligned}
$$

$$S^{\text{last}}_{r,d} \geq S_{t,u} + P^{\text{end}}_{t,u,r} \text{ if } Z_{t,u} = d \qquad \forall r \in \widehat{R},\ \forall d \in D^{\text{res}}_r,\ \forall t \in T,\ \forall u \in U_t \mid r \in Q_{t,u} \tag{16}$$

$$d^{\min}_{t,u} \leq Z_{t,u} \leq d^{\max}_{t,u} \qquad \forall t \in T,\ \forall u \in U_t \tag{17}$$

$$S^{\text{L}}_{t,u} \leq S_{t,u} \leq S^{\text{U}}_{t,u} \qquad \forall t \in T,\ \forall u \in U_t \tag{18}$$

$$\widetilde{S}^{\text{L}}_{t,v} \leq \widetilde{S}_{t,v} \leq \widetilde{S}^{\text{U}}_{t,v} \qquad \forall t \in T,\ \forall v \in V \tag{19}$$

$$\widehat{W}^{\text{start}}_{r,d} \leq S^{\text{first}}_{r,d} \leq S^{\text{last}}_{r,d} < \widehat{W}^{\text{end}}_{r,d} \qquad \forall r \in \widehat{R},\ \forall d \in D^{\text{res}}_r \tag{20}$$

$$\sigma^{\text{intraw}}_{t,0} \geq 0 \qquad \forall t \in \widetilde{T} \tag{21}$$

$$\sigma^{\text{intraw}}_{t,u} \geq 0 \qquad \forall t \in T,\ \forall u \in U_t \setminus \{0\} \tag{22}$$

$$\sigma^{\text{interw}}_{t,0} \geq 0 \qquad \forall t \in \widetilde{T} \tag{23}$$

$$\sigma^{\text{interw}}_{t,v} \geq 0 \qquad \forall t \in T,\ \forall v \in V \setminus \{0\} \tag{24}$$

Function

$$\text{used}(r, d, b) \coloneqq |\{u \in U_t \mid t \in T, r \in Q_{t,u} \wedge Z_{t,u} = d \wedge b - S_{t,u} \in P_{t,u,r}\}| \tag{25}$$

provides the number of DTs that use resource $r \in R$ on day $d \in D^{\text{res}}_r$ at time $b \in \widehat{W}_{r,d}$ (must always be either 0 or 1 in a feasible solution).

- Objective function (1) minimizes the use of the extended time windows in the front and the back of a regular service window, the number of days the treatments are finished later than their earliest possible finishing days and the violation of the "intraweek" and "interweek" soft constraints. The part of the objective function that minimizes the number of days the treatments are finished later than their earliest possible finishing days involves constants $Z^{\text{earliest}}_{t,\tau_t}$, that represent the earliest possible day treatment $t$ can be finished and are defined as:

$$Z^{\text{earliest}}_{t,\tau_t} = \begin{cases} d^{\min}_{t,1} + \left(\left\lceil \frac{\tau_t}{n^{\text{twmax}}_t} \right\rceil - 1\right)(7 - n^{\text{twmax}}_t) + (\tau_t - 1) & \text{if } \delta^{\min}_t = 1 \\ d^{\min}_{t,1} + (\tau_t - 1)\delta^{\min}_t & \text{otherwise.} \end{cases} \tag{26}$$

  Thus, only the days a treatment is finished later than $Z^{\text{earliest}}_{t,\tau_t}$ are considered.

  Constants $\gamma^{\text{extfront}}$, $\gamma^{\text{extback}}$, $\gamma^{\text{finish}}$, $\gamma^{\text{intraw}}$, and $\gamma^{\text{interw}}$ are the weights for the components of the objective function. To model the practical scenario these constants need to satisfy the following relations:

$$\gamma^{\text{extfront}} \geq \gamma^{\text{extback}} > \gamma^{\text{finish}} \tag{27}$$

$$\gamma^{\text{intraw}} \geq \gamma^{\text{interw}} \tag{28}$$

  The following concrete values for the weight constants are assumed within the context of this thesis:

  - $\gamma^{\text{extfront}} = 1$
  - $\gamma^{\text{extback}} = 1$
  - $\gamma^{\text{finish}} = 0.01$
  - $\gamma^{\text{intraw}} = 0.1$
  - $\gamma^{\text{interw}} = 0.1$

- Inequalities (2) enforce that all DTs of a therapy $t$ are scheduled in the correct order and the minimal required time between two consecutive DTs is adhered.

- Inequalities (3) enforce that all consecutive DTs of each therapy $t$ are scheduled not more than the maximal allowed time $\delta_t^{\max}$ apart.

- Inequalities (4) guarantee that, except in the last week of a treatment, at least $n_t^{\text{twmin}}$ DTs are scheduled per week.

- Inequalities (5) guarantee that for all treatments $t$ at most $n_t^{\text{twmax}}$ DTs are scheduled per week.

- Inequalities (6) ensure that if on a Saturday and on the following Monday DTs can be scheduled, that for each treatment only one of both days is used. This assures that for each treatment there is a break of at least two consecutive days per week.

- Inequalities (7) enforce that the resource consumption never exceeds the resource availability at any time during the operating hours.

- Inequalities (8) and (9) represent the soft constraints that the starting times of the activities should not deviate too much from the week's nominal starting times; deviations are determined by variables $\sigma_{t,u}^{\text{intraw}}$ and penalized in the objective function. The first treatment is not considered here and therefore excluded for therapies that have not yet started.

- Inequalities (10) and (11) represent the soft constraints that then nominal times of two consecutive weeks should not differ too much; deviations are determined by variables $\sigma_{t,v}^{\text{interw}}$ and penalized in the objective function.

- Inequalities (12) and (13) link the $Z$ variables with the $X$ variables.

- Inequalities (14) link the $X$ variables with the $Y$ variables.

- Inequalities (15) force $S_{r,d}^{\text{first}}$ to be less than or equal to the starting time of all activities using resource $r$ on day $d$.

- Inequalities (16) force $S_{r,d}^{\text{last}}$ to be greater than or equal to the finishing time of all activities using resource $r$ on day $d$.

- The domains of the DTs' days are given by (17).

- The domains of the starting times $S_{t,u}$ of the DTs are given in (18), where the bounds $S^{\mathrm{U}}_{t,u}$ and $S^{\mathrm{U}}_{t,u}$ can be calculated as the minimum and maximum times for which all resources required by their activities are available:

$$S^{\mathrm{L}}_{t,u} = \min\{b \mid \forall r \in Q_{t,u},\ \exists d \in \{d^{\min}_{t,u}, \ldots, d^{\max}_{t,u}\}(\widehat{W}^{\mathrm{start}}_{r,d} - P^{\mathrm{start}}_{t,u,r} \leq b)\} \tag{29}$$

$$S^{\mathrm{U}}_{t,u} = \max\{b \mid \forall r \in Q_{t,u},\ \exists d \in \{d^{\min}_{t,u}, \ldots, d^{\max}_{t,u}\}(b \leq \widehat{W}^{\mathrm{end}}_{r,d} - P^{\mathrm{start}}_{t,u})\} \tag{30}$$

  Should the sets over which the minimum and maximum are determined be empty, then the problem instance has no feasible solution.

- The domains of the nominal starting times of the DTs $\widetilde{S}_{t,v}$ are given by (19), where $\widetilde{S}^{\mathrm{L}}_{t,v}$ and $\widetilde{S}^{\mathrm{U}}_{t,v}$ can be calculated as follows for $v \in V^t = \{v \mid \exists u \in U_t : \{d^{\min}_{t,u}, \ldots, d^{\max}_{t,u}\} \cap D'_v \neq \emptyset\} \subseteq V$ of weeks during which DTs of therapy $t$ may be provided:

$$\widetilde{S}^{\mathrm{L}}_{t,v} = \min\{S^{\mathrm{L}}_{t,u} \mid u \in U_t \wedge (t \in \widetilde{T} \vee u > 0) \wedge \{d^{\min}_{t,u}, \ldots, d^{\max}_{t,u}\} \cap D'_v \neq \emptyset\} \tag{31}$$

$$\widetilde{S}^{\mathrm{U}}_{t,v} = \max\{S^{\mathrm{U}}_{t,u} \mid u \in U_t \wedge (t \in \widetilde{T} \vee u > 0) \wedge \{d^{\min}_{t,u}, \ldots, d^{\max}_{t,u}\} \cap D'_v \neq \emptyset\} \tag{32}$$

  For $v \notin V^t$ bounds can be set to $\pm\infty$ since the corresponding variables have no influence on the model.

- (20) restrict the domains of the variables $S^{\mathrm{first}}_{r,d}$ and $S^{\mathrm{last}}_{r,d}$ to be in the extended service window (including the regular service window).

- Inequalities (21) and (22), and Inequalities (23) and (24) restrict the domain of the $\sigma^{\mathrm{intraw}}_{t,u}$ and $\sigma^{\mathrm{interw}}_{t,u}$ variables to be non-negative.

## 4.4 Computation of Nominal Starting Times

To compute the intraweek and the interweek part of the objective function, the PTPSP model from Section 4.3 uses the auxiliary variables $\widetilde{S}_{t,v}$ that correspond to the nominal starting times of DTs. The optimal values for these variables can be calculated in polynomial time by solving an LP model. This LP model contains all the elements of the PTPSP model which are involved in the computation of the intraweek and interweek soft constraints. It is defined for each therapy $t$ separately and assumes that $S_{t,u}$ and $Z_{t,u}$ are fixed. It uses the following auxiliary variables:

$$\begin{aligned}
&V_t = \{v \in V \mid Y_{t,v} = 1\} \\
&U_{t,v} = \{u \in U_t \mid Z_{t,u} \in D_v\} && \forall v \in V \\
&\mathcal{V}_t = \{(v, v') \in V_t \times V_t \mid v' = v + 1\} \\
&S^{\min}_t = \min\{S_{t,u} \mid u \in U_t\}
\end{aligned}$$

$$S_t^{\max} = \max\{S_{t,u} \mid u \in U_t\}$$

The model's decision variables are:

| | |
|---|---|
| $\widetilde{S}_{t,v}$ | $\forall v \in V_t$ |
| $\sigma_{t,u}^{\text{intraw}}$ | $\forall u \in U_t \setminus \{0\}$ |
| $\sigma_{t,0}^{\text{intraw}}$ | if $t \in \widetilde{T}$ |
| $\sigma_{t,v}^{\text{interw}}$ | $\forall v \in V_t \setminus \{0\}$ |
| $\sigma_{t,0}^{\text{interw}}$ | if $t \in \widetilde{T}$ |

The function to be minimized is composed of those parts of the objective function 1 which contain the above decision variables:

$$\min \gamma^{\text{intraw}} \frac{1}{H^{\text{unit}}} \sum_{u \in U_t \setminus \{0\}} \sigma_{t,u}^{\text{intraw}} + \gamma^{\text{interw}} \frac{1}{H^{\text{unit}}} \sum_{v \in V_t \setminus \{n_V - 1\}} \sigma_{t,v}^{\text{interw}}$$

The decision variables are constrained by the following inequalities, which are also part of the PTPSP model. Note that $\widetilde{S}_{t,-1}$ is considered as constant.

$$|S_{t,0} - \widetilde{S}_{t,0}| \leq \sigma_{t,0}^{\text{intraw}} + \delta^{\text{intraw}} \qquad \text{if } t \in \widetilde{T} \wedge Z_{t,0} \in D_0' \qquad (33)$$

$$|S_{t,u} - \widetilde{S}_{t,v}| \leq \sigma_{t,u}^{\text{intraw}} + \delta^{\text{intraw}} \qquad \forall v \in V_t, \; \forall u \in U_{t,v} \setminus \{0\} \qquad (34)$$

$$|\widetilde{S}_{t,0} - \widetilde{S}_{t,-1}| \leq \sigma_{t,0}^{\text{interw}} + \delta^{\text{interw}} \qquad \text{if } t \in \widetilde{T} \wedge Z_{t,0} \in D_0' \qquad (35)$$

$$|\widetilde{S}_{t,v'} - \widetilde{S}_{t,v}| \leq \sigma_{t,v'}^{\text{interw}} + \delta^{\text{interw}} \qquad \forall (v, v') \in \mathcal{V}_t \qquad (36)$$

$$\sigma_{t,0}^{\text{intraw}} \geq 0 \qquad \text{if } t \in \widetilde{T} \qquad (37)$$

$$\sigma_{t,u}^{\text{intraw}} \geq 0 \qquad \forall u \in U_t \setminus \{0\} \qquad (38)$$

$$\sigma_{t,0}^{\text{interw}} \geq 0 \qquad \text{if } t \in \widetilde{T} \qquad (39)$$

$$\sigma_{t,v}^{\text{interw}} \geq 0 \qquad \forall v \in V_t \setminus \{0\} \qquad (40)$$

$$S_t^{\min} \leq \widetilde{S}_{t,v} \leq S_t^{\max} \qquad \forall v \in V_t \qquad (41)$$

- Constraints (33) – (40) are part of the PTPSP model.
- Constraint (41) is derived from the PTPSP model constraint $\widetilde{S}_{t,v}^{\text{L}} \leq \widetilde{S}_{t,v} \leq \widetilde{S}_{t,v}^{\text{U}}$.

The absolute values used in constraints (33) – (36) can be linearized by splitting an inequality of the form $|x| \leq y$ into $x \leq y$ and $-x \leq y$. For instance, the constraints (33) are linearized to

$$\begin{aligned} S_{t,0} - \widetilde{S}_{t,0} &\leq \sigma_{t,0}^{\text{intraw}} + \delta^{\text{intraw}} && \text{if } t \in \widetilde{T} \wedge Z_{t,0} \in D_0' && \text{and} \\ \widetilde{S}_{t,0} - S_{t,0} &\leq \sigma_{t,0}^{\text{intraw}} + \delta^{\text{intraw}} && \text{if } t \in \widetilde{T} \wedge Z_{t,0} \in D_0'. \end{aligned}$$

The other constraints can be converted analogously.

CHAPTER 5

# Neighbourhoods

This chapter presents the neighbourhood structures that serve as building blocks for the two metaheuristics defined in Chapters 6 and 7. First the *intra-day* neighbourhood structures are defined, which change the starting times $S$ of DTs only but leave the day assignments unchanged. Then the *inter-day* neighbourhood structures are explained, which change $Z$ by moving DTs to different days.

For modelling the intra-day neighbourhood structures, we exploit the fact that all DTs on a day require the same beam resource B exactly once to define a unique sequence of the DTs scheduled on a particular day. Let $G_d = \{(t,u) \mid t \in T, u \in U_t, Z_{t,u} = d\}$ be the set of DTs assigned to day $d \in D'$. We then encode a solution into a sequence $((t,u)_i)_{i=1}^{|G_d|}$ of DTs which is sorted by the times from which on they use the beam B, i.e., according to $S_{t,u} + P^{\text{start}}_{t,u,\text{B}}$. To evaluate the objective function we have to decode a sequence of DTs to obtain an actual time assignment. Algorithm 5.1 shows this decoding for a given working day $d \in D'$, its set of DTs $G_d$ and a sequence $((t,u)_i)_{i=1}^{|G_d|}$. The procedure starts by initializing the time marker $C_r$ to the earliest time a resource $r$ becomes available. In the main loop each DT in the sequence is assigned to the earliest possible start time at which all resources are available. First, at Line 3 the start time $S_{t,u}$ is set to the earliest time at which no required resource is used before the 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 $C_r$ time markers are set to the times when the corresponding resources become free after the just scheduled DT.

Starting from a solution $(Z,S)$, a set $\widetilde{\mathcal{N}}(Z,S)$, containing the neighbours of $(Z,S)$ in an encoded form, will be defined. For the intra-day neighbourhood structures, for example, a neighbour is encoded as a permutation $\pi : \{1,\ldots,|G_d|\} \to \{1,\ldots,|G_d|\}$ on the index set of $((t,u)_i)_{i=1}^{|G_d|}$, which changes the order of the DTs on day $d$. The inter-day neighbourhoods employ different encodings for the neighbours, such as a tuple consisting

**Algorithm 5.1:** Time assignment of a given sequence of DTs.

**Input:** A day $d$ and a sequence $((t,u)_i)_{i=1}^{|G_d|}$ of DTs

$C_r \leftarrow W_{r,d}^{\text{start}} \qquad \forall r \in R,\ d \in D_r^{\text{res}}$;
**for** $(t,u) \leftarrow (t,u)_1, \ldots, (t,u)_{|G_d|}$ **do**
  $S_{t,u} \leftarrow \max_{r \in Q_{t,u}} (C_r - P_{t,u,r}^{\text{start}})$;
  **while** $\exists r \in Q_{t,u} \wedge \exists \overline{W}_{r,d,w} \in \overline{W}_{r,d} : [S_{t,u} + P_{t,u,r}^{\text{start}}, S_{t,u} + P_{t,u,r}^{\text{end}}) \cap \overline{W}_{r,d,w} \neq \emptyset$ **do**
    $S_{t,u} \coloneqq \overline{W}_{r,d,w}^{\text{end}} - P_{t,u,r}^{\text{start}}$;
  **end**
  $C_r \leftarrow S_{t,u} + P_{t,u,r}^{\text{end}} \qquad \forall r \in Q_{t,u}$;
**end**

of a therapy's index together with a number stating how many days the therapy is shifted. The final neighbourhood $\mathcal{N}(Z,S)$ is then derived from $\widetilde{\mathcal{N}}$ by mapping each encoded neighbour to a solution $(Z,S)$. How this is done depends on the encoding and is explained for each inter-day neighbourhood structure separately.

Preliminary tests have shown that some neighbourhood structures produce neighbourhoods which are too big to be explored in a reasonable time. In order to reduce the computational effort, filters will be introduced. A filter describes a subset $\mathcal{N}^{\text{filter}}(Z,S)$ of neighbours which will not be evaluated by the local search. A neighbourhood with applied filter is then defined as $\mathcal{N}^{\text{filtered}}(Z,S) = \mathcal{N}(Z,S) \setminus \mathcal{N}^{\text{filter}}(Z,S)$. Different filters can also be combined. Let $P$ and $Q$ be two filters. Then $\mathcal{N}(Z,S) \setminus (\mathcal{N}^{\text{P}}(Z,S) \cup \mathcal{N}^{\text{Q}}(Z,S))$ defines a set of neighbours which are accepted by both filters. In other words, only the most promising neighbours are kept for evaluation. Contrarily, $\mathcal{N}(Z,S) \setminus (\mathcal{N}^{\text{P}}(Z,S) \cap \mathcal{N}^{\text{Q}}(Z,S))$ defines a set of neighbours which are accepted by at least one of the filters.

## 5.1 Permuting Daily Treatments

The first intra-day neighbourhood structure, abbreviated as $\mathcal{N}_{c,d}^{S1}$, considers permutations of $c$ DTs within day $d$. For instance, $\mathcal{N}_{2,d}^{S1}(Z,S)$ contains all solutions resulting from exchanging two DTs on day $d$. We further define $\mathcal{N}_{2,D'}^{S1} = \bigcup_{d \in D'} \mathcal{N}_{2,d}^{S1}$ in order to simplify the definition of IG and the 2PA in the next chapters.

The set of encoded neighbours $\widetilde{\mathcal{N}}_{c,d}^{S1}(Z,S)$ is formally defined as the set of all permutations $\pi : \{1, \ldots, |G_d|\} \to \{1, \ldots, |G_d|\}$ of the DTs on day $d$ with $|\{(m,n) \in \pi \mid m \neq n\}| \leq c$. A final neighbour is constructed by decoding $((t,u)_{\pi(1)}, ..., (t,u)_{\pi(|G_d|)})$ using Algorithm 5.1.

The following filter operations are employed to reduce the search space:

1. The *adjacent* filter causes the DTs' starting times to change only slightly by ensuring that only adjacent DTs change their places. To formalize this filter, let

$\pi \in \widetilde{\mathcal{N}}^{S1}_{c,d}(Z,S)$ and $X = \{m \mid (m,n) \in \pi, m \neq n\}$. Then this filter accepts the neighbour if $(\max X - \min X) < c$. An application of this filter makes sense if the DTs are already in the right place regarding the resources' availability periods, but the way they are interleaved with each other could still be improved. As for the intraweek and interweek softconstraints, we assume that the DTs of a therapy already start at roughly the same time on a day. Therefore, if a DT is moved to a very different starting time, we can expect the intraweek and interweek softconstraints to be violated to a higher degree. Such changes are excluded by this filter. The neighbourhood's size is reduced considerably by this filter: For any day $d$ the neighbourhood $N^{S1}_{2,d}(Z,S)$ contains $\frac{|G_d|(|G_d|-1)}{2}$ solutions, which is quadratic in $|G_d|$. The filtered neighbourhood contains only $|G_d| - 1$ solutions.

2. The *gap* filter rejects all DT permutations $\pi$ that increase the number of times an irradiation room is used in a row. The rationale of this definition is that the beam is idle between two consecutive DTs taking place in the same irradiation room, which may lead to an increased usage of the extended time windows.

3. The *nominal starting time (NST)* filter considers the $\widetilde{S}_{t,v}$ values for all therapies $t$ scheduled on the considered day $d$, where $v$ is the week of $d$, i.e. $d \in D_v$. It computes the aggregated deviation of the DTs' starting times from the corresponding nominal starting times $\widetilde{S}_{t,v}$ before the move as $\Delta = \Sigma\{\max(0, \mathrm{abs}(\widetilde{S}_{t,v} - S_{t,u}) - \delta^{\mathrm{intraw}}) \mid t \in T, u \in U_t, (t,u) \in G_d\}$. Let $(t,u)_i$ be a DT from the sequence of DTs before the move. Let further $j = \pi(i)$ and $(t',u')$ be the $j^{\mathrm{th}}$ DT in this sequence. Then the starting time $S'_{t,u}$ that $(t,u)$ would have after the move is estimated as $S'_{t,u} = S_{t',u'}$. Based on this estimation, the aggregated deviation of the nominal starting times from the starting times the DTs would approximately have *after* the move is calculated as $\Delta' = \Sigma\{\max(0, \mathrm{abs}(\widetilde{S}_{t,v} - S'_{t,u}) - \delta^{\mathrm{intraw}}) \mid t \in T, u \in U_t, (t,u) \in G_d\}$. This filter then accepts the neighbour if $\Delta' \leq \Delta$.

## 5.2 Moving Daily Treatments

This intra-day neighbourhood structure, abbreviated as $\mathcal{N}^{S2}_d$, moves a DT to a different place within a day $d$ of the encoded solution. We further define $\mathcal{N}^{S2}_{D'} = \bigcup_{d \in D'} \mathcal{N}^{S2}_d$ in order to simplify the definition of the 2PA and IG in the next chapters.

This neighbourhood structure is defined similarly to the insertion neighbourhood from [PR14]. Let $((t,u)_i)^{|G_d|}_{i=1}$ be the encoded solution of DTs on day $d$ and $\pi = (1, \ldots, |G_d|)$ its index sequence. The $\mathcal{N}^{S2}_d$ neighbourhood of the encoded solution is the result of the consideration of all pairs of positions $j, k \in \{1, \ldots, |G_d|\}$ of $\pi$, $j \neq k$, where the DT in position $j$ is removed from the encoded solution and inserted in position $k$. The resulting index sequence after such a movement is

$$\pi_{(j,k)} = (1, \ldots, j-1, j+1, \ldots, k, j, k+1, \ldots, |G_d|)$$

if $j < k$, or

$$\pi_{(j,k)} = (1, \ldots, k-1, j, k, \ldots, j-1, j+1, \ldots, |G_d|)$$

if $j > k$. The set of moves $I$ is defined as

$$I = \{(j,k) \mid j \neq k, 1 \leq j,k \leq |G_d| \wedge j \neq k-1, 1 \leq j \leq |G_d|, 2 \leq k \leq |G_d|\}$$

and the set of all encoded neighbours $\widetilde{\mathcal{N}}_d^{S2}(Z,S)$ is defined as the set of all permutations $\pi_{(j,k)}$ with $(j,k) \in I$. A final neighbour is constructed by decoding the reordered sequence of DTs $((t,u)_{\pi(1)}, ..., (t,u)_{\pi(|G_d|)})$ on day $d$ using Algorithm 5.1.

This neighbourhood structure supports the *gap* and *NST* filter from $\mathcal{N}_d^{S1}$. The *adjacent* filter is not employed because instead of moving a DT to a neighbouring position, one can use $\mathcal{N}_d^{S1}$ to get the same DT sequence.

## 5.3 Moving Daily Treatments across Day Boundaries

The first inter-day neighbourhood structure, abbreviated as $\mathcal{N}^{Z1}$, moves a DT $u$ of a therapy $t$ to a later (or earlier) day and re-assigns all daily treatments of $t$ conflicting with $u$ to new days.

The set of encoded neighbours for a solution $(Z,S)$ is defined as $\widetilde{\mathcal{N}}^{Z1}(Z,S) = \{(t,u,r) \mid t \in T, u \in U_t, r \in \{0,1\}\}$, where $r$ denotes the direction into which the DT $(t,u)$ should be moved.

A final neighbour is constructed by Algorithm 5.2. Suppose, the algorithm is called with $(t_0,u_0,r)$ where $r = 1$ indicating that the DT $(t_0,u_0)$ should be moved to a later day. Between Line 2 and 6, all DTs after $(t_0,u_0)$, including $(t_0,u_0)$ itself, are removed from the schedule. Line 13 assigns DT $(t_0,u_0)$ to the first working day which has all required resources for it. In Line 15 all DTs which have been removed previously are assigned to days after the newly assigned DT $(t_0,u_0)$. Function check_constraint_$\text{n}^{\text{twmin}}$ returns false if Constraint 4 of the formal model is violated, which is the case if not enough DTs are assigned to a week.

Function schedule_dt_after$(Z,S,t,u,d)$ tries to assign the DT $(t,u)$ to a day $d' > d$, minimizing $(d'-d)$ while respecting the constraints 2, 3, 5 and 6 of the formal model as well as the resource requirements and availabilities. The function returns $d'$ or $-1$ if the DT could not be assign to a day, s.t. a feasible schedule results. The function schedule_dt_before$(Z,S,t,u,d)$ is defined analogously but tries to assign DT $(t,u)$ to a day $d' < d$, minimizing $(d-d')$ instead.

## 5.4 Moving Therapies

This neighbourhood structure, abbreviated as $\mathcal{N}^{Z2}$, moves whole therapies by moving every DT of a therapy by at least $n$ days, where $n$ takes on all sensible values.

The set of encoded neighbours for a solution $(Z,S)$ is defined as $\widetilde{\mathcal{N}}^{Z2}(Z,S) = \{(t,n) \in T \times \mathbb{Z} \setminus \{0\} \mid d_{t,0}^{\min} - Z_{t,0} \leq n \leq d_{t,0}^{\max} - Z_{t,0}\}$.

**Algorithm 5.2:** Decoding a neighbour from $\widetilde{\mathcal{N}}^{Z1}$

**Input:** An encoded neighbour $(t_0, u_0, r)$

1 $d \leftarrow Z_{t_0,u_0}$;
2 $U_{\text{part}} \leftarrow \{u \in U_{t_0} \mid (r = 0 \rightarrow u \leq u_0) \wedge (r = 1 \rightarrow u \geq u_0)\}$;
3 **for** $u \in U_{part}$ **do**
4 $\quad Z^{\text{old}}_{t_0,u} \leftarrow Z_{t_0,u}$;
5 $\quad Z_{t_0,u} \leftarrow -1$; `// remove day assignment in solution`
6 **end**
7 **if** $r = 0$ **then**
8 $\quad$ **if** *schedule_dt_before*$(Z, S, t_0, u, d) = -1$ **then return** false;
9 $\quad$ **for** $(u \leftarrow u_0 - 1; u \geq 0; u \leftarrow u - 1)$ **do**
10 $\quad\quad$ **if** *schedule_dt_before*$(Z, S, t_0, u, Z^{old}_{t_0,u} + 1) = -1$ **then return** false;
11 $\quad$ **end**
12 **else if** $r = 1$ **then**
13 $\quad$ **if** *schedule_dt_after*$(Z, S, t_0, u, d) = -1$ **then return** false;
14 $\quad$ **for** $(u \leftarrow u_0 + 1; u \leq \max U_{part}; u \leftarrow u + 1)$ **do**
15 $\quad\quad$ **if** *schedule_dt_after*$(Z, S, t_0, u, Z^{old}_{t_0,u} - 1) = -1$ **then return** false;
16 $\quad$ **end**
17 **end**
18 **if** *check_constraint_n*$^{twmin}(Z, S, t_0) = false$ **then return** false;
19 **return** true;

A final neighbour is constructed by Algorithm 5.3. Assume the algorithm is called with $(t_0, n)$ where $n > 0$, indicating that each DT of therapy $t_0$ should be provided at least $n$ days later. The loop in Line 1 removes all day assignments of therapy $t_0$ from the schedule. Line 10 schedules the first DT of the therapy, s.t. it is provided *exactly* $n$ days later. The loop in Line 17 schedules the remaining DTs, s.t. they are provided *at least* $n$ days later. If $n < 0$, the last DT of the therapy is scheduled to an earlier day in Line 10. Afterwards all remaining DTs are scheduled to earlier days in Line 12. The functions occurring in this algorithm are explained in Section 5.3.

## 5.5 Shifting Therapies

In a nutshell, this neighbourhood structure, abbreviated as $\mathcal{N}^{Z3}$, moves whole therapies $t$ by assigning every $i^{\text{th}}$ DT, $i \in U_t$, of $t$ to the $(i+n)^{\text{th}}$ DT's day of $t$ for a sensible range of values for $n$. The first or last $|n|$ DTs of the therapy are scheduled on the latest or earliest days, respectively, s.t. all constraints are satisfied.

Let $l = \max(d^{\min}_{t,0} - Z_{t,0}, -\tau_t)$ denote the lower bound for $n$ and $u = \min(d^{\max}_{t,0} - Z_{t,0}, \tau_t)$ denote the upper bound. Then the set of encoded neighbours for a solution $(Z, S)$ is defined as $\widetilde{\mathcal{N}}^{Z3}(Z, S) = \{(t, n) \in T \times \mathbb{Z} \setminus \{0\} \mid l \leq n \leq u\}$.

**Algorithm 5.3:** Decoding a neighbour from $\widetilde{\mathcal{N}}^{Z2}$

**Input:** An encoded neighbour $(t_0, n)$

1 **for** $u \in U_{t_0}$ **do**
2 | $Z^{\text{old}}_{t_0,u} \leftarrow Z_{t_0,u}$;
3 | $Z_{t_0,u} \leftarrow -1$; `// remove day assignment in solution`
4 **end**
5 **if** $n < 0$ **then**
6 | $u_0 \leftarrow \tau_t - 1$;
7 **else**
8 | $u_0 \leftarrow 0$;
9 **end**
10 **if** *schedule_dt*$(Z, S, t_0, u_0, Z^{old}_{t_0,u_0} + n) = false$ **then return** false;
11 **if** $n < 0$ **then**
12 | **for** $u \leftarrow u_0 - 1; u \geq 0; u \leftarrow u - 1$ **do**
13 | | $d^{\text{before}} \leftarrow Z^{\text{old}}_{t_0,u} + n + 1$;
14 | | $d \leftarrow \text{schedule\_dt\_before}(Z, S, t_0, u, d^{\text{before}})$;
15 | **end**
16 **else**
17 | **for** $u \leftarrow u_0 + 1; u < \tau_{t_0}; u \leftarrow u + 1$ **do**
18 | | $d^{\text{after}} \leftarrow Z^{\text{old}}_{t_0,u} + n - 1$;
19 | | $d \leftarrow \text{schedule\_dt\_after}(Z, S, t_0, u, d^{\text{after}})$;
20 | **end**
21 **end**
22 **if** *check_constraint_*$n^{twmin}(Z, S, t_0) = false$ **then return** false;
23 **return** true;

A final neighbour is constructed by Algorithm 5.4. Suppose, the algorithm is called with $(t_0, n)$ where $n = 1$, indicating that each DT of therapy $t_0$ (except for the last one) should be assigned to the day of its subsequent DT. The loop in Line 2 removes all day assignments of therapy $t_0$ from the schedule. In Line 6 every DTs, except for the last one, is assigned to the day of its subsequent DT. Finally, the loop in Line 18 assigns the last DT to the next working day which provides the required resources. The functions occuring in this algorithm are explained in Section 5.3.

**Algorithm 5.4:** Decoding a neighbour from $\widetilde{\mathcal{N}}^{Z3}$

**Input:** An encoded neighbour $(t_0, n)$

$U_{\text{part}} \leftarrow \{u \in U_{t_0} \mid \max(0, -n) \leq u \leq \min(\max(U_{t_0}) - n, \max U_{t_0})\}$;
**for** $u \in U_{t_0}$ **do**
  $Z^{\text{old}}_{t_0,u} \leftarrow Z_{t_0,u}$;
  $Z_{t_0,u} \leftarrow -1$; `// remove day assignment in solution`
**end**
**for** $u \in U_{part}$ **do**
  **if** $schedule\_dt(S, t_0, u, Z^{old}_{t_0,u+n}) = false$ **then return** false;
**end**
$U_{\text{rest}} \leftarrow U_{t_0} \setminus U_{\text{part}}$;
**if** $n < 0$ **then**
  $d \leftarrow Z^{\text{old}}_{t_0,0}$;
  **for** $(u \leftarrow \max U_{rest}; u \geq 0; u \leftarrow u - 1)$ **do**
    $d \leftarrow$ schedule_dt_before$(S, t_0, u, d)$;
    **if** $d = -1$ **then return** false;
  **end**
**else if** $n > 0$ **then**
  $d \leftarrow Z^{\text{old}}_{t_0,\tau_{t_0}-1}$;
  **for** $(u \leftarrow \min U_{rest}; u < \tau_{t_0}; u \leftarrow u + 1)$ **do**
    $d \leftarrow$ schedule_dt_after$(S, t_0, u, d)$;
    **if** $d = -1$ **then return** false;
  **end**
**end**
**if** $check\_constraint\_n^{twmin}(S, t_0) = false$ **then return** false;
**return** true;

CHAPTER 6

# 2-Phase Approach

This chapter presents a heuristic search method for solving the PTPSP, which acts in two phases. The first phase of the 2-Phase Approach (2PA) starts with a day assignment created by TWCH and applies a VND method on it, which is composed solely of inter-day neighbourhood structures. After reaching a local optimum, the second phase starts by greedily assigning the DTs to starting times using TWCH's time assignment phase. Afterwards a VND which consists of the intra-day neighbourhood structures $\mathcal{N}^{S1}_{2,D'}$ and $\mathcal{N}^{S2}_{2,D'}$ optimizes the DTs' starting times without changing their day assignment. The algorithm terminates when this VND reaches a local optimum.

An important aspect which has to be considered is that the $\widetilde{S}_{t,v}$ values may become suboptimal during a move operation and need to be readjusted by solving the LP model 4.4. If this is done before every solution evaluation, the objective function is guaranteed to compute an accurate value. However, preliminary tests have shown that the local search is by far too time consuming if new values for the $\widetilde{S}_{t,v}$ variables are computed for each evaluated neighbour. This is mainly due to the fact that each time the values of the $\widetilde{S}_{t,v}$ variables are updated, possible improvements can be found on days other than the currently considered one. In order to reach sufficient performance of the local search component, we have to limit the number of times the $\widetilde{S}_{t,v}$ variables are computed. We achieve this by including a pseudo neighbourhood structure $\widetilde{\mathcal{S}}$ as last considered neighbourhood of the second VND. This $\widetilde{\mathcal{S}}(Z, S)$ pseudo neighbourhood contains exactly one neighbour resulting from a reevaluation of the model from Section 4.4. The $\widetilde{S}_{t,v}$ variables are updated only when the intra-day neighbourhood structures have reached a local optimum on all days. If the recomputation improves the objective value, the first intra-day neighbourhood structure from the VND is evaluated again.

A crucial aspect of the 2PA is the choice of neighbourhood structures and the order in which they are used in the VNDs. Since we did not know with certainty which neighbourhood structures perform best in this context, we made a decision on the basis of statistical tests. Section 9.2 explains how this was done.

CHAPTER 7

# Iterated Greedy Approach

This chapter presents an IG algorithm for solving the PTPSP. As explained in Chapter 3, an IG algorithm starts with a solution created by a construction heuristic and repeatedly applies a destruction and a construction phase to obtain better solutions. Frequently, a local search algorithm is applied to the initial solution and after the construction phase to further boost the performance. Preliminary results of this approach have been published in [MHRR17]. Note, however, that $\gamma^{\text{intraw}}$ and $\gamma^{\text{interw}}$ were both set to 0 in this paper, instead of 0.1.

The initial solution of the proposed IG is computed using TWCH from [MRSR16], which is explained in Chapter 2. The remaining components of the IG are discussed in the following sections.

## 7.1 Local Search

The design of the neighbourhood 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 neighbourhoods that can be searched rather fast, still allowing to complete a reasonable number of iterations of the IG, while improving the solution significantly in most cases. To achieve this, we restrict ourselves to a local search method that is only able to modify the starting times of DTs, i.e., the day assignment is considered to be fixed. Hence, we are only able to improve on the objective function terms that consider the use of extended service windows and the violation of the interweek and intraweek soft constraints. The local search uses the neighbourhood structure $\mathcal{N}^{S1}_{2,D'}$, which maps a solution to a set of all possible neighbours that result from exchanging two DTs on some working day $d \in D'$. To accelerate the local search procedure we restrict the neighbourhood structure to the most promising moves by applying the filters *adjacent* and *gap* in such a way that a move is only evaluated if both considered DTs are

either adjacent in sequence $((t,u)_i)_{i=1}^{|G_d|}$ or it is likely that an exchange produces a tighter scheduled day.

Similar to the VND of the 2PA from Chapter 6, $\mathcal{N}_{2,D'}^{S1}$ is combined with a pseudo neighbourhood structure $\widetilde{\mathcal{S}}$ in a VND-like way, which causes the $\widetilde{S}_{t,v}$ variables to be updated by reevaluating the model from Section 4.4.

## 7.2 Destruction and Construction

The destruction and construction phase of the IG from Maschler et al. [MRSR16] consists of removing the DTs of randomly selected therapies from the schedule, followed by applying TWCH's day assignment for the removed therapies and solving TWCH's time assignment from scratch. However, w.r.t. the local search algorithm from Section 7.1 discarding the whole time assignment during destruction and construction is disadvantageous since no parts of the old time assignment of an affected day are transferred to the new one. We overcome this drawback by replacing TWCH's day assignment with an insertion heuristic which preserves the sequence of unchanged DTs and inserts the removed ones in a greedy way.

Algorithm 7.1 shows the used destruction and construction phase in detail. It starts by invalidating the day and time assignment of $\beta \cdot n_T$ randomly selected therapies, where $\beta \in (0,1]$ is the destruction rate. Afterwards, TWCH's day assignment is applied to reassign the DTs of the removed therapies to potentially new days. The insertion heuristic for the time assignment, which is inspired by the NEH heuristic [NEH83], is defined in the foreach loop at Line 3 and is applied for each working day. It starts by initializing $G_d$ to the set of DTs that have been assigned to day $d$ and which have not been removed by the destruction phase. Analogously, $G'_d$ is defined as the set containing all DTs assigned to day $d$ that have been removed and for which a new starting time has to be found. As with the neighbourhood structures, a unique sequence $((t,u)_i)_{i=1}^{|G_d|}$, named $s$, can be defined by sorting the DTs according to the time they first require the beam resource. In each step a not yet considered random DT from $G'_d$ is inserted at all possible positions of the sequence $s$ and scheduled using Algorithm 5.1. All of these $|G_d|+1$ partial time assignments are compared and finally the best one is kept. To this end a sequence is considered better if the objective value, which depends on the extended time used and the violation of the interweek and intraweek soft constraints, is smaller. In case of a tie we prefer the option with the smaller makespan. The rationale behind the latter criterion is that in particular after destruction 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.

**Algorithm 7.1:** Destruction and construction phases.

**Input:** A solution $(Z, S)$

select a set $T'$ of $\beta \cdot n_T$ random therapies and remove their day and time assignments;
apply TWCH's day assignment for the set of removed therapies;
**foreach** $d \in D$ **do**
  $G_d \leftarrow \{(t,u) \mid t \in T \setminus T', u \in U_t, Z_{t,u} = d\}$;
  $G'_d \leftarrow \{(t,u) \mid t \in T', u \in U_t, Z_{t,u} = d\}$;
  **foreach** $(t',u') \in G'_d$ **do**
    let $s$ be a sequence $((t,u)_i)_{i=1}^{|G_d|}$ of $G_d$ resulting by sorting the elements according to $S_{t,u} + P^{\text{start}}_{t,u,\text{B}}$;
    best_obj $\leftarrow \infty$; best_MS $\leftarrow \infty$; $s'_{\text{best}} \leftarrow ()$;
    **for** $i \leftarrow 1$ **to** $|G_d| + 1$ **do**
      $s' \leftarrow ((t,u)_1, \ldots, (t,u)_{i-1}, (t',u'), (t,u)_i, \ldots, (t,u)_{|G_d|})$;
      schedule $s'$ with Algorithm 5.1;
      obj $\leftarrow$ objective value of the current partial solution;
      MS $\leftarrow$ makespan of current day $d$;
      **if** obj $<$ best_obj $\vee$ (obj $=$ best_obj $\wedge$ MS $<$ best_MS) **then**
        best_obj $\leftarrow$ obj; best_MS $\leftarrow$ MS; $s'_{\text{best}} \leftarrow s'$;
      **end**
    **end**
    schedule $s'_{\text{best}}$ with Algorithm 5.1;
    $G_d \leftarrow G_d \cup \{(t',u')\}$;
  **end**
**end**

## 7.3 Improved Iterated Greedy

Compared to the 2PA, the IG has the advantage of being able to escape local optima due to the destruction and construction phase. However, it can be expected that the local search phase of the IG is less effective than the VND of the 2PA because it uses only a single neighbourhood to optimize a solution. So an obvious improvement to the IG is to use a VND in the local search phase. To be precise, the improved IG works as follows.

We first generate a day assignment using the first phase of TWCH. This day assignment is improved by a VND that is composed of inter-day neighbourhood structures. Then the DTs are greedily assigned to starting times using the time assignment phase of TWCH. Afterwards, the starting times are optimized by a VND which consists of intra-day neighbourhoods.

The destruction and construction phase removes the DTs of randomly selected therapies, which are then reassigned by applying TWCH's day assignment. The day assignment of these DTs is then improved using a VND that consists of inter-day neighbourhoods.

Then TWCH's time assignment phase is applied, followed by a VND that optimizes the starting times of the DTs. Note that the VNDs of the construction phase comprise the same neighbourhood structures as the VNDs that are applied on the initial solution. Section 9.2 shows how the neighbourhood structures were selected.

CHAPTER 8

# Postprocessing

All neighbourhood structures use Algorithm 5.1 for assigning the DTs to starting times. This method is designed to generate a tight schedule with as little use of extended time as possible in an efficient way, which makes it a reasonable decoder for a local search method. However, a schedule where each DT is scheduled as early as possible may not be optimal due to the intraweek and interweek soft constraints which penalize a solution where the DTs of a therapy do not start at roughly the same time. In this section, an LP model is presented which is used to improve a given schedule by introducing breaks between consecutive DTs. It is indended to be applied as a post-processing step.

We assume that a start solution is given by

- $Z' = \{Z'_{t,u} \in D' \mid t \in T, u \in U_t\}$ denoting the days when all the DTs are scheduled,
- $S' = \{S'_{t,u} \geq 0 \mid t \in T, u \in U_t\}$ denoting the starting times of the DTs on the respective days, and
- $\widetilde{S}'_{t,v}$ for each therapy $t \in T$ and each week $v \in V$ denoting the nominal starting time of the DTs within the whole week $v$ when the treatment takes place in this week.

The idea of the model is to shift DTs s.t. their respective order w.r.t. the use of the beam, specified by the input solution, does not change. Moreover, a DT's position w.r.t. unavailability periods of its required resources is not modified. Fixing all these properties allows us to formulate a linear program which can be efficiently solved.

We formulate the model for a specific day $d \in D'$ in week $v \in V : d \in D_v$ in terms of the following additional sets and constants:

- $T^d = \{t \mid t \in T, \exists u \in U_t, Z'_{t,u} = d\}$, denoting the therapies scheduled on day $d$.

- $G^d = \{(t,u) \mid t \in T^d, u \in U_t, Z'_{t,u} = d\}$, denoting the DTs on day $d$.
- $\mathcal{G}^d = \{(t,u,t',u') \mid (t,u) \in G^d, (t',u') \in G^d, (\forall (t^*,u^*) \in G^d : S'_{t^*,u^*} + P^{\text{start}}_{t^*,u^*,B} \leq S'_{t,u} + P^{\text{start}}_{t,u,B} \vee S'_{t^*,u^*} + P^{\text{start}}_{t^*,u^*,B} \geq S'_{t',u'} + P^{\text{start}}_{t',u',B})\}$, defining the successor for every DT w.r.t. the beam resource.
- $\overline{\mathcal{W}}_{r,d}$ denoting the set of all unavailability periods $[\overline{W}^{\text{start}}_{r,d,w}, \overline{W}^{\text{end}}_{r,d,w})$ for resource $r \in R$ on day $d$.
- $K^{\text{before}}_{t,u,r} = \max\left(\left\{\widehat{W}^{\text{start}}_{r,d}\right\} \cup \left\{\overline{W}^{\text{end}}_{r,d,w} \;\middle|\; \begin{bmatrix}\overline{W}^{\text{start}}_{r,d,w}, \overline{W}^{\text{end}}_{r,d,w}\end{bmatrix} \in \overline{\mathcal{W}}_{r,d}, \\ \overline{W}^{\text{start}}_{r,d,w} < S'_{t,u} + P^{\text{start}}_{t,u,r}\right\}\right)$ for every $(t,u) \in G^d$ and $r \in Q_{t,u}$, denoting the end of the last unavailability period of resource $r$ before DT $(t,u)$.
- $K^{\text{after}}_{t,u,r} = \min\left(\left\{\widehat{W}^{\text{end}}_{r,d}\right\} \cup \left\{\overline{W}^{\text{start}}_{r,d,w} \;\middle|\; \begin{bmatrix}\overline{W}^{\text{start}}_{r,d,w}, \overline{W}^{\text{end}}_{r,d,w}\end{bmatrix} \in \overline{\mathcal{W}}_{r,d}, \\ \overline{W}^{\text{start}}_{r,d,w} > S'_{t,u} + P^{\text{start}}_{t,u,r}\right\}\right)$ for each $(t,u) \in G^d$ and $r \in Q_{t,u}$, denoting the start of the first unavailability period of resource $r$ after DT $(t,u)$.

The model uses

- starting time variables $S_{t,u}$ for each DT $(t,u) \in G^d$,
- variables $\sigma^{\text{intraw}}_{t,u}$ for tracking the intraweek costs for each DT $(t,u) \in G^d$,
- variables $S^{\text{first}}_{r,d}$ and $S^{\text{last}}_{r,d}$ for tracking the first and last time, respectively, each resource $r \in R$ is used, and
- variables $y_{t,u,r}$ denote the release time of resource $r \in R$ for each DT $(t,u) \in G^d$.

The model reads as follows:

$$\begin{aligned}
\min\ & \gamma^{\text{extfront}} \sum_{r \in \widehat{R}} \max\left(W^{\text{start}}_{r,d} - S^{\text{first}}_{r,d}, 0\right) + \\
& \gamma^{\text{extback}} \sum_{r \in \widehat{R}} \max\left(S^{\text{last}}_{r,d} - W^{\text{end}}_{r,d}, 0\right) + \\
& \gamma^{\text{intraw}} \sum_{(t,u) \in G^d} \sigma^{\text{intraw}}_{t,u} + \\
& \gamma^{\text{scatter}} \sum_{r \in R^{\text{scatter}}} \varphi^{\text{scatter}}_r \\
& \qquad \left(S^{\text{last}}_{r,d} - S^{\text{first}}_{r,d} - \sum_{\substack{(t,u) \in G^d, \\ r \in Q_{t,u}}} (P^{\text{end}}_{t,u,r} - P^{\text{start}}_{t,u,r})\right)
\end{aligned} \tag{1}$$

$$
\begin{aligned}
\text{s.t. } & |S_{t,u} - \widetilde{S}'_{t,v}| - \sigma^{\text{intraw}}_{t,u} \leq \delta^{\text{intraw}} && \forall (t,u) \in G^d : u > 0 \vee t \in \widetilde{T} && (2)\\
& S^{\text{first}}_{r,d} \leq S_{t,u} + P^{\text{start}}_{t,u,r} && \forall (t,u) \in G^d, \forall r \in Q_{t,u} && (3)\\
& S^{\text{last}}_{r,d} \geq y_{t,u,r} && \forall (t,u) \in G^d, \forall r \in Q_{t,u} && (4)\\
& S_{t,u} + P^{\text{end}}_{t,u,r} \leq y_{t,u,r} && \forall (t,u) \in G^d, \forall r \in Q_{t,u} && (5)\\
& y_{t,u,r} \leq y_{t',u',r} && \forall (t,u,t',u') \in \mathcal{G}^d, \forall r \in R \setminus Q_{t',u'} && (6)\\
& y_{t,u,r} \leq S_{t',u'} + P^{\text{start}}_{t',u',r} && \forall (t,u,t',u') \in \mathcal{G}^d, \forall r \in Q_{t',u'} && (7)\\
& K^{\text{before}}_{t,u,r} \leq S_{t,u} + P^{\text{start}}_{t,u,r} && \forall (t,u) \in G^d, \forall r \in Q_{t,u} && (8)\\
& S_{t,u} + P^{\text{end}}_{t,u,r} \leq K^{\text{after}}_{t,u,r} && \forall (t,u) \in G^d, \forall r \in Q_{t,u} && (9)\\
& S^{\text{L}}_{t,u} \leq S_{t,u} \leq S^{\text{U}}_{t,u} && \forall (t,u) \in G^d && (10)\\
& \widehat{W}^{\text{start}}_{r,d} \leq S^{\text{first}}_{r,d} \leq S^{\text{last}}_{r,d} \leq \widehat{W}^{\text{end}}_{r,d} && \forall r \in R && (11)\\
& \sigma^{\text{intraw}}_{t,u} \geq 0 && \forall (t,u) \in G^d && (12)\\
& \widehat{W}^{\text{start}}_{r,d} \leq y_{t,u,r} \leq \widehat{W}^{\text{end}}_{r,d} && \forall (t,u) \in G^d, \forall r \in R && (13)
\end{aligned}
$$

- (1) provides the objective function.
- (2) calculates the intraweek violation.
- (3), (4) set variables that track the earliest and latest time each resource is used.
- (5) sets release time for used resources.
- (6) set release time for unused resources.
- (7) sets a DT's starting time, s. t. it does not use a previously used resource.
- (8), (9) ensure that the unavailability windows are respected.
- The remaining constraints impose domain restrictions; especially note the $S^{\text{first}}_r \leq S^{\text{last}}_r$.

CHAPTER 9

# Computational Study

This chapter shows the experimental results of local search using the individual neighbourhood structures, the two-phase approach and the iterated greedy metaheuristic.

The benchmark instances were created in such a way to describe the expected situation at MedAustron as well as possible. We use 40 benchmark instances which can be divided into four groups of 100, 150, 200 and 300 therapies per instance, respectively. In the following, an instance name encodes the number of therapies followed by a consecutive number. Each therapy has to start within a window of 14 days and consists of up to 35 DTs, reflecting the duration of real particle therapies. Each DT requires as resources at least the beam for a relatively short time interval and one of three irradiation rooms. The beam and the three rooms are regularly available from $\widetilde{W}_d^{\text{start}}$ for 14 hours and have an extended availability period of 10 hours. There are further resources, such as the personnel, which are, however, sufficiently dimensioned to be not the primary reasons of substantial use of extended service time. The first DT of each therapy needs to be provided before noon on Monday or Tuesday. This constraint is modeled by introducing an additional resource that spans half of the regular opening time on these two weekdays. For more details on the instance generation, see [MRSR16].

## 9.1 Local Search

This section shows the performance of the local search method, defined by Algorithm 3.1, using the neighbourhood structures defined in Chapter 5. We will show that the selection of the neighbourhood structure has a great impact on the final solution's objective value, the number of moves which are needed to reach a local optimum as well as the total runtime of the algorithm. We will also compare different step functions and show that the next improvement step function with a randomized order of neighbours is superior to the other ones.

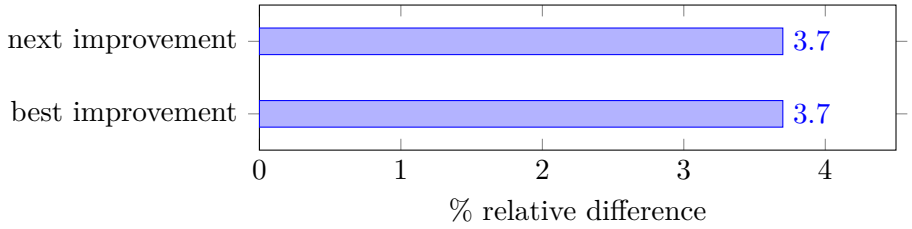


Figure 9.1: Relative differences between the average objective value of TWCH and the average objective values of a local search with both step functions using neighbourhood structure $\mathcal{N}_{2,d}^{S1}$. All three filters are enabled. The values are calculated as $100\% \cdot (obj_T - obj_L)/obj_T$, where $obj_T$ denotes the average objective value of TWCH and $obj_{\mathrm{L}}$ represents the average objective value of one of the local search methods.

The initial solution for the local search method is generated with TWCH. Table 9.1 shows the average objective values for these initial solutions together with their standard deviations on all instances over 30 runs. All experiments in this section are performed without a time limit.

### 9.1.1 Intra-Day Neighbourhood Structures

Because local search can be used with both next and best improvement, we first have to find out which step function is better w.r.t. our problem setting. To that end, we applied the local search method first with next improvement and then with best improvement using the neighbourhood structure $\mathcal{N}_{2,d}^{S1}$. We enabled all three filters in this experiment because it turned out that a local search with $\mathcal{N}_{2,d}^{S1}$ using the best improvement step function may take over an hour to reach a local optimum if some of the filters are disabled. When applying next improvement, the enumeration of the neighbours is also randomized. Table 9.2 shows the performance of local search with both step functions using neighbourhood structure $\mathcal{N}_{2,d}^{S1}$ with all filters enabled. Figure 9.1 shows the relative difference between the average objective values of these local search methods and the average objective value of TWCH.

The test results show that a single move with the next improvement step function takes much less time because each step searches in general only a small part of the neighbourhood, whereas the best improvement step function searches the whole neighbourhood for the best solution in each step. A local search using the best improvement strategy even exceeds an hour on the largest instances if all filters are disabled. And such long convergence times are unacceptable if the neighbourhood to be explored is only a part of a more complex heuristic search method like the 2PA or IG. A more thorough analysis of individual runs has shown that randomizing the order of the neighbours helped to remove a bias towards exchanges at the beginning of the days. This leads to shorter iteration times, especially at the end of a local search run. Regarding the average objective values, however, there is almost no difference. A Wilcoxon rank sum test with a significance

| Instance | *obj* | $\sigma(obj)$ | *t [s]* |
|---|---|---|---|
| 100-01 | 151.36 | 5.16 | 0.30 |
| 100-02 | 209.27 | 13.30 | 0.30 |
| 100-03 | 98.70 | 4.15 | 0.30 |
| 100-04 | 171.57 | 6.29 | 0.40 |
| 100-05 | 164.75 | 8.41 | 0.30 |
| 100-06 | 130.85 | 6.08 | 0.30 |
| 100-07 | 132.00 | 6.34 | 0.30 |
| 100-08 | 179.88 | 6.03 | 0.30 |
| 100-09 | 97.06 | 6.38 | 0.30 |
| 100-10 | 164.35 | 5.72 | 0.40 |
| 150-01 | 226.93 | 8.08 | 0.60 |
| 150-02 | 367.83 | 8.97 | 0.60 |
| 150-03 | 276.50 | 7.22 | 0.60 |
| 150-04 | 217.41 | 6.86 | 0.60 |
| 150-05 | 191.88 | 9.67 | 0.50 |
| 150-06 | 394.33 | 10.37 | 0.60 |
| 150-07 | 342.74 | 9.10 | 0.50 |
| 150-08 | 342.38 | 6.09 | 0.60 |
| 150-09 | 306.89 | 9.82 | 0.50 |
| 150-10 | 250.77 | 9.62 | 0.60 |
| 200-01 | 373.13 | 10.45 | 0.90 |
| 200-02 | 368.83 | 6.55 | 0.80 |
| 200-03 | 285.77 | 7.20 | 0.90 |
| 200-04 | 345.90 | 9.54 | 0.80 |
| 200-05 | 318.00 | 9.03 | 0.90 |
| 200-06 | 288.30 | 9.57 | 0.90 |
| 200-07 | 221.67 | 7.35 | 0.80 |
| 200-08 | 226.95 | 9.24 | 0.80 |
| 200-09 | 231.98 | 8.48 | 0.80 |
| 200-10 | 501.41 | 9.01 | 0.80 |
| 300-01 | 335.84 | 10.49 | 1.70 |
| 300-02 | 542.20 | 16.30 | 1.70 |
| 300-03 | 353.66 | 13.43 | 1.60 |
| 300-04 | 576.11 | 16.46 | 1.70 |
| 300-05 | 340.53 | 8.66 | 1.60 |
| 300-06 | 375.81 | 9.77 | 1.70 |
| 300-07 | 296.57 | 9.08 | 1.60 |
| 300-08 | 307.13 | 10.96 | 1.60 |
| 300-09 | 258.59 | 6.73 | 1.60 |
| 300-10 | 310.35 | 10.45 | 1.50 |

Table 9.1: Average objective values *obj* of 30 runs, corresponding standard deviations $\sigma(obj)$, and median processing times for TWCH

| Instance | $\mathcal{N}_{2,d}^{S1}$ (next improvement) | | | | $\mathcal{N}_{2,d}^{S1}$ (best improvement) | | | |
|---|---|---|---|---|---|---|---|---|
| | *obj* | $\sigma(obj)$ | *Moves* | *t [s]* | *obj* | $\sigma(obj)$ | *Moves* | *t [s]* |
| 100-01 | **145.27** | 6.31 | 181.50 | 0.50 | 146.24 | 5.50 | 160.50 | 5.50 |
| 100-02 | **204.35** | 11.15 | 151.50 | 0.50 | 205.88 | 10.53 | 134.00 | 5.50 |
| 100-03 | 94.47 | 4.32 | 93.00 | 0.40 | **93.24** | 3.47 | 79.00 | 2.20 |
| 100-04 | **166.33** | 7.27 | 106.00 | 0.50 | 168.76 | 8.03 | 91.00 | 3.20 |
| 100-05 | 159.17 | 6.90 | 123.50 | 0.50 | **158.30** | 7.29 | 117.50 | 4.40 |
| 100-06 | 127.09 | 5.90 | 98.50 | 0.50 | **125.66** | 7.10 | 79.50 | 2.40 |
| 100-07 | 129.26 | 8.18 | 87.00 | 0.50 | **129.12** | 5.40 | 71.50 | 2.10 |
| 100-08 | 176.69 | 6.64 | 120.50 | 0.50 | **175.66** | 6.48 | 104.00 | 3.70 |
| 100-09 | 92.65 | 6.26 | 91.00 | 0.40 | **92.15** | 5.59 | 74.00 | 2.10 |
| 100-10 | 159.83 | 6.54 | 96.50 | 0.50 | **159.27** | 6.42 | 90.00 | 3.30 |
| 150-01 | 223.63 | 8.81 | 170.50 | 0.90 | **219.88** | 9.25 | 147.50 | 9.50 |
| 150-02 | **360.15** | 9.98 | 219.50 | 0.90 | 362.35 | 7.76 | 198.50 | 12.80 |
| 150-03 | 273.08 | 5.84 | 127.50 | 0.80 | **272.47** | 7.51 | 109.00 | 5.90 |
| 150-04 | 209.81 | 5.69 | 182.00 | 0.90 | **208.37** | 6.87 | 157.00 | 9.00 |
| 150-05 | 183.47 | 8.71 | 172.50 | 0.70 | **181.98** | 8.11 | 150.00 | 7.30 |
| 150-06 | **386.90** | 10.45 | 226.00 | 0.90 | 387.79 | 8.09 | 193.00 | 11.10 |
| 150-07 | **332.12** | 8.09 | 170.00 | 0.80 | 334.80 | 8.53 | 149.00 | 7.90 |
| 150-08 | 339.50 | 7.59 | 140.00 | 0.80 | **335.75** | 8.44 | 120.00 | 6.10 |
| 150-09 | **299.35** | 10.14 | 171.00 | 0.80 | 300.52 | 9.79 | 157.50 | 8.60 |
| 150-10 | **242.04** | 9.08 | 200.00 | 0.90 | 242.08 | 8.05 | 184.00 | 10.50 |
| 200-01 | 366.02 | 8.24 | 246.50 | 1.20 | **362.13** | 8.59 | 215.50 | 18.20 |
| 200-02 | **357.07** | 5.26 | 292.50 | 1.30 | 357.84 | 5.92 | 247.50 | 20.20 |
| 200-03 | **276.39** | 6.56 | 167.50 | 1.20 | 278.27 | 5.83 | 151.00 | 9.30 |
| 200-04 | **336.43** | 7.05 | 254.50 | 1.20 | 338.68 | 7.57 | 227.50 | 18.40 |
| 200-05 | **302.81** | 8.57 | 306.50 | 1.30 | 304.53 | 8.48 | 257.00 | 20.90 |
| 200-06 | 279.26 | 6.29 | 207.00 | 1.30 | **278.19** | 4.97 | 176.50 | 15.40 |
| 200-07 | 213.87 | 7.02 | 226.50 | 1.20 | **212.36** | 7.23 | 194.00 | 14.30 |
| 200-08 | **209.50** | 7.98 | 296.50 | 1.20 | 211.79 | 10.20 | 239.00 | 17.40 |
| 200-09 | 221.60 | 6.90 | 254.00 | 1.20 | **219.61** | 6.91 | 220.00 | 15.10 |
| 200-10 | **491.50** | 7.13 | 261.00 | 1.20 | 493.57 | 7.39 | 221.50 | 17.00 |
| 300-01 | 314.72 | 12.24 | 376.00 | 2.30 | **314.49** | 9.98 | 332.00 | 39.70 |
| 300-02 | 522.09 | 16.17 | 415.50 | 2.30 | **520.66** | 12.70 | 360.50 | 42.60 |
| 300-03 | **333.45** | 10.72 | 436.00 | 2.20 | 334.40 | 11.55 | 356.00 | 39.10 |
| 300-04 | **560.93** | 14.12 | 307.00 | 2.40 | 564.59 | 17.84 | 280.50 | 33.90 |
| 300-05 | **319.66** | 10.30 | 363.00 | 2.30 | 319.70 | 10.41 | 304.00 | 33.30 |
| 300-06 | **355.57** | 11.05 | 396.00 | 2.30 | 356.74 | 9.82 | 337.00 | 39.60 |
| 300-07 | 277.07 | 8.94 | 444.00 | 2.30 | **274.94** | 8.27 | 354.50 | 43.20 |
| 300-08 | 289.37 | 10.89 | 413.50 | 2.20 | **288.41** | 11.84 | 351.00 | 37.40 |
| 300-09 | **237.50** | 8.13 | 412.00 | 2.20 | 237.71 | 8.31 | 329.00 | 40.30 |
| 300-10 | 292.06 | 8.94 | 432.00 | 2.10 | **289.66** | 10.73 | 374.00 | 40.90 |



Table 9.2: Average objective values *obj* of 30 runs, corresponding standard deviations $\sigma(obj)$ as well as median number of moves and median times for a local search with $\mathcal{N}_{2,d}^{S1}$ using next improvement and best improvement. All three filters are enabled. The initial solutions were generated with TWCH.

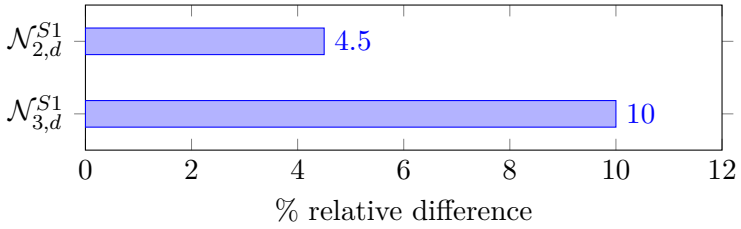


Figure 9.2: Relative differences between the average objective value of TWCH and the average objective values of a local search with $\mathcal{N}^{S1}_{2,d}$ and $\mathcal{N}^{S1}_{3,d}$. The values are calculated as $100\% \cdot (obj_T - obj_L)/obj_T$, where $obj_T$ denotes the average objective value of TWCH and $obj_{\mathrm{L}}$ represents the average objective value of one of the local search methods.

level of 95% shows that the best improvement step function indeed does not perform significantly better. Because the objective values with both step functions are similarly good, the running time is the decisive criterion. Therefore, the next improvement step function with a randomized order of neighbours is used in all later experiments.

In Section 5.1 we described a more general neighbourhood structure to permute $c$ DTs within a day $d$, denoted as $\mathcal{N}^{S1}_{c,d}$. To see the effects of a large number of permuted DTs, we have executed one local search with $\mathcal{N}^{S1}_{2,d}$ and another one with $\mathcal{N}^{S1}_{3,d}$ on the benchmark instances and compared their results. It turned out that the neighbourhood size is too large for the local search to converge with $c = 3$ in a reasonable time if no filter is used. Hence, we limited the search space for both local search runs using the *adjacent* filter. Table 9.3 shows the results for all benchmark instances. It can be seen that the solutions generated with $c = 3$ are on average by 5% to 8% better, which is due to the fact that $\mathcal{N}^{S1}_{3,d}$ can generate solutions which are unreachable just by a DT exchange. However, the computation time increases significantly with increasing $c$. Figure 9.2 depicts the relative difference between the average objective values of the local search methods and the average objective value of TWCH.

With regard to the filters, it can be expected that a local search that uses $\mathcal{N}^{S1}_{2,d}$ without any filter generates the best solutions. Indeed, Figure 9.3 as well as Table 9.4, which summarize the influence of the filters on the performance of the local search, show that a local search produced the best solutions on average for most instances if no filter was used. Moreover, this table suggests that the solutions found by the local search which uses the *gap* filter have very similar objective values than the ones produced by the local search which does not utilize any filter. This impression can be observed by a Wilcoxon rank sum test with a significance level of 95% showing that for only 8 out of 40 instances significantly better solutions can be found without using a filter. However, the *gap* filter removes so many worse solutions from the search space that the local search which uses this filter reaches a local optimum twice as fast for most instances. It can be concluded that it is not advisable to use the neighbourhood structure $\mathcal{N}^{S1}_{2,d}$ without any filter if convergence time is a critical factor. When comparing the results of a local search

| Instance | $\mathcal{N}_{2,d}^{S1}$ | | | | $\mathcal{N}_{3,d}^{S1}$ | | | |
|---|---|---|---|---|---|---|---|---|
| | *obj* | $\sigma(obj)$ | *Moves* | *t [s]* | *obj* | $\sigma(obj)$ | *Moves* | *t [s]* |
| 100-01 | 145.54 | 7.11 | 224.00 | 0.80 | **137.81** | 5.85 | 419.00 | 22.30 |
| 100-02 | 200.15 | 13.18 | 192.00 | 0.80 | **189.54** | 10.32 | 422.50 | 28.50 |
| 100-03 | 92.37 | 3.47 | 125.00 | 0.60 | **91.57** | 4.05 | 265.50 | 26.10 |
| 100-04 | 168.63 | 7.84 | 150.50 | 0.70 | **159.62** | 6.95 | 357.50 | 27.80 |
| 100-05 | 156.04 | 5.88 | 161.00 | 0.70 | **148.43** | 6.17 | 413.50 | 22.40 |
| 100-06 | 125.19 | 7.02 | 150.00 | 0.70 | **121.90** | 7.06 | 339.00 | 30.00 |
| 100-07 | 127.14 | 8.25 | 120.00 | 0.70 | **120.86** | 5.94 | 364.00 | 24.80 |
| 100-08 | 173.52 | 4.90 | 179.00 | 0.70 | **165.45** | 5.62 | 483.00 | 34.90 |
| 100-09 | 90.33 | 4.67 | 145.00 | 0.60 | **82.54** | 6.00 | 399.50 | 34.80 |
| 100-10 | 158.95 | 4.11 | 130.00 | 0.70 | **153.79** | 5.12 | 305.50 | 24.10 |
| 150-01 | 221.07 | 7.34 | 228.50 | 1.20 | **211.21** | 7.67 | 517.50 | 47.50 |
| 150-02 | 358.63 | 7.18 | 284.00 | 1.40 | **345.55** | 9.12 | 598.00 | 55.20 |
| 150-03 | 269.19 | 7.14 | 179.00 | 1.20 | **260.06** | 5.25 | 391.50 | 39.80 |
| 150-04 | 206.20 | 6.79 | 235.50 | 1.20 | **196.81** | 6.57 | 560.00 | 59.20 |
| 150-05 | 180.45 | 8.26 | 266.50 | 1.20 | **159.94** | 5.72 | 717.00 | 52.30 |
| 150-06 | 383.25 | 8.39 | 288.50 | 1.30 | **362.14** | 8.73 | 692.50 | 67.20 |
| 150-07 | 329.86 | 9.94 | 243.00 | 1.20 | **316.34** | 9.24 | 676.50 | 66.90 |
| 150-08 | 335.27 | 6.64 | 207.00 | 1.20 | **326.33** | 7.74 | 540.50 | 55.20 |
| 150-09 | 297.48 | 9.46 | 265.00 | 1.20 | **281.56** | 7.98 | 648.50 | 77.90 |
| 150-10 | 238.27 | 9.04 | 274.00 | 1.30 | **228.74** | 7.62 | 531.00 | 56.20 |
| 200-01 | 359.41 | 12.24 | 342.00 | 1.80 | **343.91** | 11.26 | 696.00 | 76.80 |
| 200-02 | 356.21 | 6.37 | 363.50 | 1.90 | **342.10** | 6.06 | 806.00 | 93.60 |
| 200-03 | 274.53 | 5.90 | 263.50 | 1.70 | **259.02** | 6.42 | 784.00 | 95.70 |
| 200-04 | 335.56 | 11.08 | 377.50 | 1.90 | **317.96** | 5.86 | 798.50 | 92.50 |
| 200-05 | 303.21 | 9.56 | 411.50 | 1.90 | **284.11** | 7.06 | 951.00 | 116.80 |
| 200-06 | 280.97 | 6.65 | 263.00 | 1.80 | **270.06** | 7.44 | 576.00 | 102.40 |
| 200-07 | 212.41 | 8.90 | 317.50 | 1.80 | **198.17** | 7.58 | 763.00 | 90.80 |
| 200-08 | 205.86 | 7.37 | 413.00 | 1.70 | **192.59** | 8.82 | 942.00 | 105.00 |
| 200-09 | 218.71 | 7.60 | 317.00 | 1.70 | **202.62** | 7.48 | 737.00 | 75.20 |
| 200-10 | 490.59 | 7.31 | 326.50 | 1.80 | **467.86** | 7.95 | 737.50 | 93.50 |
| 300-01 | 310.45 | 8.49 | 570.00 | 3.30 | **288.22** | 12.55 | 1360.00 | 193.80 |
| 300-02 | 514.20 | 15.62 | 536.50 | 3.40 | **490.20** | 12.15 | 1320.50 | 198.50 |
| 300-03 | 330.03 | 8.93 | 634.00 | 3.20 | **301.07** | 10.40 | 1477.50 | 196.40 |
| 300-04 | 558.36 | 14.65 | 420.50 | 3.20 | **523.42** | 15.87 | 1168.00 | 172.60 |
| 300-05 | 320.93 | 9.60 | 520.00 | 3.20 | **294.82** | 9.64 | 1313.50 | 187.20 |
| 300-06 | 355.87 | 10.78 | 540.50 | 3.30 | **334.26** | 8.46 | 1382.00 | 210.70 |
| 300-07 | 271.72 | 8.14 | 600.00 | 3.20 | **249.54** | 9.66 | 1429.50 | 225.90 |
| 300-08 | 285.08 | 10.47 | 601.00 | 3.20 | **263.99** | 8.18 | 1411.50 | 198.40 |
| 300-09 | 236.66 | 8.71 | 607.00 | 3.20 | **211.43** | 5.46 | 1446.50 | 235.70 |
| 300-10 | 288.54 | 8.32 | 568.50 | 3.10 | **252.64** | 9.33 | 1572.50 | 205.40 |



Table 9.3: Average objective values *obj* of 30 runs, corresponding standard deviations $\sigma(obj)$ as well as median number of moves and median times for a local search with $\mathcal{N}_{2,d}^{S1}$ and $\mathcal{N}_{3,d}^{S1}$. The *adjacent* filter is enabled, all others are disabled.

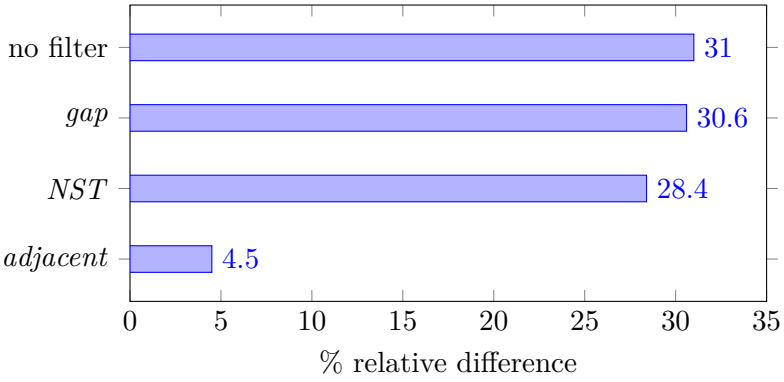


Figure 9.3: Relative differences between the average objective value of TWCH and the average objective values of a local search with different filters enabled. The values are calculated as $100\% \cdot (obj_T - obj_L)/obj_T$, where $obj_T$ denotes the average objective value of TWCH and $obj_\mathrm{L}$ represents the average objective value of one of the local search methods.

using the *gap* filter with the results of a local search using the *NST* filter by means of a Wilcoxon rank sum test with a significance level of 95%, we observed that significantly worse solutions were found when the *NST* filter was used for most instances. But the difference in the objective values is relatively small. However, it takes the local search with *gap* on average more than three times as long to converge than with the other filter because the *NST* filter drops many more worse solutions than *gap*. In other words, the improvement per time interval is larger with *NST* than with *gap*. This makes $\mathcal{N}_{2,d}^{S1}$ with enabled *NST* filter a good candidate for the first neighbourhood structure of a VND because the first neighbourhood structure in a VND is used more often than the other ones. With enabled *adjacent* filter, the local search provides almost no improvement because the size of the filtered neighbourhoods is only linear in the number of DTs per day and the filter nevertheless accepts many obviously bad solutions, especially on schedules containing sequences of DTs from two alternating rooms.

Table 9.5 and Figure 9.4 compare the results of two versions of local search: One using $\mathcal{N}_{2,d}^{S1}$ and another one using $\mathcal{N}_{d}^{S2}$. In both cases, there is no filter applied. It can be seen that it takes the version using $\mathcal{N}_{d}^{S2}$ about twice as long to converge. That is mainly because the neighbourhood $\mathcal{N}_{d}^{S2}(Z, S)$, containing $|G_d|(|G_d| - 1)$ solutions, is twice as large as $\mathcal{N}_{2,d}^{S1}(Z, S)$. Another observation we can make is that the local search method that uses $\mathcal{N}_{2,d}^{S2}$ finds significantly worse solutions. To understand the reason for that, one must know that the DTs of the initial solutions of TWCH are already well interleaved with regard to the room assignment. And on such solutions it is difficult to move a DT to a different place on the same day without inducing idle time for the beam. Suppose, for example, that every second DT is assigned to the same room and all resources are available throughout the day for the sake of simplicity. Then almost every move would

| Instance | $\mathcal{N}_{2,d}^{S1}$ (no filter) | | | $\mathcal{N}_{2,d}^{S1}$ $(gap)$ | | | $\mathcal{N}_{2,d}^{S1}$ $(NST)$ | | | $\mathcal{N}_{2,d}^{S1}$ $(adjacent)$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $obj$ | $\sigma(obj)$ | $t$ [s] | $obj$ | $\sigma(obj)$ | $t$ [s] | $obj$ | $\sigma(obj)$ | $t$ [s] | $obj$ | $\sigma(obj)$ | $t$ [s] |
| 100-01 | **107.73** | 5.36 | 19.40 | 110.76 | 4.89 | 8.70 | 112.34 | 5.72 | 2.70 | 145.54 | 7.11 | 0.80 |
| 100-02 | 151.08 | 8.15 | 21.50 | **150.09** | 6.74 | 10.40 | 154.81 | 8.21 | 3.40 | 200.15 | 13.18 | 0.80 |
| 100-03 | **68.57** | 2.61 | 16.90 | 69.44 | 2.77 | 6.90 | 71.98 | 3.30 | 2.50 | 92.37 | 3.47 | 0.60 |
| 100-04 | **127.49** | 5.69 | 19.90 | 129.28 | 5.56 | 8.50 | 131.70 | 7.06 | 2.70 | 168.63 | 7.84 | 0.70 |
| 100-05 | **123.28** | 6.13 | 19.70 | 124.32 | 6.81 | 9.80 | 130.63 | 6.95 | 2.40 | 156.04 | 5.88 | 0.70 |
| 100-06 | **94.04** | 5.16 | 20.40 | 95.65 | 5.51 | 8.20 | 95.65 | 4.85 | 3.00 | 125.19 | 7.02 | 0.70 |
| 100-07 | **92.49** | 4.79 | 20.40 | 92.57 | 3.63 | 8.90 | 93.63 | 6.03 | 2.90 | 127.14 | 8.25 | 0.70 |
| 100-08 | **138.69** | 4.47 | 19.90 | 141.03 | 5.74 | 9.80 | 145.08 | 4.83 | 2.80 | 173.52 | 4.90 | 0.70 |
| 100-09 | **58.48** | 5.22 | 18.60 | 59.90 | 4.78 | 7.60 | 62.22 | 4.57 | 2.40 | 90.33 | 4.67 | 0.60 |
| 100-10 | 124.74 | 4.68 | 19.50 | **124.11** | 4.87 | 9.00 | 128.96 | 4.69 | 2.60 | 158.95 | 4.11 | 0.70 |
| 150-01 | **163.59** | 7.49 | 37.90 | 165.68 | 6.25 | 15.80 | 170.18 | 7.02 | 5.30 | 221.07 | 7.34 | 1.20 |
| 150-02 | 287.47 | 5.97 | 40.30 | **284.63** | 6.55 | 19.70 | 298.05 | 6.26 | 6.20 | 358.63 | 7.18 | 1.40 |
| 150-03 | **215.73** | 6.80 | 37.10 | 216.71 | 4.74 | 14.50 | 224.30 | 6.32 | 4.90 | 269.19 | 7.14 | 1.20 |
| 150-04 | **144.92** | 5.31 | 38.00 | 146.17 | 5.95 | 16.60 | 151.78 | 6.03 | 5.20 | 206.20 | 6.79 | 1.20 |
| 150-05 | **125.83** | 5.27 | 30.60 | 127.33 | 6.34 | 13.50 | 133.66 | 5.01 | 4.70 | 180.45 | 8.26 | 1.20 |
| 150-06 | **314.04** | 5.80 | 36.80 | 316.27 | 6.37 | 17.90 | 329.14 | 7.13 | 4.70 | 383.25 | 8.39 | 1.30 |
| 150-07 | **279.07** | 6.34 | 35.00 | 279.89 | 8.14 | 14.60 | 281.11 | 7.47 | 5.10 | 329.86 | 9.94 | 1.20 |
| 150-08 | 273.61 | 6.33 | 43.40 | **270.33** | 8.20 | 18.70 | 275.21 | 6.28 | 6.40 | 335.27 | 6.64 | 1.20 |
| 150-09 | **242.49** | 6.55 | 40.30 | 243.23 | 7.88 | 18.20 | 247.78 | 7.14 | 6.00 | 297.48 | 9.46 | 1.20 |
| 150-10 | **170.27** | 6.12 | 39.10 | 171.94 | 7.05 | 17.00 | 178.53 | 4.71 | 5.30 | 238.27 | 9.04 | 1.30 |
| 200-01 | **288.65** | 8.84 | 45.20 | 289.30 | 9.29 | 20.30 | 299.13 | 5.16 | 6.90 | 359.41 | 12.24 | 1.80 |
| 200-02 | 286.44 | 3.67 | 47.10 | **284.89** | 4.23 | 20.80 | 292.26 | 4.34 | 6.60 | 356.21 | 6.37 | 1.90 |
| 200-03 | **203.00** | 4.70 | 47.50 | 203.87 | 5.95 | 19.90 | 209.42 | 5.71 | 7.00 | 274.53 | 5.90 | 1.70 |
| 200-04 | **240.05** | 5.95 | 50.30 | 240.14 | 6.66 | 23.10 | 248.22 | 5.51 | 7.60 | 335.56 | 11.08 | 1.90 |
| 200-05 | **219.95** | 4.52 | 48.20 | 220.06 | 7.17 | 20.80 | 229.98 | 5.88 | 6.90 | 303.21 | 9.56 | 1.90 |
| 200-06 | 202.48 | 3.52 | 56.20 | **202.32** | 4.35 | 22.40 | 205.37 | 4.66 | 8.50 | 280.97 | 6.65 | 1.80 |
| 200-07 | **141.00** | 4.19 | 47.60 | 142.28 | 6.44 | 20.10 | 146.72 | 6.78 | 6.80 | 212.41 | 8.90 | 1.80 |
| 200-08 | **142.65** | 5.62 | 44.00 | 145.24 | 5.37 | 18.40 | 149.88 | 6.94 | 6.40 | 205.86 | 7.37 | 1.70 |
| 200-09 | **150.38** | 5.87 | 40.10 | 153.53 | 7.03 | 18.00 | 157.73 | 8.31 | 5.70 | 218.71 | 7.60 | 1.70 |
| 200-10 | **400.52** | 7.63 | 48.50 | 401.13 | 7.33 | 22.20 | 420.82 | 8.24 | 7.70 | 490.59 | 7.31 | 1.80 |
| 300-01 | **176.47** | 5.52 | 71.70 | 179.22 | 5.34 | 29.90 | 187.69 | 7.61 | 11.60 | 310.45 | 8.49 | 3.30 |
| 300-02 | 396.17 | 14.35 | 76.50 | **393.13** | 11.51 | 37.30 | 407.53 | 13.72 | 12.10 | 514.20 | 15.62 | 3.40 |
| 300-03 | **195.71** | 8.66 | 68.30 | 201.03 | 7.72 | 30.10 | 205.18 | 11.72 | 10.90 | 330.03 | 8.93 | 3.20 |
| 300-04 | **401.58** | 13.42 | 87.40 | 404.67 | 11.65 | 35.90 | 413.98 | 11.35 | 13.60 | 558.36 | 14.65 | 3.20 |
| 300-05 | **170.27** | 4.46 | 78.30 | 171.35 | 5.46 | 32.20 | 184.58 | 6.14 | 12.50 | 320.93 | 9.60 | 3.20 |
| 300-06 | **236.89** | 8.74 | 75.60 | 240.51 | 7.67 | 33.70 | 247.90 | 9.16 | 11.60 | 355.87 | 10.78 | 3.30 |
| 300-07 | **150.40** | 5.47 | 72.80 | 153.44 | 6.05 | 31.70 | 159.95 | 7.21 | 11.40 | 271.72 | 8.14 | 3.20 |
| 300-08 | **161.03** | 4.67 | 71.10 | 162.06 | 5.79 | 30.50 | 169.53 | 6.44 | 11.20 | 285.08 | 10.47 | 3.20 |
| 300-09 | **134.52** | 5.33 | 70.30 | 135.98 | 5.63 | 29.40 | 138.81 | 4.09 | 10.80 | 236.66 | 8.71 | 3.20 |
| 300-10 | **173.58** | 6.19 | 67.20 | 176.63 | 6.40 | 30.20 | 183.25 | 7.49 | 10.50 | 288.54 | 8.32 | 3.10 |

Table 9.4: Average objective values $obj$ of 30 runs, corresponding standard deviations $\sigma(obj)$ as well as median execution times for a local search with $\mathcal{N}_{2,d}^{S1}$ with different filters enabled

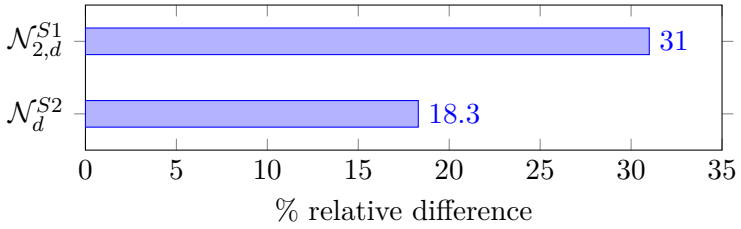


Figure 9.4: Relative differences between the average objective value of TWCH and the average objective values of a local search with $\mathcal{N}_{2,d}^{S1}$ and $\mathcal{N}_{d}^{S2}$. The values are calculated as $100\% \cdot (obj_T - obj_L)/obj_T$, where $obj_T$ denotes the average objective value of TWCH and $obj_{\mathrm{L}}$ represents the average objective value of one of the local search methods.

cause two of these DTs to adjoin each other, forcing the beam to be idle between the DTs. In contrast, there are many ways to exchange two DTs in this example without inducing idle time for the beam, such as exchanging two arbitrary DTs which are assigned to the same room. Nevertheless, $\mathcal{N}_{d}^{S2}$ is useful as part of a VND because it generates schedules which cannot be found in the neighbourhoods of $\mathcal{N}_{2,d}^{S1}$.

### 9.1.2 Inter-Day Neighbourhood Structures

This section shows the experimental results of a local search with inter-day neighbourhood structures. The inter-day neighbourhood structures change only the $Z_{t,u}$ variables of a solution and ignore the $S_{t,u}$ variables. This is done in order to improve the efficiency of the local search by avoiding to recalculate the $S_{t,u}$ variables using Algorithm 5.1. The objective of the local search is to minimize

$$\gamma^{\text{ext}} \sum_{r \in R} \sum_{d \in D'} \eta_{r,d} + \gamma^{\text{finish}} \sum_{t \in T} (Z_{t,\tau_t} - Z_{t,\tau_t}^{\text{earliest}}), \tag{1}$$

where $\eta_{r,d} = \max(\{S_{t,u} + P_{t,u,r}^{\text{end}} - W_{r,d}^{\text{end}} \mid t \in T, u \in U_t, r \in Q_{t,u}, Z_{t,u} = d\} \cup \{0\})$ is the used time of the extended availability period of a resource $r$ on day $d$ (see [MRR17]). All other symbols in this formula are defined in Chapter 4. Since determining (1) requires the exact starting times, the usage of the resources' availability periods for a given candidate set of DTs has to be estimated. To that end, the local search uses a modified version of (1) that replaces $\eta_{r,d}$ with a surrogate $\hat{\eta}_{r,d} = \max(0, \hat{\lambda}_{r,d} - h_{r,d})$, where $\hat{\lambda}_{r,d}$ estimates the required time and $h_{r,d}$ denotes the aggregated regular availability of resource $r$ on day $d$. We calculate $\hat{\lambda}_{r,d}$ as in [MRR17], where the beam idle times, which occur e.g. when two consecutive DTs require the same room, are taken into account.

Table 9.6 compares the results of three versions of local search with different inter-day neighbourhood structures. Each row contains average figures for a particular group of benchmark instances. Figure 9.5 shows the relative difference between the average objective values of these local search methods and the average objective values of TWCH for different numbers of therapies using the objective function (1). It is not obvious but

| Instance | $\mathcal{N}_{2,d}^{S1}$ | | | | $\mathcal{N}_{d}^{S2}$ | | | |
|---|---|---|---|---|---|---|---|---|
| | *obj* | $\sigma(obj)$ | *Moves* | *t [s]* | *obj* | $\sigma(obj)$ | *Moves* | *t [s]* |
| 100-01 | **107.73** | 5.36 | 882.50 | 19.40 | 128.31 | 7.93 | 478.50 | 27.60 |
| 100-02 | **151.08** | 8.15 | 963.50 | 21.50 | 174.49 | 10.13 | 575.00 | 30.80 |
| 100-03 | **68.57** | 2.61 | 620.00 | 16.90 | 82.89 | 3.55 | 357.00 | 22.20 |
| 100-04 | **127.49** | 5.69 | 798.00 | 19.90 | 146.65 | 6.93 | 456.50 | 27.70 |
| 100-05 | **123.28** | 6.13 | 799.50 | 19.70 | 137.33 | 7.61 | 474.00 | 29.90 |
| 100-06 | **94.04** | 5.16 | 726.50 | 20.40 | 111.19 | 5.42 | 438.50 | 30.30 |
| 100-07 | **92.49** | 4.79 | 863.50 | 20.40 | 114.58 | 4.29 | 422.00 | 27.50 |
| 100-08 | **138.69** | 4.47 | 860.00 | 19.90 | 156.62 | 6.19 | 486.50 | 27.00 |
| 100-09 | **58.48** | 5.22 | 771.50 | 18.60 | 77.49 | 4.28 | 442.50 | 25.80 |
| 100-10 | **124.74** | 4.68 | 764.50 | 19.50 | 138.57 | 5.61 | 415.00 | 27.20 |
| 150-01 | **163.59** | 7.49 | 1357.00 | 37.90 | 194.86 | 7.53 | 717.50 | 56.90 |
| 150-02 | **287.47** | 5.97 | 1494.00 | 40.30 | 317.66 | 7.71 | 880.00 | 64.10 |
| 150-03 | **215.73** | 6.80 | 1182.00 | 37.10 | 243.07 | 5.85 | 621.00 | 57.70 |
| 150-04 | **144.92** | 5.31 | 1323.00 | 38.00 | 175.07 | 6.76 | 825.00 | 58.40 |
| 150-05 | **125.83** | 5.27 | 1249.50 | 30.60 | 151.81 | 6.53 | 744.50 | 51.10 |
| 150-06 | **314.04** | 5.80 | 1364.00 | 36.80 | 343.26 | 7.85 | 906.00 | 61.50 |
| 150-07 | **279.07** | 6.34 | 1238.50 | 35.00 | 300.06 | 7.85 | 807.00 | 58.10 |
| 150-08 | **273.61** | 6.33 | 1349.00 | 43.40 | 308.62 | 6.18 | 639.00 | 60.50 |
| 150-09 | **242.49** | 6.55 | 1366.00 | 40.30 | 270.53 | 8.04 | 740.00 | 60.90 |
| 150-10 | **170.27** | 6.12 | 1418.00 | 39.10 | 206.76 | 6.66 | 721.50 | 57.90 |
| 200-01 | **288.65** | 8.84 | 1585.00 | 45.20 | 323.32 | 8.83 | 926.50 | 76.80 |
| 200-02 | **286.44** | 3.67 | 1718.00 | 47.10 | 316.31 | 3.59 | 1028.00 | 87.10 |
| 200-03 | **203.00** | 4.70 | 1648.00 | 47.50 | 245.85 | 5.31 | 775.50 | 76.30 |
| 200-04 | **240.05** | 5.95 | 2077.50 | 50.30 | 291.32 | 7.48 | 1081.50 | 90.00 |
| 200-05 | **219.95** | 4.52 | 2026.50 | 48.20 | 261.31 | 8.09 | 1133.50 | 89.60 |
| 200-06 | **202.48** | 3.52 | 1768.50 | 56.20 | 248.57 | 5.02 | 788.50 | 86.50 |
| 200-07 | **141.00** | 4.19 | 1794.00 | 47.60 | 176.18 | 8.92 | 953.50 | 78.90 |
| 200-08 | **142.65** | 5.62 | 1759.00 | 44.00 | 175.44 | 6.93 | 1089.50 | 77.00 |
| 200-09 | **150.38** | 5.87 | 1641.00 | 40.10 | 181.04 | 7.04 | 1019.50 | 77.10 |
| 200-10 | **400.52** | 7.63 | 1699.00 | 48.50 | 437.38 | 8.61 | 1029.50 | 92.70 |
| 300-01 | **176.47** | 5.52 | 2892.00 | 71.70 | 227.41 | 7.08 | 2024.00 | 165.40 |
| 300-02 | **396.17** | 14.35 | 2796.50 | 76.50 | 458.37 | 10.95 | 1606.50 | 155.50 |
| 300-03 | **195.71** | 8.66 | 3103.50 | 68.30 | 251.73 | 7.69 | 1979.00 | 152.20 |
| 300-04 | **401.58** | 13.42 | 2998.50 | 87.40 | 479.51 | 14.32 | 1619.00 | 165.20 |
| 300-05 | **170.27** | 4.46 | 3108.00 | 78.30 | 230.72 | 7.34 | 1968.00 | 162.20 |
| 300-06 | **236.89** | 8.74 | 2868.00 | 75.60 | 296.43 | 9.14 | 1632.50 | 154.90 |
| 300-07 | **150.40** | 5.47 | 3076.00 | 72.80 | 205.00 | 6.93 | 1924.50 | 157.00 |
| 300-08 | **161.03** | 4.67 | 2985.50 | 71.10 | 219.37 | 7.47 | 1877.00 | 154.20 |
| 300-09 | **134.52** | 5.33 | 2870.50 | 70.30 | 175.94 | 4.46 | 1885.00 | 157.80 |
| 300-10 | **173.58** | 6.19 | 2793.00 | 67.20 | 226.35 | 8.66 | 1748.50 | 144.50 |



Table 9.5: Average objective values *obj* of 30 runs, corresponding standard deviations $\sigma(obj)$ as well as median number of moves and median times for $\mathcal{N}_{2,d}^{S1}$ and $\mathcal{N}_{d}^{S2}$ without a filter

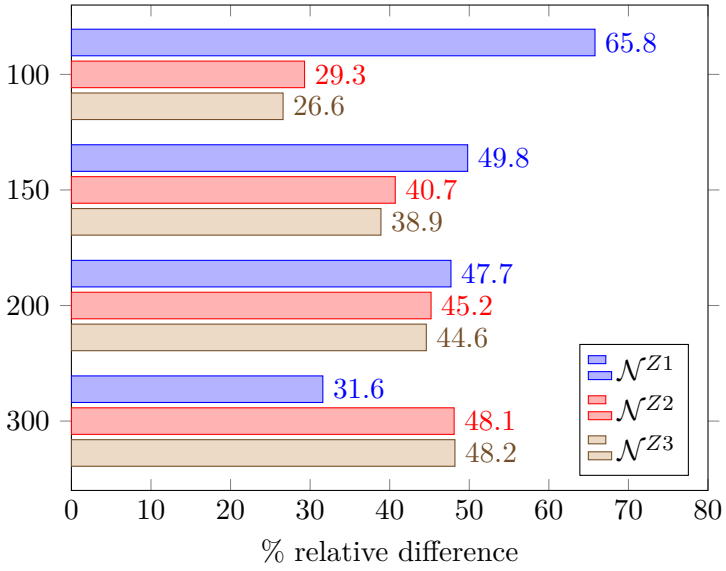


Figure 9.5: Relative differences between the average objective value of TWCH and the average objective values of a local search with $\mathcal{N}^{Z1}$, $\mathcal{N}^{Z2}$ and $\mathcal{N}^{Z3}$ for different numbers of therapies using the objective function (1). The values are calculated as $100\% \cdot (obj_T - obj_L)/obj_T$, where $obj_T$ denotes the average objective value of TWCH and $obj_L$ represents the average objective value of one of the local search methods.

a Wilcoxon rank sum test with a significance level of 95% indicates that $\mathcal{N}^{Z2}$ generates significantly better solutions for half of the instances, while $\mathcal{N}^{Z3}$ performs better on only 8 instances. The local search that uses $\mathcal{N}^{Z3}$ performs better than $\mathcal{N}^{Z2}$ especially on the largest instances with 300 therapies because for 5 of them it finds significantly better solutions, while $\mathcal{N}^{Z2}$ produces better solutions only for two of them. However, the initial solutions have only improved by a small percentage of 1% – 5% because TWCH is quite good at distributing the therapies over the whole planning horizon.

A Wilcoxon rank sum test with a significance level of 95% indicates that a local search that uses $\mathcal{N}^{Z1}$ generates significantly better solutions for most instances below 300 therapies than a local search using one of the other two inter-day neighbourhood structures. This is intuitively clear because $\mathcal{N}^{Z1}$ can stretch therapies, which is sometimes needed to reduce extended time on certain days. Interestingly, a local search which utilizes one of the other two neighbourhood structures performs statistically better on most large instances with 300 therapies as well as on some instances with 150 and 200 therapies. This result indicates that TWCH produces dense schedules which already contain stretched therapies for large instances. On such initial solutions, $\mathcal{N}^{Z2}$ and $\mathcal{N}^{Z3}$ would preserve the distances between two consecutive DTs. In contrast, $\mathcal{N}^{Z1}$ would either stretch or compress such therapies, which may increase the therapy duration or the extended time on some working days.

| Instance | $\mathcal{N}^{Z1}$ | | | $\mathcal{N}^{Z2}$ | | | $\mathcal{N}^{Z3}$ | | |
|---|---|---|---|---|---|---|---|---|---|
| | *obj* | $\sigma(obj)$ | *t [s]* | *obj* | $\sigma(obj)$ | *t [s]* | *obj* | $\sigma(obj)$ | *t [s]* |
| 100-01 | **6.43** | 0.17 | 4.20 | 14.22 | 0.78 | 1.60 | 14.79 | 0.49 | 1.50 |
| 100-02 | **8.21** | 1.11 | 6.00 | 21.36 | 0.44 | 1.40 | 21.62 | 0.31 | 1.20 |
| 100-03 | **5.53** | 0.11 | 3.30 | 7.26 | 0.38 | 1.00 | 7.38 | 0.40 | 1.10 |
| 100-04 | **8.43** | 0.56 | 7.40 | 16.40 | 0.36 | 1.60 | 16.27 | 0.30 | 1.70 |
| 100-05 | **7.02** | 0.25 | 5.10 | 12.96 | 0.48 | 1.60 | 13.13 | 0.43 | 1.50 |
| 100-06 | **6.62** | 0.10 | 4.30 | 11.53 | 0.50 | 1.70 | 12.12 | 0.22 | 1.30 |
| 100-07 | **6.01** | 0.21 | 4.70 | 13.67 | 0.80 | 1.20 | 13.56 | 0.69 | 1.20 |
| 100-08 | **7.81** | 0.95 | 6.80 | 23.60 | 0.79 | 1.60 | 26.56 | 0.52 | 1.30 |
| 100-09 | **5.98** | 0.24 | 3.70 | 6.45 | 0.25 | 1.10 | 6.44 | 0.21 | 1.10 |
| 100-10 | **6.97** | 0.13 | 5.40 | 15.14 | 0.44 | 1.80 | 16.06 | 0.28 | 1.30 |
| 150-01 | **13.80** | 1.20 | 20.60 | 16.43 | 0.69 | 6.00 | 16.21 | 0.42 | 5.30 |
| 150-02 | **18.23** | 1.26 | 22.60 | 33.16 | 0.81 | 6.10 | 34.84 | 0.81 | 5.50 |
| 150-03 | 31.19 | 1.57 | 19.40 | **29.53** | 0.46 | 5.20 | 30.25 | 0.41 | 5.00 |
| 150-04 | **12.16** | 0.23 | 13.60 | 13.32 | 0.36 | 5.10 | 13.65 | 0.28 | 5.10 |
| 150-05 | 10.79 | 0.17 | 11.10 | **8.85** | 0.25 | 4.10 | 9.20 | 0.31 | 3.90 |
| 150-06 | **47.62** | 2.80 | 26.80 | 58.66 | 0.88 | 6.50 | 60.82 | 0.66 | 5.20 |
| 150-07 | **41.21** | 1.51 | 23.80 | 44.89 | 0.71 | 6.20 | 45.98 | 0.51 | 5.10 |
| 150-08 | **32.91** | 1.99 | 23.70 | 36.55 | 0.67 | 6.00 | 38.05 | 0.70 | 5.60 |
| 150-09 | **30.12** | 3.14 | 24.90 | 38.37 | 0.69 | 5.10 | 39.85 | 0.77 | 5.10 |
| 150-10 | **12.99** | 0.51 | 21.50 | 16.72 | 0.47 | 6.00 | 16.51 | 0.34 | 5.20 |
| 200-01 | 27.36 | 2.21 | 70.70 | **21.42** | 0.57 | 14.20 | 21.80 | 0.62 | 13.00 |
| 200-02 | **25.78** | 1.30 | 66.90 | 35.69 | 0.64 | 11.20 | 35.66 | 0.60 | 13.90 |
| 200-03 | **18.33** | 0.15 | 29.30 | 21.15 | 0.56 | 14.10 | 21.23 | 0.46 | 15.60 |
| 200-04 | **18.16** | 1.85 | 62.70 | 21.55 | 0.72 | 14.00 | 22.13 | 0.81 | 16.00 |
| 200-05 | 15.31 | 0.28 | 47.80 | **14.29** | 0.40 | 11.00 | 14.63 | 0.44 | 14.90 |
| 200-06 | **19.09** | 0.58 | 60.50 | 26.25 | 0.65 | 14.00 | 27.17 | 0.56 | 13.70 |
| 200-07 | 15.14 | 0.20 | 40.10 | 15.37 | 0.51 | 10.80 | **15.14** | 0.36 | 12.60 |
| 200-08 | 13.18 | 0.25 | 28.50 | 12.33 | 0.36 | 10.70 | **11.97** | 0.28 | 10.60 |
| 200-09 | 14.40 | 0.21 | 35.70 | **12.57** | 0.27 | 9.90 | 12.65 | 0.32 | 11.70 |
| 200-10 | 57.02 | 2.41 | 60.80 | **53.89** | 0.76 | 16.30 | 54.87 | 0.84 | 13.20 |
| 300-01 | 13.84 | 0.33 | 100.30 | 12.23 | 0.29 | 29.40 | **12.22** | 0.27 | 31.70 |
| 300-02 | 49.69 | 4.52 | 265.40 | **31.28** | 1.11 | 49.00 | 32.16 | 1.00 | 53.00 |
| 300-03 | 21.51 | 0.58 | 123.30 | **15.70** | 0.46 | 39.30 | 15.96 | 0.40 | 39.60 |
| 300-04 | 36.34 | 3.16 | 234.10 | **34.54** | 1.84 | 43.90 | 34.88 | 2.06 | 48.60 |
| 300-05 | 13.80 | 0.30 | 98.30 | 12.01 | 0.30 | 31.80 | **11.90** | 0.29 | 31.40 |
| 300-06 | 26.27 | 0.42 | 142.50 | 17.67 | 0.56 | 44.90 | **17.54** | 0.37 | 47.10 |
| 300-07 | 14.03 | 0.33 | 114.00 | 10.82 | 0.27 | 31.70 | **10.32** | 0.24 | 32.60 |
| 300-08 | 14.07 | 0.58 | 105.30 | 10.75 | 0.33 | 38.70 | **10.48** | 0.32 | 32.60 |
| 300-09 | 13.63 | 0.51 | 107.90 | 10.69 | 0.24 | 31.90 | **10.22** | 0.19 | 36.10 |
| 300-10 | 20.71 | 0.55 | 137.20 | 14.24 | 0.38 | 37.10 | **13.96** | 0.26 | 31.90 |



Table 9.6: Average objective values *obj* of 30 runs, corresponding standard deviations $\sigma(obj)$ as well as median execution times for $\mathcal{N}^{Z1}$, $\mathcal{N}^{Z2}$ and $\mathcal{N}^{Z3}$

| Instance size | $IG_1$ | | $IG_2$ | $IG_3$ |
|---|---|---|---|---|
| | $\beta$ | $n^{\text{rta}-\text{noimp}}$ | $\beta$ | $\beta$ |
| 100 | 0.106 | $\lfloor 0.787 \cdot \lvert G_d \rvert \rfloor$ | 0.059 | 0.029 |
| 150 | 0.093 | $\lfloor 0.705 \cdot \lvert G_d \rvert \rfloor$ | 0.028 | 0.016 |
| 200 | 0.053 | $\lfloor 1.210 \cdot \lvert G_d \rvert \rfloor$ | 0.029 | 0.012 |
| 300 | 0.116 | $\lfloor 1.761 \cdot \lvert G_d \rvert \rfloor$ | 0.015 | 0.007 |

Table 9.7: Parameter settings for $IG_1$, $IG_2$ and $IG_3$ determined by irace. The parameters include the destruction rate $\beta$ as well as $n^{\text{rta}-\text{noimp}}$ determining the number of iterations the randomized construction heuristic is executed each day with no improvement.

## 9.2 2-Phase Approach and Iterated Greedy

In this section we perform an experimental evaluation and comparison of the 2PA from Chapter 6 with the IG approach from Chapter 7, and the simpler IG method from [MRSR16]. The IG method from [MRSR16] will be called $IG_1$ from here on, the IG approach and its improved version from Chapter 7 will be denoted as $IG_2$ and $IG_3$, respectively, in the remaining thesis. The IG method from [MRSR16] ($IG_1$) differs from $IG_2$ in two aspects. First, it uses a randomized construction heuristic instead of a more targeted local search method to improve the time assignments on individual days. Secondly, its destruction and construction phase does not preserve the time assignments of the therapies which were not removed from the schedule. All experiments in this section were conducted with a time limit of 20 minutes because the IG approaches would not detect whether a local optimum has been reached.

The strategy parameters of the metaheuristics were tuned using the automatic parameter configuration tool irace in version 2.3. In detail, irace was applied separately on each instance size to tune the destruction rate $\beta$ and $n^{\text{rta}-\text{noimp}}$ for $IG_1$ and $IG_2$, where $n^{\text{rta}-\text{noimp}}$ determines the number of iterations the randomized construction heuristic is executed each day with no improvement. On this account, we generated for each instance size five independent instances for tuning. Moreover, each irace run had a computational budget of 1000 experiments. The resulting parameter configurations are shown in Table 9.7.

The missing pieces in the definition of the 2PA and $IG_3$ are the choice of neighbourhood structures to be used in the VNDs and the order in which they are applied as well as the value for $\beta$. Since we did not know with certainty which neighbourhood structures perform best in this context, we decided to apply the Racing method in two consecutive steps to find well performing sequences of neighbourhood structures for the VNDs in 2PA and $IG_3$ as well as a suitable value for $\beta$.

The first step was dedicated to finding a sequence of neighbourhood structures for the inter-day and intra-day VNDs in both metaheuristics. To that end, we defined 15 VNDs which are composed of all possible sequences of up to 3 different inter-day neighbourhood

structures. We also defined 77 VNDs of different sequences of intra-day neighbourhood structures with different filter combinations being enabled. We then applied irace once on 2PA and $IG_3$ to find for each of them an inter-day and intra-day VND which produces better or equally good solutions than all the other VNDs on the tuning instances. Each irace run had a computational budget of 10000 experiments. The resulting VNDs are $(\mathcal{N}^{Z1}, \mathcal{N}^{Z3})$ and $(\mathcal{N}^{S1}_{2,D'}, \mathcal{N}^{S2}_{D'})$ for 2PA, where both the *gap* and *NST* filter are used in a conjunction. For $IG_3$ we found the inter-day VND $(\mathcal{N}^{Z3}, \mathcal{N}^{Z1})$, where the *adjacent* and *NST* filter are used in a disjunction, as well as the intra-day VND $(\mathcal{N}^{S1}_{2,D'}, \mathcal{N}^{S2}_{D'})$, where the *gap* and *NST* filter are applied disjunctively.

Afterwards we fixed the neighbourhood structures for both VNDs in $IG_3$, and used irace with a budget of 10000 to tune the destruction rate $\beta$. Since we expected that the optimal destruction rate $\beta$ depends on the number of therapies to be scheduled, we executed irace separately on each instance size. The resulting destruction rate is shown in Table 9.7.

Table 9.8 depicts for $IG_1$, 2PA, $IG_2$ and $IG_3$ averages of the final objective values *obj* and the corresponding standard deviation $\sigma(obj)$ over 30 runs for each of the 40 benchmark instances. Note that at the end of each run the LP model from Chapter 8 is solved in order to finetune the DTs' starting times. We start by comparing $IG_1$ with $IG_2$. According to a Wilcoxon rank sum test with a significance level of 95%, $IG_2$ is significantly better than $IG_1$ on all 40 instances. On 33 instances the average objective values of $IG_2$ are smaller by more than 25% compared to those of $IG_1$, and on 22 instances the average objective values of $IG_2$ are even halved compared to the ones from $IG_1$. The main reason for this performance improvement is the interplay between the construction phase of $IG_2$ and its local search procedure. On the one hand, the local search operator is, in general, able to provide better results than the randomized construction heuristic of $IG_1$. 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 of $IG_2$ 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 w.r.t. the $\mathcal{N}^{S1}_{2,D'}$ neighbourhood allows to reduce the time spent in the local search procedure and, consequently, increases the total number of iterations.

When comparing 2PA with $IG_1$, a Wilcoxon rank sum test with a signifance level of 95% indicates that 2PA is better on 39 out of 40 instances. However, $IG_2$ produces significantly better results than 2PA on 32 out of 40 instances, according to a Wilcoxon rank sum test. With regard to the final objective values, 2PA produces on average about 25% better solutions. The relative difference between the final objective values of $IG_2$ and 2PA is the same.

Most importantly, however, Table 9.8 clearly shows that $IG_3$ dominates the other metaheuristics. A Wilcoxon rank sum test with a significance level of 95% indicates that $IG_3$ produces significantly better solutions than $IG_2$ on all 40 instances. Metaheuristic $IG_3$ achieved the most improvement compared to $IG_2$ on instance 150-01 with the average

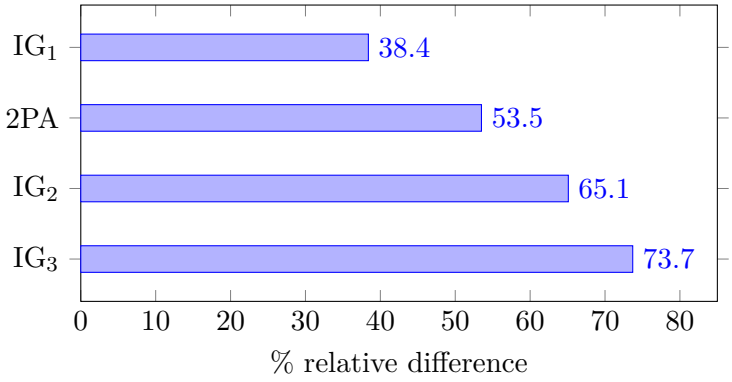


Figure 9.6: Relative differences between the average objective value of TWCH and the average objective values of $IG_1$, 2PA, $IG_2$ and $IG_3$. The values are calculated as $100\% \cdot (obj_T - obj_L)/obj_T$, where $obj_T$ denotes the average objective value of TWCH and $obj_L$ represents the average objective value of one of the metaheuristics.

objective value being smaller by 44% compared to the one of $IG_2$. On average the objective values of $IG_3$ are smaller by 25% compared to those of $IG_2$. The reason for $IG_3$ performing better than $IG_2$ is that it includes a better local search component, which optimizes both the day and time assignments after the construction phase. As can be seen in Figure 9.6, $IG_3$ improves the initial solutions of TWCH on average by 74%, which is about twice as much as the improvement of the reference method $IG_1$ from [MRSR16].

| Instance | $IG_1$ | | 2PA | | $IG_2$ | | $IG_3$ | |
|---|---|---|---|---|---|---|---|---|
| | *obj* | $\sigma(obj)$ | *obj* | $\sigma(obj)$ | *obj* | $\sigma(obj)$ | *obj* | $\sigma(obj)$ |
| 100-01 | 85.24 | 3.56 | 67.12 | 4.45 | 34.45 | 4.00 | **19.94** | 1.92 |
| 100-02 | 111.52 | 4.39 | 86.29 | 6.98 | 41.94 | 3.78 | **30.54** | 2.92 |
| 100-03 | 57.62 | 4.41 | 42.43 | 2.42 | 24.87 | 1.97 | **17.65** | 1.89 |
| 100-04 | 89.99 | 3.86 | 62.83 | 6.07 | 37.77 | 3.48 | **25.30** | 2.49 |
| 100-05 | 85.51 | 3.93 | 75.05 | 4.39 | 29.24 | 2.82 | **19.34** | 1.63 |
| 100-06 | 64.37 | 4.58 | 56.88 | 3.89 | 40.52 | 4.37 | **30.23** | 5.97 |
| 100-07 | 72.30 | 4.41 | 55.79 | 4.51 | 27.68 | 2.71 | **16.91** | 1.43 |
| 100-08 | 97.94 | 4.56 | 81.35 | 4.44 | 46.62 | 3.14 | **34.52** | 3.06 |
| 100-09 | 59.63 | 3.42 | 38.28 | 2.76 | 27.21 | 2.63 | **19.57** | 1.63 |
| 100-10 | 89.12 | 6.50 | 71.59 | 4.40 | 37.06 | 5.04 | **22.34** | 3.70 |
| 150-01 | 129.07 | 4.28 | 94.66 | 5.03 | 74.76 | 6.75 | **41.69** | 3.07 |
| 150-02 | 258.15 | 9.13 | 171.44 | 7.04 | 151.40 | 7.18 | **119.70** | 7.62 |
| 150-03 | 133.99 | 4.77 | 111.96 | 8.29 | 113.05 | 6.38 | **71.38** | 4.71 |
| 150-04 | 117.56 | 5.67 | 83.43 | 5.42 | 58.62 | 4.45 | **42.09** | 3.26 |
| 150-05 | 95.60 | 5.33 | 74.50 | 3.56 | 40.78 | 3.07 | **34.01** | 3.16 |
| 150-06 | 239.60 | 7.84 | 175.64 | 9.70 | 185.60 | 13.69 | **124.71** | 5.93 |
| 150-07 | 175.74 | 6.63 | 141.08 | 8.77 | 167.40 | 9.73 | **110.96** | 4.15 |
| 150-08 | 195.93 | 6.80 | 140.14 | 6.42 | 149.43 | 7.48 | **108.02** | 4.86 |
| 150-09 | 171.44 | 4.70 | 122.95 | 4.85 | 134.00 | 9.30 | **84.78** | 4.84 |
| 150-10 | 131.03 | 5.27 | 93.14 | 5.54 | 74.56 | 7.80 | **50.14** | 4.11 |
| 200-01 | 203.37 | 10.43 | 151.73 | 10.89 | 144.15 | 10.25 | **108.79** | 7.59 |
| 200-02 | 239.71 | 7.42 | 153.86 | 5.61 | 160.54 | 8.65 | **121.12** | 5.62 |
| 200-03 | 175.81 | 8.20 | 121.49 | 4.66 | 109.02 | 6.38 | **83.86** | 6.11 |
| 200-04 | 217.24 | 8.56 | 135.32 | 5.05 | 132.00 | 9.77 | **85.01** | 7.24 |
| 200-05 | 189.73 | 8.15 | 123.96 | 5.81 | 90.94 | 6.74 | **73.95** | 5.22 |
| 200-06 | 180.41 | 6.48 | 117.59 | 5.30 | 126.66 | 4.89 | **96.98** | 7.16 |
| 200-07 | 150.84 | 5.89 | 90.84 | 6.22 | 72.37 | 3.90 | **64.90** | 5.76 |
| 200-08 | 141.07 | 5.85 | 78.87 | 4.34 | 67.12 | 4.57 | **43.47** | 2.55 |
| 200-09 | 138.75 | 6.21 | 86.92 | 4.91 | 56.24 | 4.31 | **44.30** | 4.42 |
| 200-10 | 293.31 | 10.95 | 201.80 | 6.27 | 233.73 | 10.49 | **174.41** | 5.59 |
| 300-01 | 240.50 | 5.86 | 162.01 | 40.05 | 90.19 | 5.82 | **81.54** | 8.39 |
| 300-02 | 354.38 | 13.83 | 319.68 | 27.75 | 281.46 | 14.38 | **192.28** | 22.10 |
| 300-03 | 253.70 | 7.03 | 213.00 | 16.94 | 102.90 | 6.36 | **92.82** | 7.57 |
| 300-04 | 375.77 | 9.32 | 384.16 | 17.68 | 274.34 | 12.26 | **256.60** | 26.95 |
| 300-05 | 245.85 | 6.16 | 162.22 | 38.72 | 78.12 | 3.68 | **70.39** | 5.23 |
| 300-06 | 245.23 | 8.87 | 232.23 | 12.88 | 127.58 | 9.77 | **99.76** | 10.97 |
| 300-07 | 208.32 | 6.46 | 166.36 | 26.97 | 70.57 | 4.03 | **61.16** | 3.71 |
| 300-08 | 216.85 | 5.11 | 166.94 | 22.81 | 77.20 | 4.44 | **68.38** | 4.10 |
| 300-09 | 196.83 | 4.34 | 140.08 | 22.64 | 66.65 | 3.59 | **59.19** | 3.30 |
| 300-10 | 214.59 | 7.39 | 183.60 | 10.04 | 76.90 | 4.35 | **68.14** | 6.41 |



Table 9.8: Average objective values *obj* of 30 runs and corresponding standard deviations $\sigma(obj)$ for $IG_1$, 2PA, $IG_2$, $IG_3$ with enabled postprocessing step

CHAPTER 10

# Conclusion

This thesis considers the Particle Therapy Patient Scheduling Problem (PTPSP) in which cancer therapies consisting of sequences of treatments have to be planned within a planning horizon of several months. We presented five neighbourhood structures to be used by local search methods for the PTPSP. We distinguished between intra-day and inter-day neighbourhood structures. Two intra-day neighbourhood structures operate on a sequence of daily treatments (DTs) on a certain day resulting from sorting the DTs by the times from which on they use the beam. They change the order of the DTs by either moving a DT to a different position in this sequence or by permuting a subset of DTs. On instances occuring in practice, these neighbourhoods are too large to be explored in a reasonable time. Therefore, we defined three filters which reduce the neighbourhoods to considerably smaller subsets of promising solutions. The *adjacent* filter causes the DTs' starting times to change only slightly by ensuring that only consecutive DTs change their places. We showed that a local search using this filter converges exceptionally fast but provides almost no improvement. The *gap* filter rejects all DT permutations that increase the number of times an irradiation room is used in a row. A local search using this filter proved to produce nearly as good solutions as a local search without a filter while converging about twice as fast. The *NST* filter ensures that a move does not increase the aggregated deviation of the DTs' starting times from the corresponding nominal starting times. A local search which utilizes this filter converges about three times as fast as a local search using the *gap* filter while producing only slightly worse solutions. The inter-day neighbourhood structures move DTs to different days without considering the exact starting times of the DTs. One of them moves subsets of the DTs of a therapy, whereas the other two move whole therapies. The first inter-day neighbourhood structure has been shown to be more effective on sparse schedules of smaller instances, where it is often possible to move a DT by one day without violating a constraint. In contrast, on large instances the other inter-day neighbourhood structures are superior because they do not try to stretch or compress therapies on a dense schedule where the therapies are

already well interleaved.

We further presented two metaheuristics for the PTPSP. The first metaheuristic, which we called 2-Phase Approach (2PA), combines two Variable Neighborhood Descents (VNDs) which act on different levels. The first VND uses inter-day neighbourhood structures to optimize the day assignments of the DTs. The second VND consists of intra-day neighbourhood structures which optimize the starting times of the DTs without changing their day assignment. We showed that 2PA produces significantly better solutions than the Iterated Greedy (IG) metaheuristic from a previous work [MRSR16] of Maschler et al. on 39 out of 40 instances. The solutions are, on average, 25% better. The 2-Phase Approach performs better than the IG approach because its inter-day neighbourhood structures find promising day assignments in a very targeted and less random way than IG's destruction and construction phase.

The second metaheuristic is an IG method that enhances the IG from [MRSR16] of Maschler et al. In contrast to the latter, the presented approach aims at preserving the order of the non-removed DTs on the individual days. The resulting advantage is that more information from the incumbent solution is maintained. Compared to the previous IG our new IG metaheuristic provides significantly better results on all 40 benchmark instances. The superiority of the enhanced approach over the previous IG can be explained by the interplay between its construction phase and the applied local search technique: The local search method yields, in general, better results than applying the randomized time assignment of the therapy-wise construction heuristic (TWCH). However, due to the required encoding and decoding steps, evaluating neighbours is time consuming. Hence, to ensure that the metaheuristic is able to perform sufficiently many iterations it is important that the neighbourhood 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 that iteratively places the removed DTs into the permutation resulting from sorting the DTs according to the times at which they use the beam. Compared to 2PA the generated solutions are, on average, 25% better. We further improved our IG approach by incorporating a VND in the construction and local search phase. This improved IG generates significantly better solutions than all other approaches on all instances. It produces solutions, which are, on average, again 25% better than the solutions of IG and approximately 74% better than the initial solutions.

Our study can be continued in several directions. One way is to improve the presented local search methods. For example, our problem instances have the property that all therapies have to start on Monday or Tuesday. Furthermore, some resources are available only in the morning. These two properties can be exploited by appropriate filters for the inter-day and intra-day neighbourhood structures, respectively, which could improve the performance of the local search. It can also be observed that the presented local search methods work best for dense schedules, where it is not necessary to introduce large lags between two consecutive DTs on a day. That is because the LP model of the postprocessing step is not able to move a greedily assigned DT across an unavailability period to a later time. This problem could perhaps be solved by defining a decoder that

can schedule a DT behind an unavailability period if this leads to a better objective value. Another way to continue our study is to generalize the PTPSP formalization in such a way that a DT consists of individual activities with separate time assignments. The neighbourhood structures could then operate on the activity level instead of the DT level.

# List of Tables

# List of Algorithms

# Bibliography


[AKM07] E Aarts, J Korst, and W Michiels. Theoretical aspects of local search. *Monographs in Theoretical Computer Science. An EATCS Series. Springer, New York*, 2007.

[BBS07] Prasanna Balaprakash, Mauro Birattari, and Thomas Stützle. Improvement strategies for the f-race algorithm: Sampling design and iterative refinement. In *International workshop on hybrid metaheuristics*, pages 108–122. Springer, 2007.

[BLRP11] Edmund K Burke, Pedro Leite-Rocha, and Sanja Petrovic. An integer linear programming model for the radiotherapy treatment scheduling problem. *arXiv preprint arXiv:1103.3391*, 2011.

[BSPV02] Mauro Birattari, Thomas Stützle, Luis Paquete, and Klaus Varrentrapp. A racing algorithm for configuring metaheuristics. In *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*, pages 11–18. Morgan Kaufmann Publishers Inc., 2002.

[CCZ14] Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. *Integer programming*, volume 271. Springer, 2014.

[Dan16] George Dantzig. *Linear programming and extensions*. Princeton university press, 2016.

[GP10] Michel Gendreau and Jean-Yves Potvin. *Handbook of metaheuristics*, volume 2. Springer, 2010.

[HM06] Pierre Hansen and Nenad Mladenović. First vs. best improvement: An empirical study. *Discrete Applied Mathematics*, 154(5):802–817, 2006.

[KP09] Truword Kapamara and Dobrila Petrovic. A heuristics and steepest hill climbing method to scheduling radiotherapy patients. In *Proceedings of the 35th International Conference on Operational Research Applied to Health Services (ORAHS)*, pages 12–17. Citeseer, 2009.

[KSH+06] T Kapamara, K Sheibani, OCL Haas, CR Reeves, and D Petrovic. A review of scheduling problems in radiotherapy. In *Proceedings of the Eighteenth International Conference on Systems Engineering (ICSE2006), Coventry University, UK*, pages 201–207, 2006.

[Lar93] SN Larsson. Radiotherapy patient scheduling using a desktop personal computer. *Clinical Oncology*, 5(2):98–101, 1993.

[LFLR15] Antoine Legrain, Marie-Andrée Fortin, Nadia Lahrichi, and Louis-Martin Rousseau. Online stochastic optimization of radiotherapy patient scheduling. *Health care management science*, 18(2):110–123, 2015.

[MHRR17] Johannes Maschler, Thomas Hackl, Martin Riedler, and Günther R Raidl. An enhanced iterated greedy metaheuristic for the particle therapy patient scheduling problem. In *Proceedings of the 12th Metaheuristics International Conference. Barcelona, Spain*, 2017.

[Mit98] Melanie Mitchell. *An introduction to genetic algorithms*. MIT press, 1998.

[MM97] Oded Maron and Andrew W Moore. The racing algorithm: Model selection for lazy learners. In *Lazy learning*, pages 193–225. Springer, 1997.

[MRR17] Johannes Maschler, Martin Riedler, and Günther R Raidl. Particle therapy patient scheduling: Time estimation for scheduling sets of treatments. In *International Conference on Computer Aided Systems Theory*, pages 364–372. Springer, 2017.

[MRSR16] Johannes Maschler, Martin Riedler, Markus Stock, and Günther R Raidl. Particle therapy patient scheduling: First heuristic approaches. In *Proceedings of the 11th Int. Conference on the Practice and Theory of Automated Timetabling (to appear 2016)*, 2016.

[NEH83] Muhammad Nawaz, E Emory Enscore, and Inyong Ham. A heuristic algorithm for the $m$-machine, $n$-job flow-shop sequencing problem. *Omega*, 11(1):91–95, 1983.

[PLR08] Sanja Petrovic and Pedro Leite-Rocha. Constructive and grasp approaches to radiotherapy treatment scheduling. In *World Congress on Engineering and Computer Science 2008, WCECS'08. Advances in Electrical and Electronics Engineering-IAENG Special Edition of the*, pages 192–200. IEEE, 2008.

[PLSS06] Sanja Petrovic, William Leung, Xueyan Song, and Santhanam Sundar. Algorithms for radiotherapy treatment booking. In *Proceedings of the 25th Workshop of the UK Planning and Scheduling Special Interest Group (PlanSIG'2006), Nottingham, UK*, 2006.

[PMP09] Dobrila Petrovic, Mohammad Morshed, and Sanja Petrovic. Genetic algorithm based scheduling of radiotherapy treatments for cancer patients. In *Conference on Artificial Intelligence in Medicine in Europe*, pages 101–105. Springer, 2009.

[PMP11] Dobrila Petrovic, Mohammad Morshed, and Sanja Petrovic. Multi-objective genetic algorithms for scheduling of radiotherapy treatments for categorised cancer patients. *Expert Systems with Applications*, 38(6):6994–7002, 2011.

[PP16] Marco Pranzo and Dario Pacciarelli. An iterated greedy metaheuristic for the blocking job shop scheduling problem. *Journal of Heuristics*, 22(4):587–611, 2016.

[PR14] Quan-Ke Pan and Rubén Ruiz. An effective iterated greedy algorithm for the mixed no-idle permutation flowshop scheduling problem. *Omega*, 44:41–50, 2014.

[RS07] Rubén Ruiz and Thomas Stützle. A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research*, 177(3):2033–2049, 2007.

[Wil45] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6):80–83, 1945.

[WKW70] Frank Wilcoxon, SK Katti, and Roberta A Wilcox. Critical values and probability levels for the wilcoxon rank sum test and the wilcoxon signed rank test. *Selected tables in mathematical statistics*, 1:171–259, 1970.