TILBURG UNIVERSITY

Robust optimization of a radiotherapy pretreatment preparation workflow

Authors	Hoffmans-Holtzer,Nienke; Smolenaers,Luuk; Olofsen-van Acht,Manouk; Hoogeman,Mischa et al
Published in	Physics in Medicine and Biology
DOI	10.1088/1361-6560/acf437
Publication Date	2024-01-21
Document Version	publishersversion
Link	https://research.tilburguniversity.edu/en/publications/ffa94496-9db1-44cb-aa0e-2ded6bb00a05
Citation	Hoffmans-Holtzer, N, Smolenaers, L, Olofsen-van Acht, M, Hoogeman, M, Balvert, M & Petit, S 2024, 'Robust optimization of a radiotherapy pretreatment preparation workflow', Physics in Medicine and Biology, vol. 69, no. 2, 025022. https://doi.org/10.1088/1361-6560/acf437
Download Date	2026-02-13 22:52:49
Rights	General rights Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights. - Users may download and print one copy of any publication from the public portal for the purpose of private study or research. - You may not further distribute the material or use it for any profit-making activity or commercial gain - You may freely distribute the URL identifying the publication in the public portal" Take down policy If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Physics in Medicine & Biology

IPEM
Institute of Physics and Engineering in Medicine

PAPER

Robust optimization of a radiotherapy pretreatment preparation workflow

To cite this article: Nienke Hoffmans-Holtzer *et al* 2024 *Phys. Med. Biol.* **69** 025022

View the article online for updates and enhancements.

You may also like

- Functional CT imaging of prostate cancer
Elizabeth Henderson, Michael F Milosevic, Masoom A Haider et al.
- Estimation of rolling bearing remaining useful life using DReLU-RA-ConvGRU model based on the encoder–decoder structure
Qile Wu, Ke Xiao, Wanquan Xiong et al.
- Multiscale assessment of the accuracy of surface replication
J Marteau, M Wieczorowski, Y Xia et al.

This content was downloaded from IP address 156.83.1.179 on 22/01/2024 at 11:45

IOP Publishing Phys. Med. Biol. 69 (2024) 025022 https://doi.org/10.1088/1361-6560/acf437

Physics in Medicine & Biology

IPEM Institute of Physics and Engineering in Medicine

RECEIVED
10 May 2023

REVISED
12 August 2023

ACCEPTED FOR PUBLICATION
25 August 2023

PUBLISHED
12 January 2024

PAPER

Robust optimization of a radiotherapy pretreatment preparation workflow

Nienke Hoffmans-Holtzer[1,*], Luuk Smolenaers[2], Manouk Olofsen-van Acht[1], Mischa Hoogeman[1], Marleen Balvert[2] and Steven Petit[1]

[1] Erasmus MC Cancer Institute, University Medical Center Rotterdam, Department of Radiotherapy, Rotterdam, The Netherlands
[2] Tilburg School of Economics and Management, Department of Econometrics and Operations Research, Tilburg University, Tilburg, The Netherlands
[*] Author to whom any correspondence should be addressed.

E-mail: n.hoffmans-holtzer@erasmusmc.nl

Keywords: operations research, health care optimization, machine learning, genetic algorithm, palliative radiotherapy

Abstract

Objective. Increasing cancer incidence, staff shortage and high burnout rate among radiation oncologists, medical physicists and radiation technicians are putting many departments under strain. Operations research (OR) tools could optimize radiotherapy processes, however, clinical implementation of OR-tools in radiotherapy is scarce since most investigated optimization methods lack robustness against patient-to-patient variation in duration of tasks. By combining OR-tools, a method was developed that optimized deployment of radiotherapy resources by generating robust pretreatment preparation schedules that balance the expected average patient preparation time (F_{mean}) with the risk of working overtime (*RoO*). The method was evaluated for various settings of an one-stop shop (OSS) outpatient clinic for palliative radiotherapy. *Approach*. The OSS at our institute sees, scans and treats 3–5 patients within one day. The OSS pretreatment preparation workflow consists of a fixed sequence of tasks, which was manually optimized for radiation oncologist and CT availability. To find more optimal sequences, with shorter F_{mean} and lower *RoO*, a genetic algorithm was developed which regards these sequences as DNA-strands. The genetic algorithm applied natural selection principles to produce new sequences. A decoder translated sequences to schedules to find the conflicting fitness parameters F_{mean} and *RoO*. For every generation, fitness of sequences was determined by the distance to the estimated Pareto front of F_{mean} and *RoO*. Experiments were run in various OSS-settings. *Main results*. According to our approach, the expected F_{mean} of the current clinical schedule could be reduced with 37%, without increasing *RoO*. Additional experiments provided insights in trade-offs between F_{mean}, *RoO*, working shift length, number of patients treated on a single day and staff composition. *Significance*. Our approach demonstrated that OR-tools could optimize radiotherapy resources by robust pretreatment workflow scheduling. The results strongly support further exploration of scheduling optimization for treatment preparation also outside a one-stop shop or radiotherapy setting.

1. Introduction

According to the European Cancer Information System (ECIS), the increase in cancer incidence is estimated at 12% in 2030 and 21.4% in 2040, compared to the incidence in 2020 (ECIS 2022). This is especially concerning because in 2016, the WHO estimated a global shortage of 18 million health workers by 2030 (World Health 2016). The COVID-19 pandemic revealed already today large vulnerabilities in hospital staff capacities and the health systems' physical infrastructure, including hospital beds and medical equipment (World Health 2022). Moreover, the increasing healthcare costs and increased awareness of the environmental effects of healthcare impose additional restrictions on healthcare systems (Lichter *et al* 2022). In addition, on-demand expectations of the

© 2024 Institute of Physics and Engineering in Medicine

patients will increase. Obviously, these developments have large impacts on radiation oncology departments in Europe and beyond. An indication that many radiotherapy (RT) departments are already under strain is the high incidence of burnout among the radiation oncologist (RO), physicists and radiotherapy technicians (RTT) (Di Tella *et al* 2020, Franco *et al* 2020a, 2020b).

Part of the solution to close the gap between supply and demand for radiation oncology could be resource optimization, i.e.: optimization of resources to mitigate shortage of the multidisciplinary professionals and reduce undesirable waiting time for patients. In many fields, operation research (OR) is used for resource optimization. The field of OR applies mathematical methods to find optimal or near-optimal solutions to decision-making problems, and is applied in for instance; manufacturing to optimize production processes, logistics to optimize inventory management, healthcare for resource allocation, finance for trading strategy development, transportation for fleet management, energy to optimize production and distribution, and telecommunications to optimize network design.

To date, only few groups have explored the potential of OR for RT (Petrovic *et al* 2011, Saure *et al* 2012, Petrovic *et al* 2013, Legrain *et al* 2015, Vieira *et al* 2020, Braune *et al* 2021). In 2016, Vieira *et al* (2016) reviewed 18 papers focused on OR problems related to strategic decision making in RT, such as patient scheduling on the linear accelerators, resource capacity planning, and patient prioritization. While they acknowledge the complexity of scheduling in RT, the challenge of addressing the stochastic uncertainty of day-to-day and patient-to-patient variation in a clinical setting remains relatively underexposed. In literature, up to our knowledge, only 4 papers included stochastic uncertainty into their models; Saure *et al* (2012), Legrain *et al* (2015), Braune *et al* (2021) reported on the impact of uncertainty in relation to RT appointment scheduling, and Vieira *et al* (2018) in relation to RT pretreatment preparation. Interestingly, to date, only Vieira *et al* (2021) reported on the clinical implementation of an OR method in RT. This could suggest a discrepancy between clinical practice and (mathematical) theory, potentially due to large variation in execution times and the lack of robustness against this uncertainty of the optimized models.

In RT in general, all pretreatment preparation tasks are predefined in terms of order and executers (ROs, RTTs and medical physicists). Typically, only tasks involving patients' presence (e.g. patient consult and planning CT acquisition) have designated (patient) appointments. Only for these tasks, specific executers are appointed at a specific time, thus these preparation tasks are scheduled. Vieira *et al* (2019) investigated the impact of scheduling the treatment starting date in different scenarios. However, robust scheduling of pretreatment preparation tasks themselves has not yet been investigated, presumably because of its complexity due to the relatively large number of tasks, relations between executers, and potential dependency on input from other departments (e.g. medical oncology, histology and radiology). This complexity causes pretreatment preparation times from days to weeks, especially in curative and palliative RT for patients with good to moderate life expectancy. For patients with poor life expectancy, the radiotherapy pretreatment preparation workflow is similar, however, with limited dependency on input from other departments. At our institute, a selection of these patients is treated at the dedicated one-stop shop (OSS) outpatient clinic for palliative RT, which is running in parallel with the regular RT clinic. These OSS-patients are treated on regular C-arm linacs like most patients at our regular RT clinic. However, within the OSS-setting, 3 to 5 patients are offered palliative RT the same day they had their first consult with the RO. In order to achieve pretreatment preparation within hours, all pretreatment preparation tasks are executed by a limited number of dedicated staff. Therefore, the OSS for palliative RT is an ideal use case to study the use of OR for the robust scheduling for pretreatment preparation in RT.

Patients treated in OSSs are often frail and in pain and are treated to control symptoms, alleviate pain, and increase quality of life in their final stage of life (Jones *et al* 2014, Tiwana *et al* 2016). So, to optimize the workflow from a patient perspective, the time patients spend in the hospital (i.e.: flow time) should be minimized. Minimization of the flow time for a single patient would imply that all preparation steps are performed consecutively. For efficiency, on the other hand, it is beneficial to have multiple patients in the preparation phase simultaneously, such that the idle time between the tasks of the professionals is minimized and patient throughput is increased. From a logistical point of view, some idle time would be beneficial to account for the stochastic nature of the duration of different preparation steps and therefore avoid unnecessary delays for patients and overtime for staff. From a financial perspective, the number of professionals dedicated to the OSS versus the number of patients treated should be minimal. In other words, scheduling in a predefined OSS-setting involves a trade-off between minimizing the expected average flow time per patient, the number of patients that can be treated per day, the number of dedicated professionals involved and the risk of working overtime, i.e.: the robustness of the process. Balancing these parameters for the predefined tasks and setting at the OSS, enables the investigation of robust scheduling and optimization for sensible deployment of RT staff, while taking robustness and patients' interest into account.

The goal of the current study was to apply operation research-tools to optimize the radiotherapy treatment preparation schedule for multiple patients per day treated at the OSS, accounting for trade-offs between staff composition, flow time and risk of overtime. To this end, a method was developed that could schedule and

Table 1. Overview of the tasks and operators used in the OSS-setting at the Erasmus Medical Center. Lunch breaks are not included as task; however, they were included in the scheduling algorithm. Here RO = Radiotherapy Oncologist, RTT = Radiotherapy Technician, CT = Computed Tomography, QA = Quality Assurance, R&V = Record and Verify system.

#	Task	Executer	Description
1	Intake	RO	Patient is examined and informed about the treatment
2	Patient setup and CT acquisition	RTT1 and RTT2 (2RTTs)	Two RTTs together determine the patient setup and make a CT scan of the tumor and surrounding organs in the treatment position. This CT scan is automatically fed into the segmentation algorithm
3	Organs at risk segmentation	AutoSeg	The patient body contour and other critical organs at risk are automatically segmented (MIM Software Inc.) and the results are automatically imported in the treatment planning system (Monaco, Elekta)
4	Target volume delineation	RO	The RO delineates the tumor based on the information from the intake and the planning CT
5	Peer consultation and administration	RO	The RO discusses the results with a peer RO and reports the findings in the patients' chart. As this consultation is relatively short (7 min on average), no dedicated peers are included in the scheduling problem and peers are assumed to be promptly available, reflecting current clinical routine
6	Treatment planning preparation	RTT1 or RTT2	One of the two RTTs starts the 'Treatment planning preparation' based on the planning CT and delineations after checking the automatic segmentation results
7	Treatment planning	RTT1 or RTT2	One of the two RTTs starts the treatment planning process that results in a dedicated treatment plan, which will steer the treatment machine during dose delivery
8	Treatment planning check	RTT1 or RTT2	The treatment plan is checked for errors and omissions. This task cannot be performed by the same RTT who did the 'Treatment Planning'
9	Treatment planning approval	RO	The calculated radiotherapy dose is visualized together with the delineations onto the planning CT. Based on the visualization and corresponding statistics, the RO checks the treatment plan and will give the approval if the plan meets the criteria
10	Treatment planning post-processing	RTT1 or RTT2	After some final post-processing tasks, the approved plan (together with the planning CT, delineations and calculated dose) will be exported by one of the RTTs for AutoQA
11	Physics quality assurance check	AutoQA	This system performs an automated physics plan quality assurance including independent 3D dose calculation (SciMoCa^{TM}, Scientific-RT), if the treatment plan meets the criteria, the AutoQA system initiates dataset import in the record and verify system (R&V, Mosaiq, Elekta) of the radiotherapy treatment machine (Synergy or Versa HD, Elekta). These automated checks are run under the responsibility of a medical physics expert
12	R&V import and manual approval	RTT1 or RTT2	After the automatic import, a manual approval is performed by one of the RTTs
13	R&V import check	RTT1 or RTT2	The 'R&V import check' is performed by another RTT than the RTT who has performed the 'R&V import and manual approval'

optimize pretreatment preparation tasks, by combining a multi-objective genetic algorithm with a mixed integer linear program. This method estimates pseudo-Pareto fronts of potential schedules between the average flow time versus the risk of overtime. Solving the scheduling problem for different scenarios, i.e.: a varying number of patients, composition of the team of dedicated professionals, and opening times of the clinic, informs the user on decisions at the tactical level. Also, the selected schedules could be used for decision making at an operational level, as they are ready to be implemented in the clinic. Based on the knowledge obtained from this research, new methods could be developed that could optimize the more complex setting of an entire RT department.

2. Methods

2.1. Radiotherapy treatment preparation at the OSS

As starting point for the model, we use an OSS-setting with one dedicated RO and two all-round RTTs that are both trained to perform treatment planning (i.e.: dosimetrists). This corresponds to the clinical setting at Erasmus MC Cancer Institute, University Medical Center Rotterdam (Erasmus MC). The processing tasks, based on the Erasmus MC setting, and their executers are described in table 1. In general, these tasks apply to any

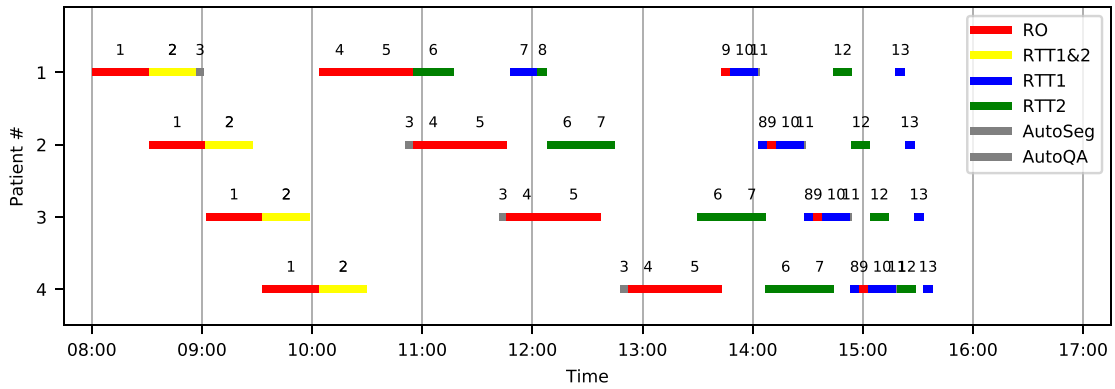

Figure 1. Schematic example of a clinical OSS workflow, with average duration of tasks calculated based on 663 retrospective patients, at the Erasmus MC with 13 tasks per patient, executed by 5 different executers; Radiation Oncologist in red, Radiotherapy Technicians in yellow (both), blue (first) and green (second), and Automatic Segmentation software in gray. The Automatic Quality Assurance (task 11) is also represented in gray; however it is hardly visible as the execution time of this task is only 1 min.

palliative radiotherapy treatment though slight differences between centers may exist. At Erasmus MC, the auto segmentation (AutoSeg) and physics independent dose verification (AutoQA) are performed automatically. All patients in the OSS setting are treated on regular C-arm linacs (Elekta's Versa HD or Synergy). A schematic example of the clinical OSS workflow at Erasmus MC is illustrated in figure 1.

2.2. Solving the OSS scheduling problem

The backbone of our approach to evolve from the current clinical schedule to optimized schedules is a genetic algorithm. The genetic algorithm optimizes the order of the different pretreatment preparation tasks, i.e. the sequence, over a number of generations by regarding this sequence as a single DNA-strand. Key in this algorithm is a decoder. As a first step, this decoder assigns the partially ordered tasks in a sequence to their dedicated executers. This could be explained by the following example in which a pretreatment preparation workflow is considered for four patients with two tasks. According to table 1, the first task would be executed by an RO and the second task by two RTTs. A hypothetical sequence of two taks per patient and four patients of [1, 4, 2, 1, 3, 4, 2, 3] would then translate to (patient 1, task 1, RO), (patient 4, task 1, RO), (patient 2, task 1, RO), (patient 1, task 2, 2RTTs), (patient 3, task 1, RO), (patient 4, task 2, 2RTTs), (patient 2, task 2, 2RTTs), (patient 3, task 2, 2RTTs). By combining these assigned tasks with the according distributions of the execution times per task, the decoder generates schedules as demonstrated in figure 1. In our OSS setting, there are 13 tasks per patient and 4 patients, so the sequence has 52 integers. Next, based on principles of natural selection, the genetic algorithm applies processes of mutation, cross-over and selection to combine parent sequences to yield new offspring sequences. Selection is based on the fitness of the sequences which is determined by 2 parameters that are calculated from the decoded schedules; (1) the expected average flow time (F_{mean}), i.e.: the average pretreatment preparation time from start of intake to start of treatment, and (2) the risk of overtime (*RoO*), i.e.: the probability that not all treatment preparation steps for the patients of a specific day can be finished within working hours.

The implementation of the genetic algorithm and the decoder are further explained in the following sections. The algorithm is implemented in Python 3.8, and the decoders (based on a mixed integer linear program (MILP) and a linear program (LP)) are optimized using Gurobi version 9.1.2, Gurobi Optimization, LLC.

2.2.1. Genetic algorithm

The genetic algorithm applied in our research is based on the non-dominated sorting genetic algorithm (NSGA-II) for bi-objective optimization as described by Deb *et al* (2002). Per generation, this genetic algorithm selects the fittest sequences determined by the distance to the pseudo-Pareto front of two conflicting objectives; the closer to the front, the fitter the sequence. Over the generations, the temporary pseudo-Pareto front will be pushed to approach its optimal shape, by means of natural selection.

A flowchart of the NSGA-II algorithm for the OSS scheduling problem is shown in figure 2 and explained below.

Our NSGA-II algorithm starts with generation *G* of *N* random sequences (1). Each sequence represents a partial ordering of the tasks to be performed. Each sequence is decoded into a schedule using a mixed integer linear program that minimizes the expected F_{mean} ($\text{MILP}_{\text{Fmean}}$) and a linear program that calculates the *RoO* (LP_{RoO}) (2), see next section for details. Next, a new generation (*G+1*) is formed consisting of *N* sequences, by

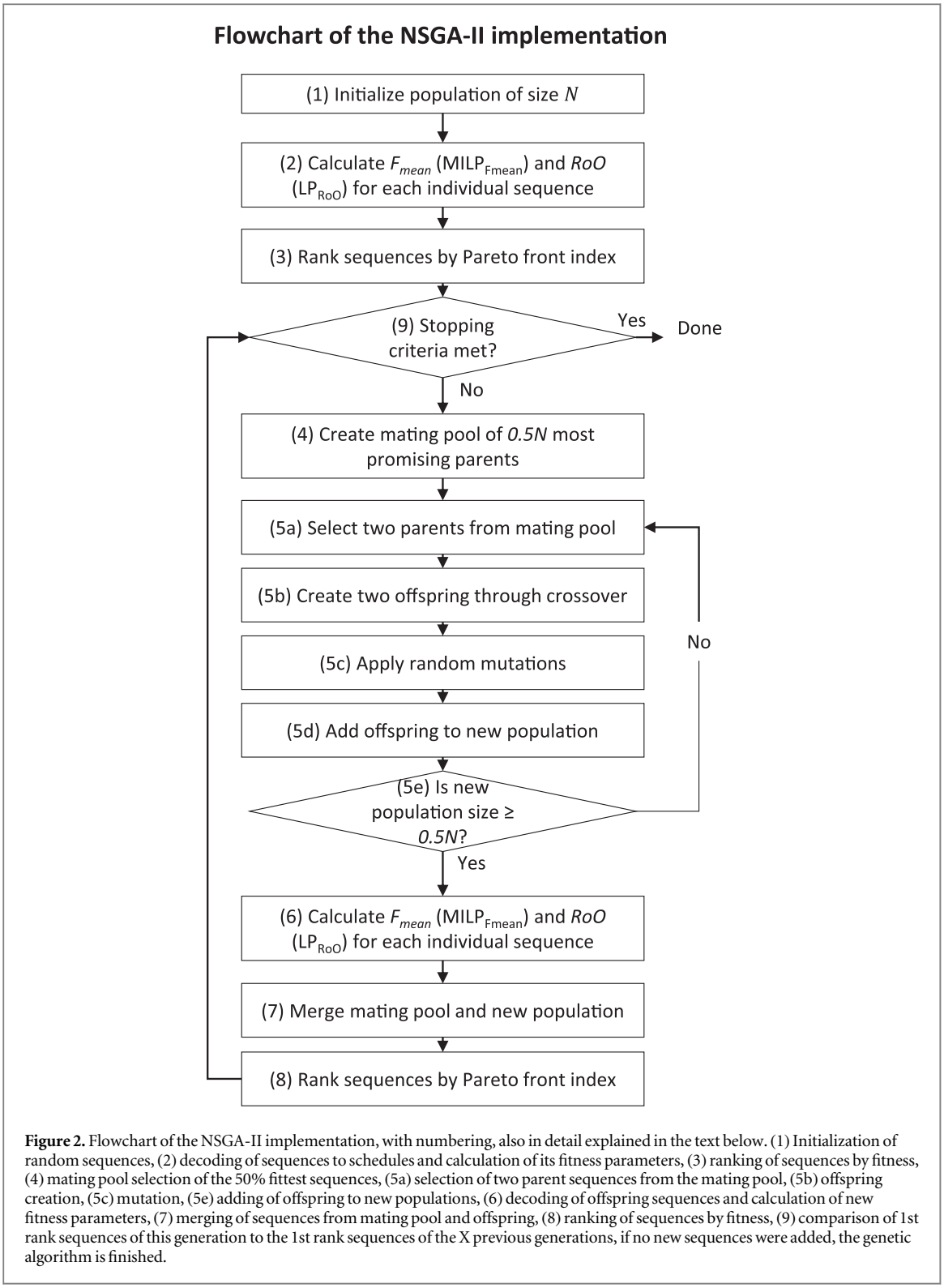

Figure 2. Flowchart of the NSGA-II implementation, with numbering, also in detail explained in the text below. (1) Initialization of random sequences, (2) decoding of sequences to schedules and calculation of its fitness parameters, (3) ranking of sequences by fitness, (4) mating pool selection of the 50% fittest sequences, (5a) selection of two parent sequences from the mating pool, (5b) offspring creation, (5c) mutation, (5e) adding of offspring to new populations, (6) decoding of offspring sequences and calculation of new fitness parameters, (7) merging of sequences from mating pool and offspring, (8) ranking of sequences by fitness, (9) comparison of 1st rank sequences of this generation to the 1st rank sequences of the X previous generations, if no new sequences were added, the genetic algorithm is finished.

first selecting the *0.5N* best sequences of the current generation *G* and creating an additional *0.5N* offspring. The *0.5N* best sequences are selected according to the Pareto front ranking method as described by Deb *et al* (2002) as follows (3).

The sequences at the pseudo-Pareto front of F_{mean} and *RoO* get rank 1 (see also figure 3) and are added to the mating pool (with size *0.5N*) of generation *G+1*. After removal of rank 1 sequences, the sequences at the new front get rank 2 and are added to the mating pool as well, and so forth until rank *r*, where *r* refers to the highest rank for which all sequences still fit entirely in the mating pool (rank 6 in figure 3). The remaining sequences to fill the mating pool upto *0.5N* are selected from rank *r+1* (rank 7 in figure 3, which would not entirely fit in the mating pool) based on a crowding-distance calculation (Deb *et al* 2002) to ensure preservation of diversity. The fronts are extracted one by one using an open-source code snippet (O'Connor 2018).

Table 2. Sets that are used in $MILP_{Fmean}$.

	$MILP_{Fmean}$
Sets	Description
$\mathscr{P}$	Set of patients
$\mathscr{O}$	Set of operator types. $\mathscr{O}$ = { ROs, RTTs, AutoSeg, AutoQA}
ε	Set of executers. $\varepsilon = \varepsilon_{ROs} \cup \varepsilon_{RTTs} \cup \varepsilon_{AutoSeg} \cup \varepsilon_{AutoQA}$ = { RO_1, RO_2 } ∪ { RTT_1, RTT_2 } ∪ {AutoSeg} ∪ {AutoQA}, in the case of two ROs and two RTTs
$\mathscr{T}_e$	Set of tasks that can be performed by executer *e*. Note that all tasks in $\mathscr{T}_{ROs}$ will be executed by either RO_1 or RO_2, and that tasks in $\mathscr{T}_{RTTs}$ will be executed by either RTT_1 and/or RTT_2
$\mathscr{T}$	$\mathscr{T} = \mathscr{T}_{ROs} \cup \mathscr{T}_{RTTs} \cup \mathscr{T}_{AutoSeg} \cup \mathscr{T}_{AutoQA}$
$\mathscr{T}_{2RTTs}$	Set of tasks that require two RTTs, where $\mathscr{T}_{2RTTs} \subset \mathscr{T}_{RTTs}$
$\mathscr{T}_{diffRTTs}$	Set of tasks that need to be performed by a different RTT than the previous task, where $\mathscr{T}_{diffRTTs} \subset \mathscr{T}_{RTTs}$
$\mathscr{X}_e$	Precedence relations of tasks for operator type $\mathscr{O}$. Each task is described as a patient-task pair *(p, t)*. For example, ((*p′, t′*), (*p*", *t*")) $\mathscr{T}$ $\mathscr{X}_{RTTs}$ means that the RTTs cannot start working on operation *t*ⲋⲋ on patient *p*" before they have started working on operation *t*' on patient *p*'

Table 3. Parameters that are used in $MILP_{Fmean}$.

	$MILP_{Fmean}$
Parameters	Description
r_t	Execution time for task *t*
M	A sufficiently large number
P	The number of patients
S	Threshold for pretreatment preparations
p_{first}	The first patient (patient 1)
t_{first}	The first task (task 1)
t_{final}	The last task (task 13)
B_{start}	Start of the break time window (expressed in minutes from start of the first task)
B_{end}	End of the break time window (expressed in minutes from start of the first task)
B_{RO}	Minimal break length for an RO
B_{RTT}	Minimal break length for an RTT

After creating the mating pool (4), the algorithm generates *0.5N* offspring (5). First, two parents are selected at random (5a) that create two offspring by a random single point cross-over (5b). Next, *M* mutations are applied to the offspring by swapping two random elements in the sequence (5c). Offspring are added to the new population (5d) to a total of *0.5N* (5e). The expected F_{mean} and *RoO* for the offspring are calculated by $MILP_{Fmean}$ and LP_{RoO} (6). The new generation is formed by the *0.5N* elements in the mating pool and the *0.5N* offspring (7) and sequences are ranked based on the new pseudo-Pareto front (8). New generations are added by the NSGA-II until no new points are added to the rank 1 fronts in *X* subsequent generations (9).

2.2.2. Decoder for computation of expected F_{mean} *and* RoO

The expected F_{mean} and *RoO* are calculated by a mixed integer linear program $MILP_{Fmean}$ and a linear program LP_{RoO}, respectively. The $MILP_{Fmean}$ has a sequence, mean durations, and possible executers of the tasks as input in order to optimize the associated schedule by minimizing its expected F_{mean}. Therefore, $MILP_{Fmean}$ has the minimized F_{mean} and optimized schedule for the associated sequence as main output. The $MILP_{Fmean}$ also optimizes the start time of each patient and task, and the time of the lunch breaks. Next, the LP_{RoO} uses the results from $MILP_{Fmean}$ to quickly optimize the schedule with the same associated sequence for a series of new scenarios from a presampled distribution of varying durations of tasks, by minimizing the new expected F_{mean}. Through LP_{RoO} the finishing time per scenario is determined and the impact of varying durations of tasks in different scenarios is evaluated. Eventually, the *RoO* is computed by determining the percentage of scenarios expected to be finished before the end of the working shift. Both $MILP_{Fmean}$ and LP_{RoO} are explained below.

2.2.2.1. $MILP_{Fmean}$*: determining start times and assigning executers to minimize the expected* F_{mean}

The expected F_{mean} of a sequence cannot be calculated directly since it depends on the start time of each patient and the executer that performs the task. Therefore, prior to solving $MILP_{Fmean}$, the sequences from the NSGA-II algorithm are translated to ordered lists of tasks per executer type (Gen *et al* 1994, Petrovic *et al* 2011), i.e.: RO,

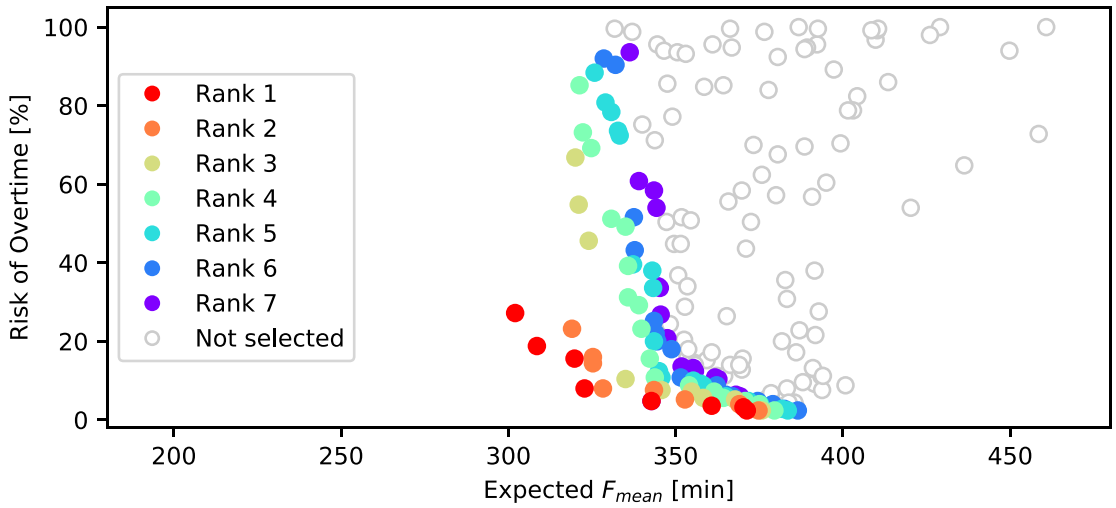

Figure 3. Example of the (pseudo-) Pareto front ranking of the fittest sequences at the 10th generation.

Table 4. Variables that were used in $\text{MILP}_{\text{Fmean}}$.

	$\text{MILP}_{\text{Fmean}}$
Variables	Description
F_p	Flow time of patient p
$s_{e,p,t}$	Time that executer e starts working on task t for patient p
$a_{e,p,t}$	1 if executer e performs task t for patient p, 0 otherwise (assignment variable). Used for RTT, AutoSeg and AutoQA tasks.
$u_{e,p}$	1 if executer e is assigned to patient p, 0 otherwise (assignment variable). Used for RO tasks, as all tasks for one patient are executed by the same RO and hence the subscript t can be dropped.
$b_{e,p,t}$	1 if executer e performs task t for patient p before lunch break, 0 otherwise (assignment variable)
bt_e	Time that executer e starts a lunch break

RTT, AutoSeg and AutoQA. The RTT tasks can be subdivided into a joint task for 2RTTs (denoted by the set $\mathcal{T}_{\text{2RTTs}}$), individual tasks for one RTT ($\mathcal{T}_{\text{RTT}}$), and checks of a previous task that needs to be done by another RTT than the one who executed the previous task ($\mathcal{T}_{\text{diffRTTs}}$). These precedence relations are stored in the set χ_e and are used as input for $\text{MILP}_{\text{Fmean}}$. The complete sets, parameters, and variables of $\text{MILP}_{\text{Fmean}}$ are described in tables 2, 3 and 4, respectively.

The objective and constraints of $\text{MILP}_{\text{Fmean}}$ are described below. $\text{MILP}_{\text{Fmean}}$ minimizes the expected F_{mean} (1) while satisfying the constraints that are imposed by the local OSS workflow, e.g.: the RO that does the intake also does the other RO tasks of the same patient, the RTTs do the patient setup and CT acquisition together, and if the treatment planning or R&V import are performed by one RTT, the other RTT performs the checks. These and other practical constraints are translated to the constraints that are listed and described below.

$$\min \frac{1}{P}\sum_{p\in P} F_p \tag{1}$$

$$\begin{aligned} \text{s.t.} \quad & s_{e,p',t'} + r_{t'} - M(1 - u_{e,p'}) \leqslant s_{e,p'',t''} \\ & +M(1 - u_{e,p''}) \quad \forall\, e \in \mathcal{E}_{ROs}, \forall\, ((p',\, t'),\ (p'',\, t'')) \in \mathcal{X}_e \end{aligned} \tag{2}$$

$$\begin{aligned} & s_{e,p',t'} + r_{t'} - M(1 - a_{e,p',t'}) \leqslant s_{e,p'',t''} + M(1 - a_{e,p'',t''}) \\ & \forall\, e \in \mathcal{E}_{\text{RTTs}} \cup \mathcal{E}_{\text{AutoSeg}} \cup \mathcal{E}_{\text{AutoQA}}\ \forall\, ((p',\, t'),\ (p'',\, t'')) \in \mathcal{X}_e \end{aligned} \tag{3}$$

$$\begin{aligned} & s_{e',p,t-1} + r_{t-1} - M(1 - a_{e',p,t-1}) \leqslant s_{e'',p,t} + M(1 - a_{e'',p,t}) \\ & \forall\, t \in \mathcal{T},\ \forall\, p \in \mathcal{P},\ \forall\, e' \in \mathcal{E}_{\text{RTTs}} \cup \mathcal{E}_{\text{AutoSeg}} \cup \mathcal{E}_{\text{AutoQA}}, \\ & \forall\, e'' \in \mathcal{E}_{\text{RTTs}} \cup \mathcal{E}_{\text{AutoSeg}} \cup \mathcal{E}_{\text{AutoQA}} \end{aligned} \tag{4}$$

$$\begin{aligned} & s_{e,p,t-1} + r_{t-1} - M(1 - a_{e',p,t-1}) \leqslant s_{e'',p,t} + M(1 - u_{e'',p}) \\ & \forall\, t \in \mathcal{T},\ \forall\, p \in \mathcal{P},\ \forall\, e' \in \mathcal{E}_{\text{RTTs}} \cup \mathcal{E}_{\text{AutoSeg}} \cup \mathcal{E}_{\text{AutoQA}},\ \forall\, e'' \in \mathcal{E}_{\text{ROs}} \end{aligned} \tag{5}$$

Table 5. Sets that are used in the LP_{RoO}.

	LP_{RoO}
Sets	Description
$\mathscr{P}$	Set of patients
$\mathscr{X}_{MILPFmean}$	Precedence relations of tasks for executer e as optimized by $MILP_{Fmean}$. Each task is described as a patient-task pair (p, t). For example, $((p', t'), (', t')) \in \mathscr{X}_{MILPFmean}$ means that the executer cannot start working on operation t' on patient p' before they have started working on operation t' on patient p'.

Table 6. Parameters that are used in the LP_{RoO}.

	LP_{RoO}
Parameters	Description
$r_{p,t}$	Execution time for task t of patient p (including patient specific lunch break time), varying per task and each patient based on a presampled execution times distribution
t_first	First task
t_final	Last task
$s^{MILPFmean}_{p,t\_first}$	Start time of the first task of every patient from $MILP_{Fmean}$ (it cannot be expected that patients arrive early)

$$s_{e',p,t-1} + r_{t-1} - M(1 - u_{e',p}) \leqslant s_{e'',p,t} + M(1 - a_{e'',p,t}) \quad \forall\, t \in \mathscr{T},\ \forall\, p \in \mathscr{P},\ \forall\, e' \in \mathcal{E}_{\text{ROs}},\ \forall\, e'' \in \mathcal{E}_{\text{RTTs}} \bigcup \mathcal{E}_{\text{AutoSeg}} \bigcup \mathcal{E}_{\text{AutoQA}} \tag{6}$$

$$s_{e',p,t-1} + r_{t-1} - M(1 - u_{e,,p}) \leqslant s_{e'',p,t} + M(1 - u_{e'',p})\ \forall\, t \in \mathscr{T},\ \forall\, p \in \mathscr{P},\ \forall\, e' \in \mathcal{E}_{\text{ROs}},\ \forall\, e'' \in \mathcal{E}_{\text{ROs}} \tag{7}$$

$$s_{e',p,t} = s_{e'',p,t}\ \forall\, t \in \mathscr{T}_{\text{2RTTs}},\ \forall\, p \in \mathscr{P},\ \forall\, e' \in \{\text{RTT}_1\},\ \forall\, e'' \in \{\text{RTT}_2\} \tag{8}$$

$$F_p \geqslant s_{e,,p,t\_final} + r_{t\_final} - s_{e,,,p,t\_first} \tag{9}$$

$$\sum_e a_{e,p,t} = 1\ \forall\, \text{e} \in \mathcal{E}_{\text{RTTs}},\ \forall\, p \in \mathscr{P},\ \forall\, t \in \mathscr{T}\text{e},\ \forall\, t \notin \mathscr{T}_{\text{2RTTs}} \tag{10}$$

$$a_{e,p,t} = 1 \quad \forall\, e \in \mathcal{E}_{\text{AutoSeg}},\ \forall\, p \in \mathscr{P},\ \forall\, t \in \mathscr{T}_{\text{e}} \tag{11}$$

$$a_{e,p,t} = 1 \quad \forall\, e \in \mathcal{E}_{\text{AutoQA}},\ \forall\, p \in \mathscr{P},\ \forall\, t \in \mathscr{T}_{\text{e}} \tag{12}$$

$$\sum_e u_{e,p} = 1 \quad \forall\, e \in \mathcal{E}_{\text{ROs}},\ \forall\, p \in \mathscr{P} \tag{13}$$

$$s_{e,p,t} + r_t \leqslant \text{S} \quad \forall\, e \in \mathcal{E}_{\text{ROs}} \bigcup \mathcal{E}_{\text{RTTs}} \bigcup \mathcal{E}_{\text{AutoSeg}} \bigcup \mathcal{E}_{\text{AutoQA}},\ \forall\, p \in \mathscr{P},\ \forall\, t \in \mathscr{T} \tag{14}$$

$$a_{e,p,t} = 0 \quad \forall\, e \in \mathcal{E}_{\text{AutoSeg}} \bigcup \mathcal{E}_{\text{AutoQA}},\ \forall\, p \in \mathscr{P},\ \forall\, t \in \mathscr{T}_{\text{RTTs}} \tag{15}$$

$$a_{e,p,t} = 0 \quad \forall\, e \in \mathcal{E}_{\text{RTTs}} \bigcup \mathcal{E}_{\text{AutoQA}},\ \forall\, p \in \mathscr{P},\ \forall\, t \in \mathscr{T}_{\text{AutoSeg}} \tag{16}$$

$$a_{e,p,t} = 0 \quad \forall\, e \in \mathcal{E}_{\text{RTTs}} \bigcup \mathcal{E}_{\text{AutoSeg}},\ \forall\, p \in \mathscr{P},\ \forall\, t \in \mathscr{T}_{\text{AutoQA}} \tag{17}$$

$$\sum_e a_{e,p,t} = 2 \quad \forall\, e \in \mathcal{E}_{\text{RTTs}},\ \forall\, p \in \mathscr{P},\ \forall\, t \in \mathscr{T}_{\text{2RTTs}} \tag{18}$$

$$a_{e,p,t'} = 1 - a_{e,p,t''} \quad \forall\, e \in \mathcal{E}_{\text{RTTs}},\ \forall\, p \in \mathscr{P},\ \forall\, t' \in \mathscr{T}_{\text{diffRTTs1}},\ \forall\, t'' \in \mathscr{T}_{\text{diffRTTs2}} \tag{19}$$

$$b_{e,p',t'} \geqslant b_{e,p'',t''} \quad \forall\, e \in \mathcal{E}_{\text{ROs}} \bigcup \mathcal{E}_{\text{RTTs}},\ \forall\, ((p', t'), (p'', t'')) \in \mathscr{X}_{\text{e}} \tag{20}$$

$$bt_e \geqslant B_{start} \quad \forall\, e \in \mathcal{E}_{\text{ROs}} \bigcup \mathcal{E}_{\text{RTTs}} \tag{21}$$

$$bt_e + B_{RO} \leqslant B_{end} \quad \forall\, e \in \mathcal{E}_{\text{ROs}} \tag{22}$$

$$bt_e + B_{RTT} \leqslant B_{end} \quad \forall\, e \in \mathcal{E}_{\text{RTTs}} \tag{23}$$

$$bt_e \geqslant s_{e,p,t} + r_t - M(1 - b_{e,p,t}) - M(1 - a_{e,p,t}) \quad \forall\, e \in \mathcal{E}_{\text{RTTs}},\ \forall\, p \in \mathscr{P},\ \forall\, t \in \mathscr{T}_{\text{e}} \tag{24}$$

$$bt_e \geqslant s_{e,p,t} + r_t - M(1 - b_{e,p,t}) - M(1 - u_{e,p}) \quad \forall\, e \in \mathcal{E}_{\text{ROs}},\ \forall\, p \in \mathscr{P},\ \forall\, t \in \mathscr{T}_{\text{e}} \tag{25}$$

$$bt_e + B_{RTT} \leqslant s_{e,p,t} + Mb_{e,p,t} + M(1 - a_{e,p,t}) \quad \forall\, e \in \mathcal{E}_{\text{RTTs}},\ \forall\, p \in \mathscr{P},\ \forall\, t \in \mathscr{T}_{\text{e}} \tag{26}$$

$$bt_e + B_{RO} \leqslant s_{e,p,t} + Mb_{e,p,t} + M(1 - u_{e,p}) \quad \forall\, e \in \mathcal{E}_{\text{ROs}},\ \forall\, p \in \mathscr{P},\ \forall\, t \in \mathscr{T}_{\text{e}} \tag{27}$$

$$s_{RO1,p\_first,t\_first} = 0. \tag{28}$$

Constraints (2) and (3) ensure that the ROs, the RTTs and automated operators are working on only one task at a time and that the ordering imposed by the ordering sequence is respected. Constraints (4)–(7) ensure that no

Table 7. Variables that are used in the LP_{RoO}.

LP_{RoO}	
Variables	Description
$s_{p,t}$	Start time for task t of patient p
F_p	Flow time of patient p

two executers can work on the same patient simultaneously. The tasks that require both RTTs need to start at the same time for both RTTs, which is enforced by constraints (8). Constraints (9) compute the expected F_{mean} for each patient. Constraints (10) ensure that every RTT task is assigned to exactly one RTT. Constraints (11) and (12) ensure that the AutoSeg and respectively AutoQA tasks are executed for all patients. Constraints (13) assign each patient to exactly one RO. All tasks need to be finished before the end of the day, which is enforced by constraints (14). Constraints (15)–(17) ensure that all tasks are executed by the required professionals, and constraints (18) assign the task that needs to be performed by two RTTs to two RTTs. If one RTT is assigned to a specific task that needs to be checked by another RTT, constraints (19) assign the check to another RTT than the one who has performed the task. Constraints (20) ensure that if a task is scheduled before the break, then all tasks preceding this task are scheduled before the break as well. The lunch breaks for all ROs and RTTs are within a fixed break time window, which is imposed by constraints (21)–(23). Constraints (24) and (25) ensure that the RTT break respectively the RO break starts after the last RTT, respectively RO task, before the break has finished, and constraints (26) and (27) place a lower bound of B_{RTT} on the lunch break duration for RTTs and B_{RO} for ROs, respectively. Constraints (28) ensure that the first task of the first patient starts at the beginning of the day and, to avoid degeneracy, that this task is assigned to the first RO (RO1).

2.2.2.2. LP_{RoO}: determining the risk of overtime

Since execution times are stochastic, the schedule may experience some delays which could result in working overtime. To address this, execution times are randomly sampled for each task to obtain a realization of the schedule for a single day, i.e.: a scenario. Sampling many scenarios results in an overview of the variability of the working day length. The *RoO* is determined as the percentage of scenarios expected to finish after the end of the working shift, according to equation (29).

$$RoO = 100\frac{\text{number of scenarios} > S}{\text{number of scenarios}}. \tag{29}$$

Here, threshold S is predefined by the end of the working shift as specified in $MILP_{Fmean}$.

The expected finish times are computed by LP_{RoO} for each schedule obtained by $MILP_{Fmean}$ while varying (pre-sampled) execution times per task for each patient. The lunch breaks are incorporated in these execution times according to break time assignment values from $MILP_{Fmean}$. LP_{RoO} uses the following sets, parameters, variables, and optimized ordering and precedence relations of $MILP_{Fmean}$, see tables 5, 6 and 7, respectively.

Based on the order of tasks, start times of the first task of every patient and the (randomly pre-sampled) duration of tasks (including break time) from $MILP_{Fmean}$, LP_{RoO} is given by;

$$\min \frac{1}{P}\sum_{p\in P} F_p \tag{30}$$

$$\text{s.t.} \quad s_{p',t'} + r_{p-1,t-1'} \leqslant s_{p'',t'''} \quad \forall\ ((p',\ t'),\ (p'',\ t'')) \in \mathscr{X}_{\text{MILPFmean}} \tag{31}$$

$$F_p \geqslant s_{p,t\_final} + r_{p-1,t\_final-1} - s_{p,t\_first} \quad \forall\ p \in \mathscr{P} \tag{32}$$

$$s_{p,t\_first} \geqslant s^{MILPFmean}_{p,t\_first} \quad \forall\ p \in \mathscr{P}. \tag{33}$$

Here, the expected F_{mean} is optimized (30) while satisfying three constraints. Constraints (31) assure that the scheduled order of the tasks for each executer of $MILP_{Fmean}$ is copied to LP_{RoO}, constraints (32) calculate the expected flow time of each patient for a scenario from the presampled distribution of task execution times, and constraints (33) ensure that the start time for every patient is not before the planned start time from $MILP_{Fmean}$. Note that LP_{RoO} does not optimize any planning decisions, it merely computes the expected finish times for different scenarios in an efficient way to determine the *RoO*.

2.3. Average execution times per task and probability distributions

As input for $MILP_{Fmean}$ and LP_{RoO}, real-life execution times of the 11 manually executed processing tasks were collected from 663 patients that were treated at the OSS of Erasmus MC from October 2019 to October 2020. As there was little variation in the execution times for the automated AutoSeg and AutoQA tasks, these were fixed

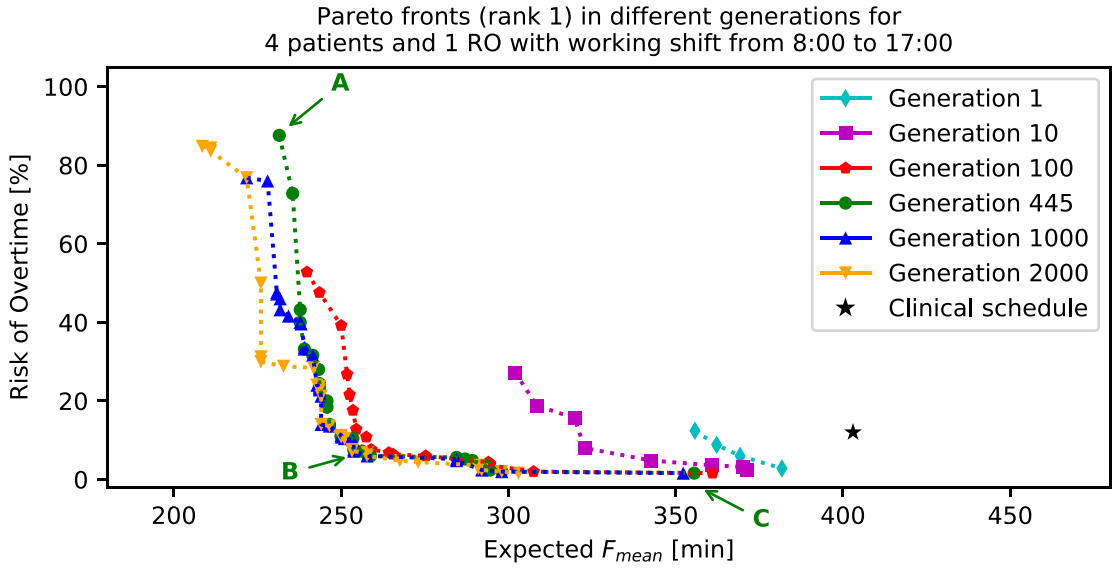

Figure 4. This figure shows pseudo-Pareto fronts of schedules for different generations of the NSGA-II algorithm. The 445th generation (green dots) fulfills the stopping criteria: i.e. the front did not change for the next 50 generations after generation 445. Every marker represents a schedule and the schedules are connected with dotted lines for readability. The green arrows indicate schedules A, B and C that are shown in figure 5. Note that as the F_{mean} increases, the risk of overtime decreases. This is may seem counterintuitive and is explained in more detail in the discussion and the caption of figure 5.

Table 8. Results of the Kolmogorov-Smirnov tests per task, based on the data of 663 retrospectively analyzed patients. The average processing times per task were used as the input for $MILP_{Fmean}$ and the distributions for sampling the variation of duration of tasks as input for LP_{RoO}. The automated tasks are noted with an * and are not assessed with Kolmogorov-Smirnov testes, but fixed at 4 min and 1 min, respectively.

#	Task	Distribution	Execution time	
			Mean [min]	Stdev [min]
1	Intake	*f*	31	10
2	Patient Setup and CT Acquisition	beta	26	13
3	Automatic Segmentation*	Not applicable	4	0
4	Target Volume Delineation	gamma	22	11
5	Peer Consultation and Administration	chi^2	29	12
6	Treatment Planning Preparation	lognorm	22	7
7	Treatment Planning	exponential	15	8
8	Treatment Planning Check	beta	5	2
9	Treatment Planning Approval	beta	5	2
10	Treatment Planning Post-Processing	gamma	15	5
11	Automatic QA*	Not applicable	1	0
12	R&V Import and Manual Approval	beta	10	5
13	R&V Import Check	gamma	5	2
	Total		190	

according to the current clinical setting at 4 min and 1 min, respectively. Based on the cleaned data, a Kolmogorov-Smirnov test was applied to determine which distribution fitted the execution times for the 11 manual tasks best. The data was tested for beta, chi-squared, exponential, *f*, gamma, logistic, lognormal, normal, pareto, powerlaw, *t*, wald, weibull-min and weibull-max distributions, by using the kstest() function from the Python scipy.stats library (2020). From the best fitting distributions, the mean execution time per task was used as input for $MILP_{Fmean}$. For LP_{RoO}, the distributions of the 13 tasks were approximated by taking 250 random samples from the fitted distribution (appendix A.1) for each patient, yielding a $13 \times 250 \times P$ matrix ($r_{p,t}$) as input for LP_{RoO}, where P is the number of patients.

2.4. Experiments

The NSGA-II algorithm was tuned for different values of population size, mutation rate and stopping criteria threshold. Based on this tuning, the following parameters were used: population size $N = 200$, number of

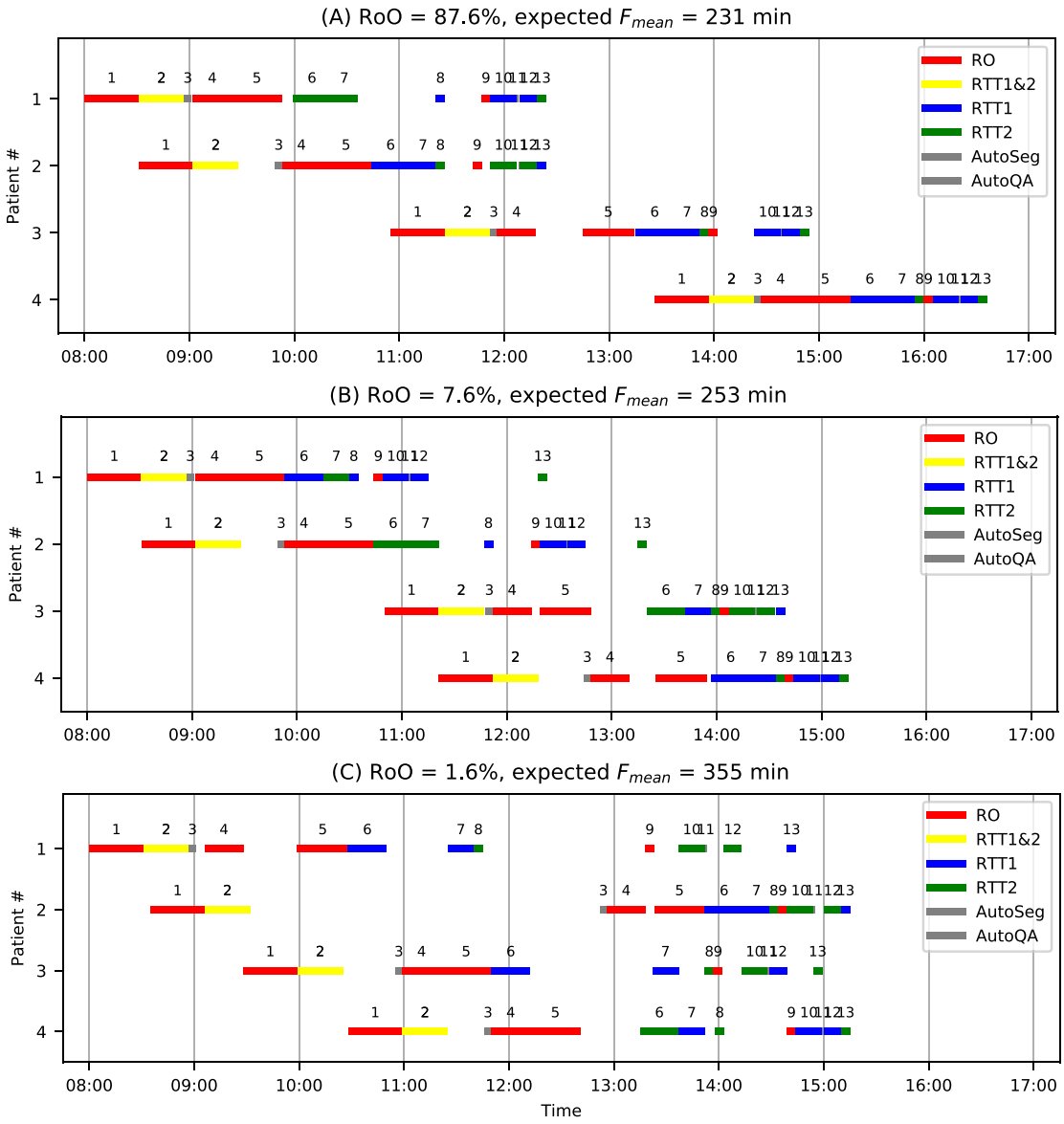

Figure 5. Example of 3 schedules at the pseudo-Pareto front of the 445th generation of the NSGA-II algorithm in the OSS-setting with 4 patients and 1 RO. The upper chart (A) shows the shortest expected F_{mean}, (B) a balanced expected F_{mean} and RoO, and (C) the lowest RoO. Schedule A has the smallest F_{mean}, but the highest risk of overtime. There are two reasons. First for schedule A the last patient is finished later than for schedule B and C. Second, a small F_{mean} is achieved by scheduling the different steps per patient close to eachother, with hardly any idle time. Because of the lack of idle time there is hardly any compensation for tasks that take longer than expected. So, in case one or more tasks take longer than expected, there is a large risk of overtime. With schedules B and C the last patients is finished earlier than for schedule A, and there is more idle time, which leads to larger F_{means} but also more compensation in case of unexpected delays. And therefore a reduced risk of overtime.

mutations M was a random number between 1 and 5, random cross-over, and stopping criteria threshold $X = 50$.

To explore the possibilities of our NSGA-II algorithm, simulations were run for various OSS-settings. First, the OSS-setting of 4 patients and 1 RO was run for 2000 generations to test the convergence of the model and the choice of $X = 50$ generations for the stopping criteria. In addition, as genetic algorithms in general are prone to get stuck in local minima, the sensitivity for the described NSGA-II algorithm was assessed by repeating this setting four more times. Second, the same setting was run for varying lengths of the preparation time (i.e.: shifts) from 8:00–17:30, 8:00–17:00, 8:30–17:00, 8:30–16:30, 9:00–16:30 and 9:00–16:00. Here, lunch break time windows were fixed between 12:30–13:30. Note that the finish time threshold for *RoO* calculation was set according to the length of the working shifts. In the third set of simulations 6 different OSS-settings are evaluated: 3 patients with 1 and 2 ROs, 4 patients with 1 and 2 ROs, and 5 patients with 1 and 2 ROs, with working shift length and break time windows kept constant like in the first experiment. The optimized schedules were compared to the clinical schedule at the time of the study.

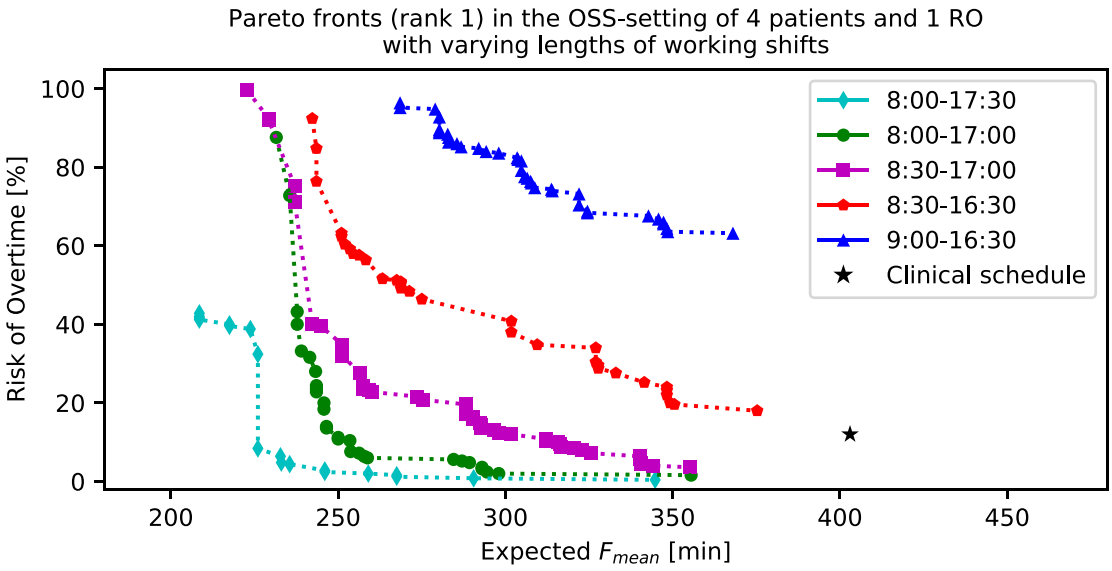

Figure 6. Pseudo-Pareto fronts of schedules for various lengths of working shifts. The thersholds for RoO calculation changed according to the lengths of the shifts. The lunch break time windows were fixed between 12:30 and 13:30. Note that the same OSS-settings are shown in green in this figure and in figure 4.

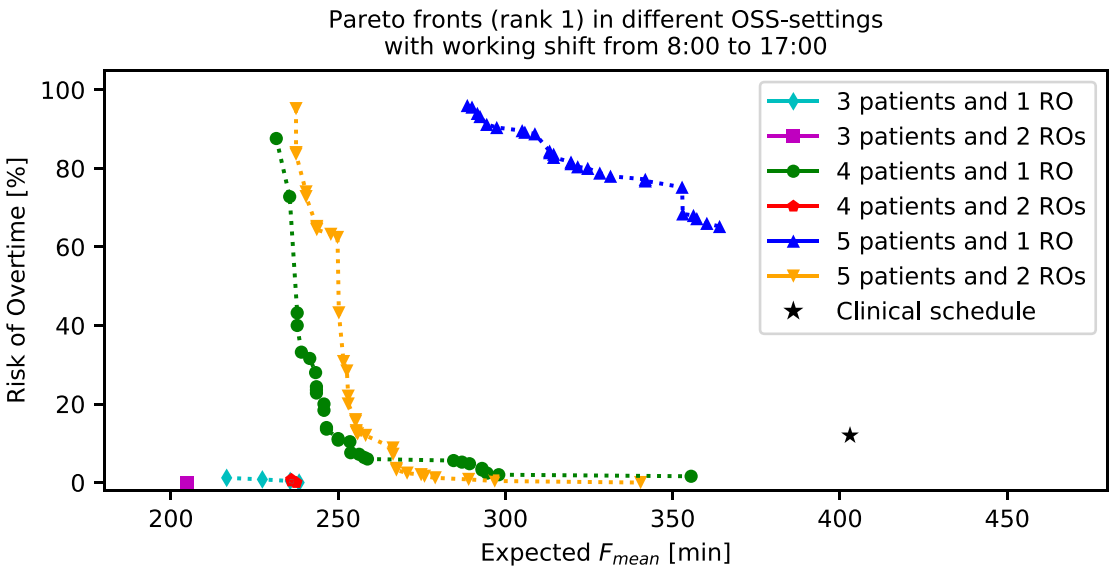

Figure 7. Pseudo-Pareto fronts of schedules for different OSS-settings. Here, the working shift length, lunch break window, and the number of RTTs are fixed. The fronts for 3 patients with 1 and 2 ROs, and for 4 patients and 2 ROs only consist of a few points, indicating that for these OSS-settings the working length shift with corresponding finish time threshold for the RoO calculations are easily met, even for short expected F_{mean}. Note that the same OSS-settings are shown in green in this figure, as in figures 5 and 6.

3. Results

3.1. Execution times

Table 8 shows the results of the Kolmogorov-Smirnov test based on the data of 663 retrospectively analysed patients, which was used to find the probability distribution of each task's execution time, used for input of $MILP_{Fmean}$ and LP_{RoO}.

3.2. Experiment 1—convergence of the NSGA-II algorithm

Figure 4 shows the pseudo-Pareto fronts of schedules from the different generations of the NSGA-II algorithm and the clinical schedule from figure 1, denoted as a star (expected $F_{mean} = 403$ min, $RoO = 12.0\%$). Note that all schedules led to a decrease in expected F_{mean} compared to the clinical schedule and often also to a smaller RoO.

For instance, schedule B (see also figure 5) indicated by the green arrow, outperforms the clinical schedule by 150 min with also a reduction in *RoO* of 4.4%.

The 445th generation (green) was the first generation that met the stopping criteria. Between generation 445th and 1000th and between the 1000th and the 2000th generation slight changes in the front are observed, but only for non-clinically relevant schedules (*RoO* above 25%). This indicates that the chosen stopping criteria were indeed suitable. Within this range, the described NSGA-II algorithm showed good robustness against getting stuck in unfavorable local minima (see appendix A.2).

From the pseudo-Pareto front of the 445th generation, Gannt charts of three schedules (indicated by green arrows in figure 4) are shown in figure 5: (A) the shortest expected F_{mean} (expected $F_{\text{mean}} = 231$ min, 87.6% *RoO*), (B) where expected F_{mean} and *RoO* are balanced (expected $F_{\text{mean}} = 253$ min, 7.6% *RoO*), and (C) the lowest *RoO* (1.6% *RoO*, expected $F_{\text{mean}} = 355$ min).

Note that a short expected F_{mean} with little waiting time from a patient perspective has the highest *RoO*, while a very low *RoO* can lead to more idle time for the executers and (unnecessary) waiting time.

3.3. Experiment 2—sensitivity to the length of the working shifts

Experiment 2 explored the effect of the length of the working shift of *RoO* and expected F_{mean} . Results are shown in figure 6. The stopping criterion was met after 1118, 445, 1529, 767, and 1446 generations, respectively. After 250 generations, there were still no feasible results for shifts from 9:00 to 16:00, therefore this run was aborted and excluded from further analysis. Interestingly, by increasing the shift with 30 min, from 8:00–17:00 to 8:00–17:30, the expected F_{mean} decreased 57 min (from 289 to 232 min), for an equal *RoO* of 4.8%. The according schedules are shown in figure A.1 in appendix A.3.

3.4. Experiment 3—evaluation of varying number of patients and staff composition

Here, the NSGA-II algorithm was deployed to explore the feasibility of various OSS-settings by changing the number of patients and available ROs, with a working shift from 8:00 to 17:00. The stopping criteria were met after 118, 8, 445, 102, 1047, and 921 generations, respectively. Figure 7 shows the corresponding pseudo-Pareto fronts. Compared to the 4 patients and 1 RO setting, the setting with 5 patients and a second RO showed an equivalent pseudo-Pareto front. The two curves intersect around $F_{\text{mean}} = 267$ min and $RoO = 5.9\%$, with a lower *RoO* for the setting of 4 patients and 1 RO before the intersection and a lower *RoO* for the setting of 5 patiets and 2 ROs after. Treating 3 patients instead of 4 with one RO led to a gain in expected F_{mean} of 139 min at an *RoO* of 1.2% and 1.6%, respectively. For the OSS setting with 5 patients and 1 RO (figure 7, blue triangles) the NSGA-II algorithm found only schedules with an *RoO* higher than 65%.

4. Discussion

In this study we presented a method that solves the scheduling problem for radiotherapy OSS outpatient clinics. The method and results can support RT management by providing insights in the trade-off between the average time patients spend in the hospital, the risk of overtime, various choices of the length of the working shift, the number of patients treated on a single day and the staff composition. The optimization code is made publicly available (Hoffmans-Holtzer 2022). As the code allows for configuration of tasks, executers and durations, it can be used to explore various OSS-settings. We showed that by solving the scheduling problem, for a standard setting of 4 patients per day and 1 RO, the best schedule leads to a decrease in expected F_{mean} of 153 min compared to the clinical schedule which was manually created. This decrease was without any costs in terms of robustness (12.0% for clinical versus 11.2% for optimized schedule), working times, number of patients treated or staff composition.

This observation shows that, by smartly scheduling the order of tasks, large workflow improvements can be observed and therefore it strongly supports the further exploration of operations research (OR) for further optimization of the radiotherapy preparation process. Additionally, the developed scheduling approach provided valuable business information regarding the underlying trade-offs between patient flow time, risk of overtime, the number of patients treated, the consequences of changing staff composition and the effect of changing working hours.

The first experiments shows a clear trade-off between F_{mean} and *RoO,* that may seem counterintuitive: the shorter the F_{mean} the higher the risk of overtime. Figure 5 shows that the schedule with the smallest F_{mean} (schedule A) is the schedule for which the working day takes longest. As there is little margin between the different tasks, a delay for a certain task and certain patient cannot be compensated for downstream and the schedule will go into overtime. With schedule C on the other hand, an unexpected delay in e.g. task 6 of patient 1 will not affect the tasks downstream, since there is some margin between the next task the operators need to do.

The margin leads to a higher F_{mean}, but at the same time allows for the compensation that is needed to reduce the *RoO*.

The second experiment demonstrated the somewhat unintuitive finding that a relatively small change in working shift length of 30 min could lead to a reduction of the expected F_{mean} of approximately 1 h. Please note here, that for a specific sequence a change of the working shift length not only changes the *RoO*, but also influences the optimal solution of $MILP_{Fmean}$ in terms of executer and break time selection, potentially leading to new (pseudo)-Pareto fronts. Therefore a relatively small change in working shift length can lead to a large Pareto front change; for example, for an *RoO* of 4.8% a reduction of the expected F_{mean} of approximately 1 h could be achieved for the 8:00–17:30 shift compared to the 8:00–17:00 shift. This underlines the potential of mathematical optimization of many schedules and the exploration of estimated Pareto fronts for different settings, in contrast to manual optimization of a limited set of schedules.

The third experiment showed that schedules with short expected F_{mean} and low *RoO* exist for the OSS-settings of 3 patients with 1 and 2 ROs and 4 patients with 2 ROs, i.e.: for these settings the working shifts could be reduced. Also, the low number of generations needed for these settings suggests that the NSGA-II algorithm was not challenged to find more optimal schedules. The setting of 5 patients and 1 RO appeared not achievable in clinical practice because of the large *RoO*. The presence of one additional RO (and an equal number of 2RTTs) would only result in a modest benefit of one additional patient that could be included, with similar F_{mean} and *RoO*.

With the experiments described above, we demonstrated the feasibility and potential gain of the application of OR-tools for optimizing the pretreatment preparation workflow in the setting of an OSS for palliative radiotherapy. The use of the genetic algorithms and (MI)LPs are not new in the field of radiotherapy and health care scheduling (Zhou *et al* 2018, Vieira *et al* 2020). However, to the best of our knowledge this is the first study that combines both methods to optimize the radiotherapy treatment preparation process, while accounting for the uncertainties in execution times that are inherent to real world clinical applications.

Even though the described model is tailored to our specific clinical constraints, generalization to other centers is relatively straightforward with the code that is publicly made available (Hoffmans-Holtzer 2022). Center-specific information such as the process flow of the patient, the processing times, and which expert performs which task is included in the algorithm as input information through either parameters or index sets and can hence be easily adapted. Some centers may have specific restrictions, e.g. due to specific treatment machines or systems, that can be taken into account easily by adding corresponding constraints to $MILP_{Fmean}$. Besides other radiotherapy OSS-settings, such a setting may occur in other workshop-like environments where expert staff can be assigned to multiple tasks (e.g. as in a dermatology clinic (Romero *et al* 2013)), or where dedicated machines are used for some of the tasks, while employees carry out multiple other manual tasks. Therefore, the OR-tools for optimizing the pretreatment preparation workflow inside and outside radiotherapy and could be of value for mitigating the increasing gap between supply and demand for radiation oncology and beyond.

This study had a couple of limitations that need to be addressed. First, the stochastic nature of the processing time per task was only accounted for in the *RoO* calculation but not in the expected F_{mean} optimization. Therefore, many schedules with the same optimal expected F_{mean} could be computed, but with different *RoO*. Nevertheless, the survival of the fittest approach by the NSGA-II takes this into account, by filtering out schedules with unfavorable idle time windows. Second, a disadvantage of genetic algorithms are that they could get stuck in local minima. However, for the settings used in this study (mutation rate, stopping criteria, size of the mating pools) the reproducibility was sufficient (appendix A.2). Third, the $MILP_{Fmean}$ optimizes the expected F_{mean}, but not the idle time for executers. This leads to efficient schedules, though the idle time between tasks of the same executer may be perceived an inefficient from the executers' perspective. In fact the idle time allows for an efficient schedule, because it allows for compensation of unexepected delays that would have led otherwise to a high *RoO*. Although the schedules were not directly mathematically optimized for idle time for executers, the algorithm will only generate schedules with balanced F_{mean} and *RoO*, and balanced idle time for the executers simultaneously. Fourth, the algorithm requires a lengthy computation of a couple of hours up to weeks depending on the length of the working shift and the staff composition. For the OSS-setting this is not a problem since the algorithm is only used for planning at the tactical level and hence needs to be run only once. However, to upscale to settings with more patients or staff, it would become a challenge, and possibly other optimization algorithms would be a better fit. With upscaling to more patients and staff, the limited speed of the $MILP_{Fmean}$ will be the main bottleneck as it will become too slow to handle the complexity caused by increased numbers of patients, tasks and executers (staff). Moreover, the genetic algorithm will need more iterations to converge to the estimated Pareto fronts, which will increase the number of times the $MILP_{Fmean}$ needs to run. In the future, this could be overcome by implementation of machine learning techniques like neural networks instead of the (MI)LPs to feed the genetic algorithms, in combination with stochastic programming.

Before upscaling, however, the approach is currently being tested in the clinical OSS-setting, to investigate whether the potential gains described by the simulations in this article are indeed robust enough against clinical practice.

5. Conclusion

To anticipate to the expected increase in cancer incidence and staff shortage, operations research tools were investigated that could optimize deployment of radiotherapy resources. For sensible application in clinical practice, these tools need to be robust against the patient-to-patient variation in the duration of tasks. This study describes a method that optimizes the radiotherapy pretreatment preparation schedule of the one-stop shop outpatient clinic for palliative radiotherapy. Based on this method, the expected average pretreatment preparation time of patients at the one-stop shop could be reduced with 37%, without increasing the risk of delays or additional costs. The experiments provided insights in the trade-offs between pretreatment preparation time, risk of working overtime, working shift length, number of patients treated on a single day and staff composition. Therefore, this approach could be of value to support tactical and strategic decision making. The results strongly support further exploration of scheduling optimization for radiotherapy preparation also outside a one-stop shop or radiotherapy setting.

Acknowledgments

Part of this research was funded by Elekta (Elekta AB, Stockholm, Sweden) and the Dutch Research Council (VI. Veni.192.043, NWO). The department of radiotherapy also has a research collaboration with Accuray Inc., Sunnyvale, CA, USA, Varian, a Siemens healthineers company, Palo Alto, CA, USA, and Vision RT, London, UK.

The authors gratefully acknowledge Nathalie Swart, Olijn Tims, Ilse de Pree and Cleo Slagter for their input regarding the clinical schedule and execution times. We also thank René Peeters for the valuable discussions regarding classical scheduling problems.

Data availability statement

The data that support the findings of this study are openly available at the following URL: https://github.com/NienkeHoltzer/one-stop-shop_scheduling.

Appendix

A.1. Determining sample size for RoO-matrix

This experiment describes the choice of the number of samples for adequate *RoO* sampling. First, a large $13 \times 10\,000 \times 4$ *RoO-matrix* is built with processing times that are sampled from the Kolmogorov-Smirnov distributions. Based on this matrix, the clinical schedule in figure 1 is evaluated for 10 000 simulations, resulting in 10 000 different durations of the complete flow. Table A.1 shows the results for different sample sizes; the mean, 25%, 50% and 75% percentile are approximately the same when using 250 up to 10 000 samples. Based on these findings, an *RoO* sample size of 250 is presumed to be sufficient and therefore used in all experiments.

A.2. NSGA-II algorithm robustness against unfavorable local minima

This experiment investigates sensitivity of the NSGA-II algorithm for getting stuck in unfavorable local minima. Figure 8 shows that the when the NSGA-II is run for 5 times for the same setting and with constant parameters (population size $N = 200$, number of mutations $M =$ a random number between 1 and 5, random cross-over and stopping criteria threshold $X = 50$), all runs show higly comparable results, especially for the clinically relevant schedules (*RoO* below 25%).

A.3. Comparison of working shifts 8:00–17:30 and 8:00–17:00 with 4.8% RoO

Figure A.1 shows schedules for shifts from 8:00–17:00 (A) and 8:00–17:30 (B). Both schedules are at the estimated Pareto front and have an *RoO* of 4.8%. The expected F_{mean} decreases with approximately 1 h (from 289 to 232 min) when the shift length is increased by 30 min from 8:00–17:00 to 8:00–17:30.

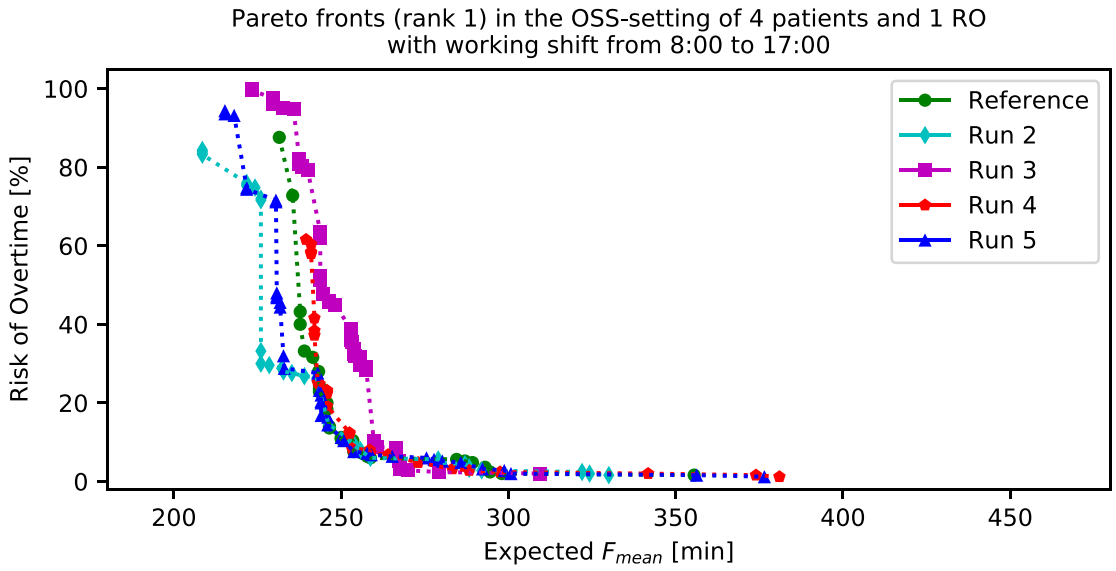

Figure 8. Pseudo-Pareto fronts of schedules for the same setting as evaluated in experiment 1. The experiment presented in figure 4 is repeated four times to investigate how well the genetic algorithm converges. In all cases the genetic algorithm reached similar solutions in the (clinically relevant) range with a risk of over time below 15%. Note that the Reference run is shown in green in this figure and is the same as in figures 5, 6 and 7.

Table A.1. Results of the evaluation of the duration of the working day for the schedule of the clinical example shown in figure 1 by LP_{RoO} for different sample sizes. The selected sample size of 250 is highlighted in grey.

Sample size [#]	Mean (min)	Std (min)	Min (min)	25% percentile (min)	50% percentile (min)	75% percentile (min)	Max (min)
50	511	30	441	495	512	528	597
100	514	34	432	490	509	533	607
250	509	31	432	488	508	528	607
500	510	29	429	489	508	528	607
1000	510	30	405	489	509	529	607
2500	510	30	405	487	509	529	634
10 000	510	30	405	489	508	530	634

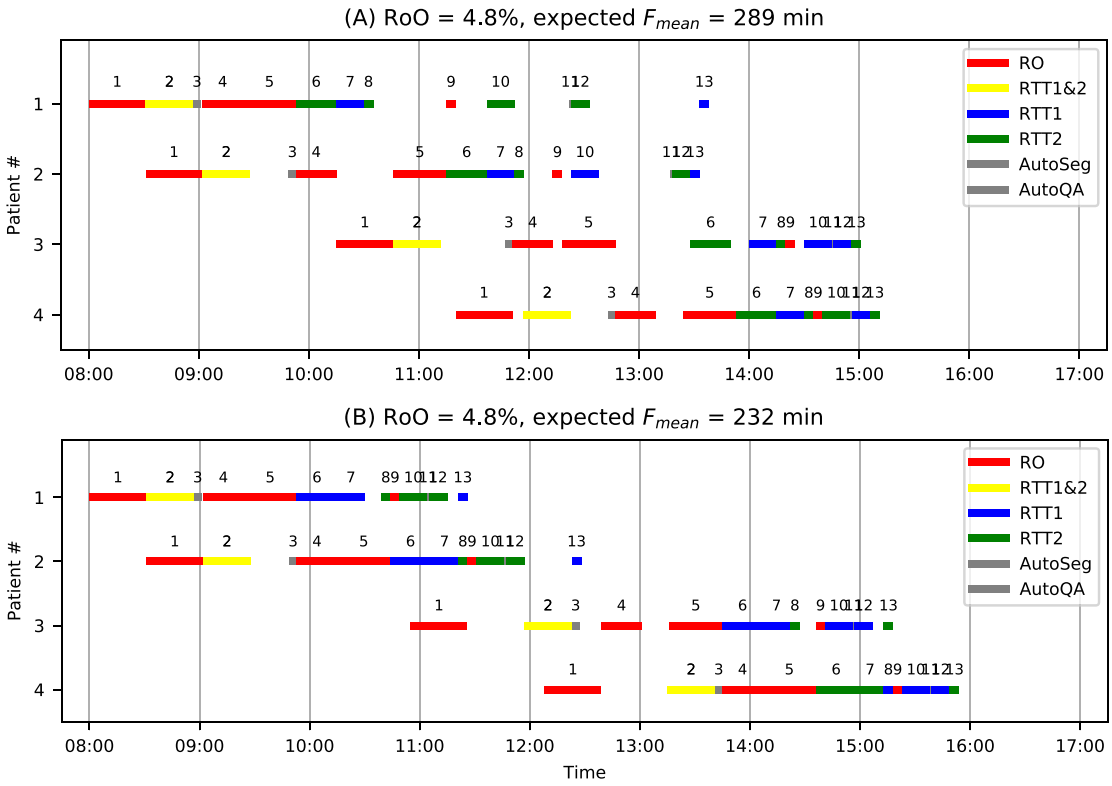

Figure A.1. Schedules for shifts from 8:00–17:00 (A) and 8:00–17:30 (B). Both schedules are at the estimated Pareto front and have an RoO of 4.8%.

ORCID iDs

Nienke Hoffmans-Holtzer https://orcid.org/0000-0001-8775-1963
Luuk Smolenaers https://orcid.org/0009-0009-2863-3903
Mischa Hoogeman https://orcid.org/0000-0002-4264-9903
Marleen Balvert https://orcid.org/0000-0002-2376-9301
Steven Petit https://orcid.org/0000-0001-9787-8239

References

Braune R, Gutjahr W J and Vogl P 2022 Stochastic radiotherapy appointment scheduling *Central Eur. J. Operations Res.* **30** 1239–77

Deb K, Pratap A, Agarwal S and Meyarivan T 2002 A fast and elitist multiobjective genetic algorithm: NSGA-II *IEEE Trans. Evol. Comput.* **6** 182–97

Di Tella M, Tesio V, Bertholet J, Gasnier A, Del Portillo E G, Spalek M, Bibault J-E, Borst G, Van Elmpt W and Thorwarth D 2020 Professional quality of life and burnout among medical physicists working in radiation oncology: the role of alexithymia and empathy *Phys. Imaging Radiat. Oncol.* **15** 38–43

ECIS 2022 Estimated incidence by year: summary [Online]. Available: (https://ecis.jrc.ec.europa.eu/explorer.php?$0-4$1-AE28E$4-1,2$3-0$6-0,85$5-All$7-7$21-0$2-All$CLongtermChart3_1$X0_14-$CLongtermChart3_2$X1_14) [Accessed Dec 13th 2022]

Franco P, Tesio V, Bertholet J, Gasnier A, Del Portillo E G, Spalek M, Bibault J-E, Borst G, Van Elmpt W and Thorwarth D 2020a Professional quality of life and burnout amongst radiation oncologists: the impact of alexithymia and empathy *Radiother. Oncol.* **147** 162–8

Franco P, Tesio V, Bertholet J, Gasnier A, DEL Portillo E G, Spalek M, Bibault J-E, Borst G, Van Elmpt W and Thorwarth D 2020b The role of alexithymia and empathy on radiation therapists' professional quality of life *Techn. Innov. Patient Support Radiat. Oncol.* **15** 29–36

Gen M, Tsujimura Y and Kubota E 1994 Solving job-shop scheduling problems by genetic algorithm *Proc. of IEEE Int. Conf. on Systems, Man and Cybernetics* (IEEE) pp 1577–82

Hoffmans-Holtzer N, Smolenaers L, Olofsen-Van Acht M, Hoogeman M, Balvert M and Petit S 2022 Python code for Robust Optimization (GitHub, Inc) Available: (https://github.com/NienkeHoltzer/one-stop-shop_scheduling)

Jones J A, Lutz S T, Chow E and Johnstone P A 2014 Palliative radiotherapy at the end of life: a critical review *CA: Cancer J. Clin.* **64** 295–310

Legrain A, Fortin M-A, Lahrichi N and Rousseau L-M 2015 Online stochastic optimization of radiotherapy patient scheduling *Health Care Manag. Sci.* **18** 110–23

Lichter K E, Anderson J, Sim A J, Baniel C C, Thiel C L, Chuter R, Collins A, Carollo E, Berg C D and Coleman C N 2022 Transitioning to environmentally sustainable, climate-smart radiation oncology care *Int. J. Radiat. Oncol. Biol. Phys.* **113** 915–24

O'Connor P 2018 Python Code for Determining Pareto Efficiency (GitHub, Inc) Available: https://github.com/QUVA-Lab/artemis/blob/4bf2fe0cd390219c2399e2256c8fbc1aff16fa85/artemis/general/pareto_efficiency.py

Petrovic D, Castro E, Petrovic S and Kapamara T 2013 Radiotherapy scheduling *Automated Scheduling and Planning* (Berlin, Heidelberg: Springer) (https://doi.org/10.1007/978-3-642-39304-4_7)

Petrovic D, Morshed M and Petrovic S 2011 Multi-objective genetic algorithms for scheduling of radiotherapy treatments for categorised cancer patients *Expert Syst. Appl.* **38** 6994–7002

Romero H L, Dellaert N P, Van Der Geer S, Frunt M, Jansen-Vullers M H and Krekels G A M 2013 Admission and capacity planning for the implementation of one-stop-shop in skin cancer treatment using simulation-based optimization *Health Care Manag. Sci.* **16** 75–86

Saure A, Patrick J, Tyldesley S and Puterman M L 2012 Dynamic multi-appointment patient scheduling for radiation therapy *Eur. J. Oper. Res.* **223** 573–84

The SciPy community 2020 *scipy.stats.kstest; SciPy v1.8.0 Manual* [Online]. Available: (https://docs.scipy.org/doc/scipy//reference/generated/scipy.stats.kstest.html#scipy.stats.kstest) [Accessed Sept 21th 2023]

Tiwana M S, Barnes M, Kiraly A and Olson R A 2016 Utilization of palliative radiotherapy for bone metastases near end of life in a population-based cohort *BMC Palliative Care* **15** 1–5

Vieira B, Demirtas D, Van De Kamer B , J, Hans E W and Van Harten W 2019 Improving workflow control in radiotherapy using discrete-event simulation *BMC Med. Inform. Decis. Making* **19** 1–13

Vieira B, Demirtas D, Van De Kamer J B, Hans E W and Van Harten W 2018 A mathematical programming model for optimizing the staff allocation in radiotherapy under uncertain demand *Eur. J. Oper. Res.* **270** 709–22

Vieira B, Demirtas D, Van De Kamer J B, Hans E W, Jongste W and Van Harten W 2021 Radiotherapy treatment scheduling: implementing operations research into clinical practice *Plos one* **16** e0247428

Vieira B, Demirtas D, Van De Kamer J B, Hans E W, Rousseau L-M, Lahrichi N and Van Harten W H 2020 Radiotherapy treatment scheduling considering time window preferences *Health Care Manag. Sci.* **23** 520–34

Vieira B, Hans E W, Van Vliet-Vroegindeweij C, Van De Kamer J and Van Harten W 2016 Operations research for resource planning and-use in radiotherapy: a literature review *BMC Med. Inform. Decis. Making* **16** 1–11

WORLD HEALTH ORGANIZATION 2016 Annex 1: Health workforce requirements for implementation of the Global strategy on human resources for health *Global strategy on human resources for health: workforce 2030* (Geneva, Switserland: WHO Document Production Services) 46ISBN: 978 92 4 151113 1

WORLD HEALTH ORGANIZATION 2022 1.3 Assessing the impact of the COVID-19 pandemic, 2020–2021 *World health statistics 2023: monitoring health for the SDGs, Sustainable Development Goals* (Geneva, Switserland: WHO Document Production Services) 17–23 978-92-4-007432-3

Zhou L, Geng N, Jiang Z and Wang X 2018 Multi-objective capacity allocation of hospital wards combining revenue and equity *Omega* **81** 220–33