LUND UNIVERSITY

Master of Science Thesis
Niklas Eliasson

Automation of robust Pareto front-based radiotherapy treatment planning for prostate cancer patients

Niklas Eliasson

Supervisors

Hunor Benedek, Andre Haraldsson, Per Engström, Tommy Knöös and Crister Ceberg

Department of Medical Radiation Physics,
Faculty of Science
Lund University
www.msf.lu.se

1 Svensk populärvetenskaplig sammanfattning

Antalet rapporterade cancerfall i Sverige har mer än dubblerats sedan 1970-talet. Cancerfonden rapporterade att 61100 personer blev diagnostiserade med cancer under 2015 och att prostatacancer är den mest förekommande typen av cancer. Ett vanligt sätt att behandla prostatacancer är extern strålbehandling, där en strålkälla (vanligtvis en linjäraccelerator) som befinner sig utanför kroppen används för att bestråla tumören med joniserande strålning.

För varje patient skapas en individuell behandlingsplan, vilken innehåller instruktioner för hur behandlingsmaskinen skall leverera behandlingen. Då planen skapas definieras mål för hur mycket olika områden skall eller får bestrålas. Syftet för målen är att säkerställa en bra behandling samtidigt som närliggande organ skyddas. Till exempel bör tumören bestrålas så mycket som läkarna har ordinerat samtidigt som bestrålningen av känsligare organ som till exempel ändtarmen hålls låg. Genom att bestråla känsliga områden så lite som möjligt och stäva efter att minst uppnå de kliniska målen så minskas risken för de negativa konsekvenser som är kopplade till att bestråla området. De satta målen konkurrerar ofta med varandra och planerarna får finna en balans mellan rätt bestrålning av tumören och så lite bestrålning som möjligt till de känsligare områdena i närheten av tumören.

För de mer avancerade behandlingsmetoderna används inversdosplanering för att skapa en behandlingsplan. Då ger dosplaneraren sina mål för fallet till ett datorprogram som används för att skapa behandlingsplanen. Programmet använder en optimeringsalgoritm för att finna en optimal behandlingsplan baserat på de mål som satts och viktningen mellan målen. För att påverka vilken optimal behandlingsplan som beräknas fram av programmet så modifierar dosplaneraren de givna målen och hur viktiga de ska vara.

Det är vanligt att målen konkurrerar med varandra vilket kräver att en balans mellan målen hittas. För att finna de bästa möjliga behandlingsplanerna kan man söka efter Pareto-optimala planer. En behandlingsplan är Pareto-optimal om det inte går att förbättra ett av målen utan att försämra ett av de andra målen. Tillsammans kallas alla Pareto-optimala planerna en Pareto-font. Planerna i en Pareto-fronten är inte optimal för alla mål utan de planer som är bättre för ett mål blir sämre för ett annat mål.

Dessvärre är det tidskrävande att skapa och samla in information om många behandlingsplaner, men hela processen går att automatisera med hjälp av programmering. Ett program som efterliknar mänsklig planerare och följer en inversdosplaneringprocess kan skapas. Programmet använder därefter information från de skapade behandlingsplanerna för att bestämma vilka av dem som är Pareto-optimala i förhållande till de satta målen. Därefter kan en front av behandlingsplaner visas upp för att få en bättre överblick över konkurrensen mellan målen.

Skillnader mellan behandlingstillfället och tillfället för skapandet av behandlingsplanen kan uppstå på grund av prostatans rörelse i kroppen. Detta påverkar behandlingen, som då inte utförs som planerat. Genom att undvika behandlingsplaner som påverkas mycket av rörelsen så kan rörelsens påverkan minimeras. Hur mycket olika behandlingsplaner påverkas kan testas genom att efterlikna rörelsen i datorn då behandlingsplanen skapas och redan då utesluta planer som påverkas för mycket av prostatans rörelse.

Automatisering av dosplaneringsprocessen ger många fördelar. Arbetet som krävs för att utvärdera behandlingsplaner med en front av Pareto-optimala planer blir mindre. De skapade behandlingsplanerna kommer alltid få samma kvalité, vilket inte påverkas av vilken dosplanerare som skapar behandlingsplanen eller hur lång tid dosplaneraren lägger på att skapa behandlingsplanen. Dessutom kan tillägg skapas till programmet vilket ger möjligheten att testa behandlingsplanerna för påverkan av prostatans rörelse, eller lägga till andra tillägg som maskininlärning vilket hjälper programmet finna de bättre lösningarna snabbare.

2 Abstract

Purpose/objective

The main objective was to create a software that automates the process of creating treatment plans used for Pareto front-based dose planning for prostate cancer patients. A second objective/purpose was to add a robustness test to this program to evaluate the effect of prostate movements on the treatment plans.

Material/method

An IronPython program was designed to control and collect information from the treatment planning system (TPS) RayStation 5 and by using built-in libraries made by the creators of the Raystation 5.

The patients selected were prostate cancer patients with treatment of only the prostate, not including nearby lymph nodes or seminal vesicles. Hypo-fractionation treatment plans were created with a prescription of seven fractions of 6.1Gy with a total dose of 42.7Gy, one fraction every other day.

A comparison was made between a treatment plan created by a dose planner and Pareto fronts extracted from 1440 treatment plans automatically generated with the plan generation program.

The robustness test was evaluated on one patient by using an isocenter shift of (x, y, z) = (0.45cm, 0.02cm, -0.58cm).

Result

The software consisted of two programs. The first program used the optimizer and dose calculator in RayStation 5 to create deliverable treatment plans. It inputted different objective functions and/or weights to the optimizer. For each change, the optimizer would find a new optimal treatment plan. It saved DVH-data for evaluation of the plans. A built-in robustness test was added to the program to test the effect of prostate movements on the treatment plans it constructed. It moved the finished treatment plan's isocenter and the dose difference was calculated.

The second program, created for evaluation, loaded the saved data from the program generating the treatment plans. In the program, multiple data sets was loaded and compared. It visualized both the Pareto fronts based on the collected plans and all the dose volume histograms (DVHs)

for one plan at a time (for a selected plan from the first graph). The program could determine if treatment plans were Pareto optimal and/or clinically acceptable. It was also used to visualize the robustness test, where the static treatment plans and treatment plans with a moved isocenter were plotted in the same graph.

The programs automated the process and reduced the work needed to only some preparation before starting the program. The time to create treatment plans for a Pareto front was greatly reduced, as the program could save one plan every 2-3 minutes.

When comparing a Pareto front consisting of automatically generated treatment plans and a plan created by a planner, the dose planner's treatment plan ended up near the Pareto front in all cases.

Discussion

Prostate cancer was selected for this study due to the fact that it is a comparably simple case involving only a few OARs. The only trade-off that needed to be visualized is the one between the target and the rectum. Thus, only the rectum goal needs to be changed to be able to show the trade-off. If more OARs would be added, it would have taken longer time to generate enough treatment plans to represent the trade-offs.

There are, however, improvements that could be made. For further automation, and to decrease the generation time, machine learning could be used.

The robustness test was able to show how the dose distribution would be affected by an isocenter movement. Several improvements could be made. For instance, there are several fractions in a treatment course, and the same movement does not occur each time. There might be a continuous movement during the treatment delivery, not only one big movement. Furthermore, the data used was for a 30-minute interval, while in reality, a patient has come and gone in less than half that time.

Conclusion

A program has been developed and implemented that can be used for automation of creation of treatment plans used for Pareto front-based dose planning. A robustness test was built-in to allow for comparison between the created plans with respect to how much prostate motion would affect them. Features such as machine learning would be a good tool to further automate the process and to reduce the generation time.

3 Abbreviations

IMRT = **I**ntensity-**M**odulated **R**adiation **T**herapy

VMAT = **V**olumetric **M**odulated **A**rc **T**herapy

MLC = **M**ulti**L**eaf **C**ollimator

MU = **M**onitor **U**nits

FFF = **F**lattening **F**ilter **F**ree

ROI = **R**egion **O**f **I**nterest

CTV = **C**linical **T**arget **V**olume

PTV = **P**lanning **T**arget **V**olume

OAR = **O**rgan **A**t **R**isk

DVH = **D**ose **V**olume **H**istogram

GUI = **G**raphical **U**ser **I**nterface

WPF = **W**indows **P**resentation **F**oundation

MRI = **M**agnetic **R**esonance **I**maging

TPS = **T**reatment **P**lanning **S**ystem

4 Content

1 Svensk populärvetenskaplig sammanfattning 1
2 Abstract 3
3 Abbreviations 5
5 Introduction 8
6 Aim 10
7 Theory 11
7.1 Inverse planning for IMRT and VMAT 11
7.2 Multi-criteria optimization and Pareto optimality 11
7.3 RayStation 13
7.4 Robust treatment planning 14
7.5 Programming 15
7.5.1 IronPython 15
8 Material and method 16
8.1 Patient selection and treatment plan prescription 16
8.2 Programming 17
8.3 Computer specifications 18
8.4 Comparison with dose planner 18
8.5 Prostate movement data 19
8.5.1 Treatment plan robustness test 20
9 Results 21
9.1 The software 21
9.1.1 Treatment plan generator 21
9.1.2 Treatment plan evaluation program 23
9.1.3 Example of usage of the programs 23
9.2 Comparison with dose planner 26
9.3 Robustness test 30
10 Discussion 31
10.1 The software 31
10.2 Comparison with dose planner 33
10.3 Prostate cancer case 35
10.4 Robust comparison 35
10.5 Machine learning 36
11 Conclusions 37
12 Future prospects 37

13 Acknowledgments 39
14 References 39

5 Introduction

The number of reported cancer cases in Sweden has more than doubled since 1970. The Swedish Cancer Society reported that 61100 people were diagnosed with cancer during 2015, where 10439 of them were prostate cancer patients [1]. External beam radiotherapy can be used as a non-invasive treatment for prostate cancer [2]. The treatment attempts to irradiate the target volume with a large dose of ionizing radiation, while healthy tissue should receive an as low dose as possible. The treatment is split up into multiple fractions and a conventional treatment may consist of 39 fractions, 2Gy per fraction and 1 fraction per working day. The treatment can also be hypo-fractionated, in which case a treatment may, for instance, consist of 7 fractions, 6.1Gy per fraction and one fraction every other working day. Data from a 2-year follow-up shows that the hypo-fractionated treatment has a low incidence of side effects with no significant difference compared to the conventional treatment [3, 4].

A linear accelerator can be used to deliver external beam radiotherapy and it has several treatment techniques to deliver the treatment. Two of the more complex treatments are intensity-modulated radiation therapy (IMRT) and volumetric modulated arc therapy (VMAT). For IMRT the fields are delivered in static angles, however, each field is split into beamlets where the multileaf collimators either move during irradiation or between subfields. During a VMAT treatment the gantry rotates around the patient. While the accelerator head is rotating, machine parameters such as rotation speed, multileaf collimation and beam fluence are modified. Both techniques can deliver highly conformable treatments within a short delivery time [5, 6].

The planning process for IMRT and VMAT treatment is done by experienced dosimetrists or physicists, and the optimization part of the planning often requires a trial and error process in which the planner manually modifies the optimization parameters between each optimization to improve the treatment plan. The experience of the planner and the time spent making the treatment plan affects its quality. By automating this process, time can be saved and the created treatment plans will have a more uniform quality [7].

Voet el al. automated this process with their in-house-developed algorithm called iCycle, which can automatically generate IMRT and VMAT plans. The algorithm uses a multi-criteria optimization to create Pareto optimal IMRT plan, were the beam fluence and the beam angles are optimized. The automated VMAT planning consists of a two-step process. First, a 23-beam

IMRT plan is created, and secondly, a plan containing all constraints and objectives is used to create a template for a VMAT-plan [8, 9].

RaySearch has included a multi-criteria optimization algorithm called rayNavigator, which can create Pareto optimal VMAT treatment plans. The algorithm creates a set of Pareto optimal plans in respect to fluence from which the planner can choose. This is instead of the planner undergoing/performing the iterative process of changing the objective functions and weights. These plans have undergone a fluence based calculation along with a machine parameter optimization, which they use to minimize the DVH error compared to the final deliverable plan. Meaning that the dose distributions shown for the used it will be slightly changed when a complete the final dose distribution is performed [10].

A different way of evaluating the quality of the treatment plan is using the Pareto front with deliverable treatment plans. This will better represent the treatment delivered to the patient. The Pareto front shows the actual trade-off between underdosage to target regions of interest (ROIs) and dose to organs-at-risk (OARs) based on clinical goals. Furthermore, the Pareto front gives a broad perspective making it a good tool for comparisons between different treatment techniques, such as differences between different machines, energies or use of flattening filter or not.

The delivery of the treatment may be affected by intra-fractional movements. The prostate can move, causing a difference between the planned and delivered dose to the patient. Azcona el al [11] found that the motion magnitude varies greatly between patients, although the mean displacement was generally less than 2 mm in each of the spatial directions. The movement also varies between fractions, which was shown in a study made by Kupelian et al [12]. Amro et al [13] studied how the daily rotations and translocations of the prostate affect the dose coverage during radiotherapy. They tracked 26 patients undergoing IMRT treatment of the prostate and reconstructed the delivered dose to the prostate for different planning target volume (PTV) margins. They found that 39%, 65% and 84% of patient had an adequate clinical target volume (CTV) coverage for PTV margins of 2 mm, 3 mm and 5 mm, respectively. This implies that despite a PTV margin of 5 mm, 16% of patients received a significant underdosage of the CTV due to movement.

Since prostate movements affects the treatment dose coverage, minimizing the effects is important. One way of doing this is creating robust treatment plans. The robustness is not the same for all treatment plans and they get affected differently by movements. By performing a

robustness test on the treatment plans, the plans most affected by intra-fractional movements can be found and avoided.

6 Aim

The main objective of the present work was to create a software that automatically creates treatment plans used for Pareto front-based dose planning for prostate cancer patients. A second objective was to add a robustness test to this software to evaluate the dosimetric effect of intra-fractional prostate movements on the created treatment plans.

7 Theory

7.1 Inverse planning for IMRT and VMAT

In forward treatment planning, a dose planner modifies the beam angles, wedges and beam-weights until the treatment plan meets the dose prescription. In inverse treatment planning the planner instead changes the objectives and constraints that are given to a computer program [14]. This computer program then tries to find a treatment plan that minimise the total objectives and constraints function by using an optimizing algorithm. Examples of types of objectives and constraints are shown in *Table 1*.

Table 1: Dose prescription objectives and constraints.

Min dose	Minimum dose allowed in the entire ROI.
Max dose	Maximum dose allowed in the entire ROI.
Min DVH	Minimum dose allowed in a specific volume of the ROI.
Max DVH	Maximum dose allowed in a specific volume of the ROI.
Uniform Dose	A function that penalizes if the dose in a ROI deviates from a certain dose level.
Min EUD	A function that penalizes EUD values below a specified EUD level in a ROI.
Max EUD	A function that penalizes EUD values above a specified EUD level in a ROI.
Target EUD	A function that penalizes EUD values below and above a specified EUD level in a ROI.
Dose Fall-Off	Is like a max dose function but the dose levels boundary decreases further away from target.

A constraint is a goal for the optimizer that must be achieved if it is achievable while an objective is a prioritized goal that the optimizer should try to achieve where the priority/weight determines how important the goal is compared to the other objectives.

7.2 Multi-criteria optimization and Pareto optimality

If we have an optimization problem with an objective function $f_0(x)$, the optimal solution is found by minimizing the objective functions value (or score). A multi-criteria optimization has

a composite objective function consisting of several objective functions, where the composite objective function is a weighted sum of the objective functions. When the objectives are noncompeting, there is one optimal solution, if an optimal solution exists. It can be found by minimizing the composite objective functions values (or score). However, when the objective functions are competing, infinite many Pareto optimal solutions exists [15]. In a multi-criteria optimization, a solution is said to be Pareto optimal (or efficient) if it is not possible to improve one objective without impairing another objective [16, 17, 18].

If we have a multi-criteria optimization problem with an objective function $f_0(x)$ with the achievable scores in $\mathcal{O}$ with objective values in $\mathbb{R}^2$, then a point x^{po} is Pareto optimal if it is feasible and $f_0(x^{po})$ has the minimum score. That it has a minimum score means that it has a minimum value of the weighted sum of the objective functions. To illustrate this, imagine a cone $K \subseteq \mathbb{R}^2_+$ and $f_0(x) - K$ represents all the feasible scores that are better than or equal to $f_0(x^{po})$. A point that is both in $\mathcal{O}$ and in $f_0(x) - K$ is Pareto optimal or,

$$(f_0(x^{po}) - K) \cap \mathcal{O} = \{f_0(x^{po})\} \tag{1}$$

This is illustrated in *Figure 1* where the black line represents all the Pareto optimal solutions in $\mathcal{O}$ [15].

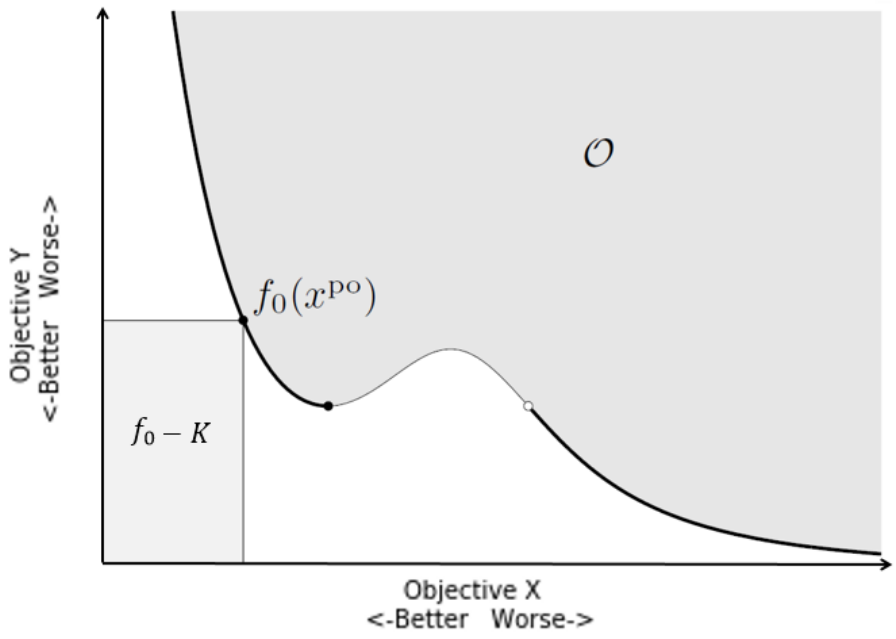


Figure 1: Illustration of an optimization problem with an objective function $f_0(x)$ with the achievable scores in $\mathcal{O}$ with objective values in $\mathbb{R}^2$. The black line on the edge of the $\mathcal{O}$ is the Pareto front. The point x^{po} where $f_0(x^{po})$ has a minimum score is Pareto optimal since there

is no point in O that can achieve a lower score in any of the directions (this can be illustrated with the grey box to the lower left of the point i.e. $f_0 - K$, since there is no part of O except for $f_0(x^{po})$ that intersects that area) [15].

This can be used for the inverse planning process in radiotherapy. This is a multi-criteria optimization problem with the objective for coverage of target competing with the objectives for sparing of the nearby OARs. The aim is to find the best suitable treatment plan for the patient. When setting up a Pareto set, Pareto optimal treatment plans with identical plan parameters for the same patient are used. The objective functions given to the TPS's optimizer are changed both regarding weights and descriptions to mimic the inverse treatment planning process of a treatment planner. If an objective function description is changed it will alter the achievable scores. It is not only the weighting factor of the weighted sum that is changes, it is the resulting solutions that become different set of Pareto curves. However, we choose to use the Pareto concept for all solutions together.

Pareto fronts can also be used for comparison between treatment techniques. This is done by constructing several Pareto fronts where single a parameter has been varied between the fronts. [17, 18].

One could also make a Pareto front comparison by using the so called clinical distance [16]. This is a clinically relevant measure for evaluating plan quality differences with the use of Pareto fronts. The measure is calculated by using a combination of clinical scaling factors, adding clinical meaning by scaling the different evaluation parameters based on their clinical importance. This also makes it possible to make comparisons between multi-dimensional Pareto fronts, as more dimensions than three would otherwise be difficult to visualize [16].

7.3 RayStation

RayStation is a treatment planning system developed by RaySearch Laboratories (Stockholm, Sweden).

In RayStation 5, both single- and multiple arc VMAT treatment plans can be created and optimized. The optimization process consists of a fluence optimization step, sequencing into machine parameters, and finally a direct optimization of machine parameters [19].

The first step of the optimization process is to define objectives and constraints, which define the desired outcome of the optimization. All objectives and constraints are connected to a

structure (ROI), such as CTV, PTV or rectum. There are multiple types of objectives and constraints as shown in *Table 1*. Each objective has a weight factor assigned.

The weight factors describe the relative importance of the included objectives.

The optimizer minimizes the value of the composite objective function, which is a linear combination of the initial objectives and constraints and the treatment plan that yields the lowest value is given as the optimal treatment plan. In the search for a better treatment plan the objective function can be changed by adding or deleting objectives, changing weights on current objectives or changing limits on the current objectives.

7.4 Robust treatment planning

The prostate is located near the bladder and rectum, as shown in *Figure 2*. Therefore, the prostate exhibits both intra- and inter-fractional movement due to differences in rectum and bladder filling.

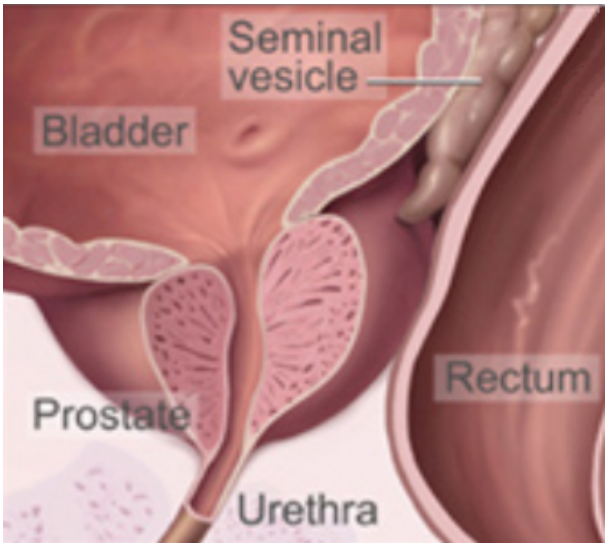


Figure 2: Illustration of the location of the prostate, close to the bladder and rectum [20].

The movements of the prostate and the region around it will affect the dose distribution and thereby the resulting treatment delivery. One could attempt to mimic the prostate movement during the treatment planning to see how it affects the dose distribution. A simple way of doing this is to move the isocenter of the treatment plan based on the prostate movement. However, a shift of the treatment plan's isocenter moves not only the prostate, it moves the entire patient relative to the treatment.

To make a realistic model one could consider that the treatment is given in multiple fractions, and that the prostate does not necessarily exhibit a similar movement pattern in each fraction. This can be adjusted for by making several isocenter movements, one for each fraction. To evaluate how the entire treatment is affected, the dose distributions for each fraction can be added together.

The prostate may also move while the treatment is being delivered. This could be adjusted for by splitting the arc in to segments and then making several smaller isocenter movements which together results in the total movement for one fraction.

7.5 Programming

7.5.1 IronPython

A feature in RayStation 5 is the ability to create scripts in the language IronPython. The scripting allows the user to control the TPS through an API that uses IronPython as scripting language.

IronPython is an open-source implementation of Python integrated with Microsoft's .Net Framework. The .Net software framework provides a way of building and deploying applications in windows.

8 Material and method

8.1 Patient selection and treatment plan prescription

The treatment plans were based on the phase III HYPO study of hypo-fractionated radiotherapy for prostate cancer, excluding the lymph nodes or seminal vesicle [3, 4]. The treatment consisted of 7 fractions, 6.1Gy per fraction and a total dose of 42.7Gy. One fraction is given every other weekday

All treatment plans used a one arc VMAT treatment technique where the gantry rotated counter clockwise one full arc from 181° to 179° with a 5-degree collimator angle. The treatments were planned with 10 MV photons, with a dose normalized to 100% covers 50% of the target volume and using PTV as target volume. The PTV margin used was 7 mm.

The evaluation of the treatment plan was done with the clinical goals used for the hypo-fractionated arm in the phase III HYPO study. In *Table 2*, the clinical goals are listed along with descriptions of the goals.

Table 2: A list of clinical goals according to the HYPO study [4].

Dose-volume objectives/constraints

Priority	Volume	Hypofractionated arm	
1	CTV	$D_{min} \geq 95\%$ $D_{min} \geq 40.6$ Gy	The minimum dose to CTV shall be greater than or equal to 95% of the prescribed dose, i.e. $D_{mean,PTV}$.
2	PTV	$V_{95\%} \geq 95\%$ $V_{40.6Gy} \geq 95\%$	The 95% isodose shall cover at least 95% of PTV.
3	Rectum	$V_{90\%} \leq 15\%$ $V_{38.4Gy} \leq 15\%$	Less than 15 % of the outlined rectal volume should receive doses greater than 90% of the prescribed dose.
4	PTV	$D_{99\%} \geq 90\%$ $D_{99\%} \geq 38.4$ Gy	The "near minimum dose" to PTV should be greater than or equal to 90% of the prescribed dose.
5	Rectum	$V_{75\%} \leq 35\%$ $V_{32Gy} \leq 35\%$	Less than 35 % of the outlined rectal volume should receive doses greater than 75% of the prescribed dose.
6	Femoral heads	$D_{max} \leq 70\%$ $D_{max} \leq 29.9$ Gy	The maximum dose to the femoral heads should be less than or equal to 70% of the prescribed dose
7	Rectum	$V_{65\%} \leq 45\%$ $V_{28Gy} \leq 45\%$	Less than 45 % of the outlined rectal volume should receive doses greater than 65% of the prescribed dose.
8	Body	$D_{max} \leq 105\%$ $D_{max} \leq 45.7$ Gy	The maximum global dose should be less than or equal to 105% of the prescribed dose

8.2 Programming

A majority of the work behind this thesis consisted of writing code. This was done in two languages, Python and IronPython. Multiple standard libraries for both languages was used (such as *Math*, *Threading* or *Os*). The software consisted of two separate program one to control Raysation 5 and one to evaluate the result.

The program controlling RayStation 5 was created in IronPython and used an API created by the developers. Methods and functions from the package can be used for the optimization or to get a dose value for a ROI at a relative volume. A GUI was create using Windows presentation foundation (WPF). To avoid the GUI from being blocked by the main thread and thus rendering it unresponsive during calculations, multithreading was used. When starting the creation and saving of the treatment plans, a new thread was created for it so that it did not run on the same thread as the GUI.

The evaluation program was written in Python 2.7. The python package Tkinter was used to create a GUI for this program. To plot data loaded by the program the python package matplotlib was used along with package Numpy to structure the data.

8.3 Computer specifications

The software run on a computer with an Intel® Xeon® processor E5-1603 (4 cores with 2.80 GHz processor base frequency) and 32-GB RAM memory.

8.4 Comparison with dose planner

A new VMAT treatment plan was created by the dose planner at the clinic. The treatment plans were created in Eclipse instead of in RayStation 5, since this was the dose planner's preferred TPS. The dose planner created a treatment plan with the parameters described in *Patient selection and treatment plan prescription* above. The treatment planner was told to create a plan as good as possible based on the clinical goals, and as if it were to be displayed on the routinely plan review conference.

The finished treatment plan was imported to RayStation 5, where a script was used to save information from the treatment plan to be loaded in the evaluation program. The script did not affect the treatment plan, it only saved data from it in the same way that the plan generation program did for the treatment plans it generated.

The plan generation program created and saved DVH-data for evaluation from 1440 treatment plans for the same patient. The process took 3 days, where each plan took around 3 minutes (started on Friday morning and was done on Monday morning). The changes made to the objective functions were the dose values, relative volumes, and weights of the three rectum objective functions (based on the three rectum clinical goals shown in *Table 2*). The generated treatment plans were compared with the one created by the dose planner in the evaluation program.

When generating 1440 treatment plans only the rectum objective functions were changed sett the value of the weights to 50-200 steps of 50 and 200-403 steps of 7. The rectum objective functions updates together, the input used for dose and volume for the three rectum objective functions is shown in table 3 and table 4.

Table 3: The dose values used for the tree rectum objective functions to create 1440 different treatment plans.

OF Number	Dose [cGy]								
	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8	Step 9
1	2989.0	2775.5	2562.0	2348.5	2135.0	1921.5	1708.0	1494.5	1281.0
2	2562.0	2348.5	2135.0	1921.5	1708.0	1494.5	1281.0	1067.5	854.0
3	2135.0	1921.5	1708.0	1494.5	1281.0	1067.5	854.0	640.5	427.0

Table 4: The volume values used for the tree rectum objective functions to create 1440 different treatment plans.

OF Number	Volume [%]				
	Step 1	Step 2	Step 3	Step 4	Step 5
1	15.0	12.5	10.0	7.5	5.0
2	20.0	17.5	15.0	12.5	10.0
3	30.0	25.0	22.5	20.0	17.5

The programs generates all combinations of the weights, dose and volume giving 1440 treatment plans.

8.5 Prostate movement data

The data for the prostate movements used in the robustness test was collected from patients at our clinic. The patient group is the same as the one for which the treatment plans were created for, however not from the same individual patients. It was collected from prostate cancer patients with treatment for only the prostate. Three gold markers had been inserted into the prostate of the patients (gold thread, 5 mm long and 1 mm in diameter). These markers are used for positioning of the patient during treatment.

The translation (and rotation) of the prostate was measured using MRI. The patients were examined with a dedicated prostate protocol. The sequences used to identify the prostate movement were in the beginning and the end of the protocol (same sequence but at different times). The time between the two sequences was 30 minutes. Between the two sequences, the

patient stayed on the examination table inside the MRI machine. The sequence used was a large T2-weighted MR image (field of view covering the body contours) with a slice thickness of 2.5 mm. These image are routinely used for contouring the target and other regions of interest, identifying the gold markers, and for dose planning.

In the images, the markers were identified manually. The two images were registered against each other based on the gold markers. The point match registration module in the Eclipse TPS was used to obtain the translation and rotation of the prostate. The coordinates in *Table* were used for the robustness test.

Table5: Coordinates for the prostate movement translations that were used for the robustness test. The coordinates were specified according to IEC 61217. The data was collected from prostate cancer patients in the same clinic.

Patient	x [mm]	y [mm]	z [mm]
1	-0.0199	-0.3997	0.1295
2	2.6686	-1.2311	-0.4585
3	1.0171	-0.9968	2.2378
4	4.5388	-5.8185	0.2078
5	-6.782	1.5125	-0.7559
6	0.0148	-2.212	1.6952
7	-1.6148	-0.9878	-2.7943
8	5.3745	2.3044	-1.6039
9	-0.2094	-0.4655	0.0205
10	0.1946	-0.8707	-0.4639

8.5.1 Treatment plan robustness test

The robustness test was evaluated on one patient by using an isocenter shift of (x, y, z) = (0.45cm, 0.02cm, -0.58cm) (IEC 61217).

9 Results

9.1 The software

A software was created that could automatically generate multiple treatment plans, save information and visualize the saved data for evaluation. The process of creating treatment plans was fully automated. However, some interaction was needed to prepare the program generating the treatment plans. This preparation required around 5-15 minutes of work.

9.1.1 Treatment plan generator

This program could only be started from the RayStation 5 scripting tab. It was used to control what actions RayStation 5 should take. The user could enter information about the treatment plan that would be created. This process was simplified with templates, which contained a set of preselected values of all the treatment parameters that were going to be used. It contained for example plan parameters such as treatment technique, modality and energy. What was required from the user was to select a patient, a template, a case, an examination name and a treatment machine. The template also had a predefined set of objective functions. These objective functions could be modified or removed, and new objective functions could be added. When the objective functions were modified, the sampled density of the treatment plans and the amount of the treatment plans created could be determined, since it was the user that decided weights and descriptions (doses, volumes, etc.) of the objective functions.

The program worked similarly to a dose planner, using inverse treatment planning and repeating it many times. It caused RayStation 5 to perform desired actions by using commands that could be found in the RayStation 5 scripting guidelines. Creating and saving treatment plans followed the steps shown in *Figure 3*. In the first step, the program created a treatment plan with the plan parameters entered by the user. The program looped over all combinations of objective function values i.e. the weights of the objective functions, the dose values of the objective functions, and the volumes of the objective functions. The objective functions could also be grouped together so that they were updated together.

The optimization was divided into two parts (40 iterations each) and a collapsed cone dose calculation was made at the end of each sequence see *Figure 3*. This was done to produce deliverable treatment plans based on realistic dose calculations. In the seventh step, the program saved data to a text file. This saved storage space and simplified for the evaluation program to obtain the information needed. It saved the parameters used to create the treatment plan, what

objective functions that was used, and DVH data both to plot DVH graphs and to evaluate the clinical goals. Each treatment plan created in a set was saved in the same text file.

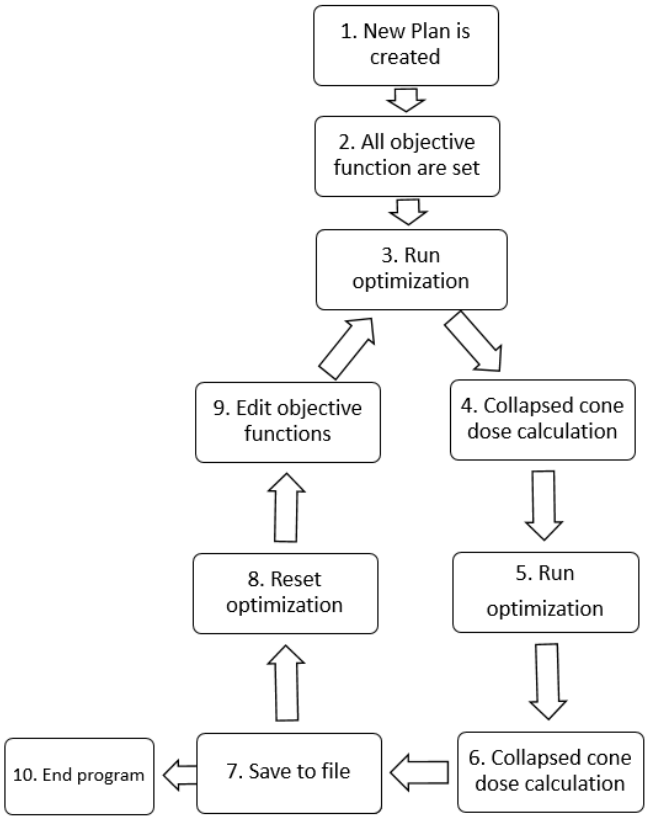


Figure 3: The process followed by the program when it creates a series of treatment plans for a patient and collects data from them.

Step 1: Create a new treatment plan based on the parameters selected.

Step 2: The program gives objective functions that should be used by RayStation 5.

Step 3: RayStation 5 starts optimization based on the objective function.

Step 4: RayStation 5 runs a collapsed cone dose calculation.

Step 5: RayStation 5 runs a second optimization.

Step 6: RayStation 5 run a second collapsed cone dose calculation.

Step 7: Data is saved to file. If all treatment plans have been created move end the program, otherwise continue creating treatment plans.

Step 8: Reset the optimization.

Step 9: Edit the objective functions by changing the weights, dose values or volumes, all based on the entries made by the user at the start of the generation. Go back to step 3.

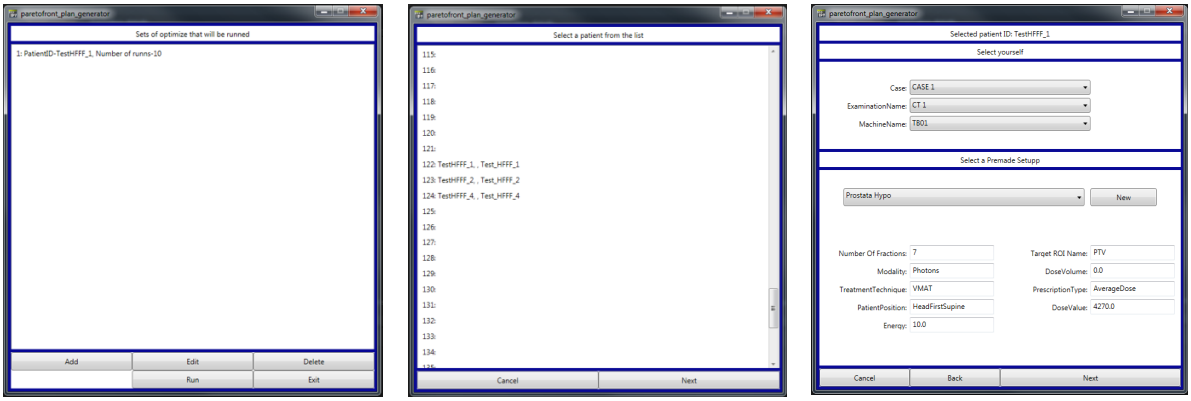
9.1.2 Treatment plan evaluation program

The evaluation program was not connected to RayStation 5. It was used to creating Pareto fronts from the data collected by the program that generated the treatment plans. It was used to load one or several saved files, giving the option of comparing multiple sets of treatment plans. For example, one could compare two sets of treatment plans with different energies. The program determines which treatment plans were Pareto optimal compared to the other treatment plans in a set. It also could determine if the treatment plans were clinically acceptable (based on the clinical goals in *Table 1*). It was also possible to change which clinical goals that were included in the test. The program had two graphs. One contained the trade-off between under-dosage to target ROIs and dose to OARs based on two clinical goals. For this graph non-Pareto optimal and/or non-clinically acceptable treatment plans could be exclude if desired. This gave the option of only showing the treatment plans that represented the Pareto front and also were clinically acceptable. What clinical goals that were represented in the graph could be changed through two dropdown menues. When a point (treatment plan) was selected in the graph, information about the treatment plan was shown below the graph, and DVHs were shown in the second graph (for the ROIs that were selected to be saved in the treatment plan generation program). The information needed to recreate the selected treatment plan could be retrieved.

The created Pareto front was in a clinical space, i.e. the solution contained treatment plans that were deliverable for the given set of plan parameters (treatment technique, energy, modality etc.).

9.1.3 Example of usage of the programs

In this section, an end-to-end usage of the program is presented. We wanted to create Pareto fronts for the patient Test_HFFF_1. To do that we start up RayStation 5. On the right side of the screen the scripting tab could be found, where the program to generate the treatment plans could be started. When running the program, the menu shown in *Figure 4 (a)* could be seen (without listed item 1). To add a new set to be generated the *Add*-button was pressed. Then a list of all the patients in RayStation 5 database was shown as in *Figure 4 (b)*. The patient Test_HFFF_1 was selected. The next step was to select plan parameters, shown in *Figure 4 (c)*. With three dropdown menus, the case, examination and treatment machine could be selected. In this test case, *CASE 1*, *CT 1* and *TB01* were selected. Thereafter a template was selected with the dropdown box in the lower section *Figure 4 (c)*. When a template was selected, the plan parameters were shown below the dropdown menu.



(a) (b) (c)

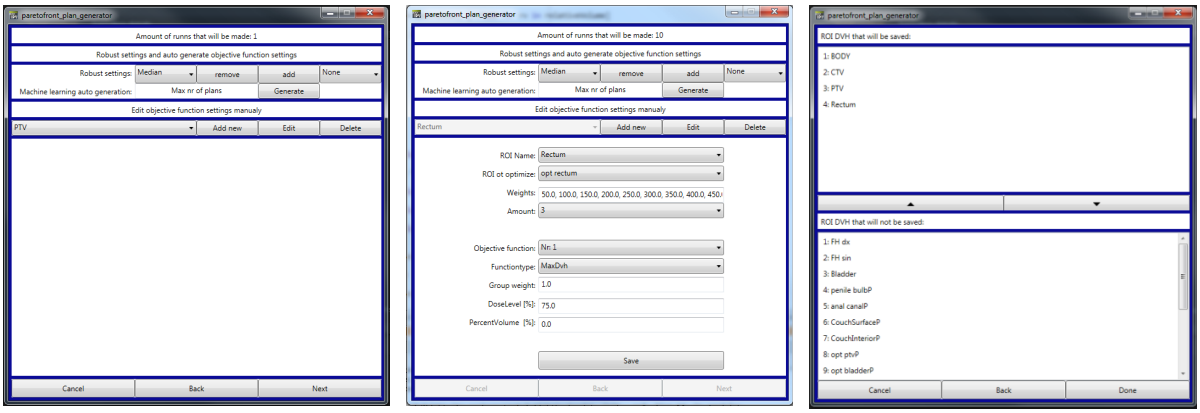
Figure 4: Images from the program that generates the treatment plans. Three section of the program is shown, the main menu (a), selection of patient (b) and selection of plan parameters (c).

The next step was to decide if a robustness test should be performed on the treatment plans. This was done on the row for robust settings as shown in *Figure 5 (a)*. The last dropdown menu on that line shows which test should be performed (in this case no test was selected). With the first dropdown menu, a prostate movement could be selected. The list contained ten prostate movement options in (x, y, z) coordinates, median of them and average of them.

In this part of the program the objective functions were modified. From the selected template, predefined objective functions were already determined. The objective functions were grouped, so that several objective functions would update together. A group could be selected in a dropdown box shown under the section, *edit objective functions settings manually,* in *Figure 5 (a).* A group of objective functions could be modified or removed. New groups could also be created. A group of objective functions for the rectum ROI is selected in *Figure 5 (b)*. To create ten treatment plans where the difference between them was the weight of the rectum objective function, weights would be added in the weight text box shown in *Figure 5 (b)*. In this case weights 50 to 500 were used with a sample distance of 50 (the weights were relative, so the weights of the other objective functions should be considered before selecting a weight). For an objective function group several other parameters could be change, such as what ROI should be used for the evaluation, what ROI should be used for the optimization, and the objective functions in a group. For each objective function in a group, the function type, group weight (relative weight compared to the weight given to the group) and descriptions (dose, volume, etc.) could be modified. If two dose values or volumes would be entered for the rectum objective

functions, then twenty treatment plans would be generated. Each weight would be used for each dose value or volume (i.e. for all combinations entered by the user).

The final step was to select what DVHs should be save for each plan. In *Figure 5 (c) BODY*, *CTV*, *PTV* and *Rectum* are selected to be saved. When *Done* was clicked we would get back to the screen in *Figure 4 (a)*, where the creation of the plans could be started (or another set of plans could be queued).



(a) (b) (c)

Figure 5: Images from the program that generates the treatment plans. Three section of the program is shown, where the objective functions are modified (a), where an objective function is modified (b) and selection DVH to be saved (c).

The saved files could then be loaded into the evaluation program. In *Figure 6*, a file with 1440 treatment plans had been loaded into the program. To the left in the figure the trade-off between two clinical goals can be seen. There are checkboxes under the figure to exclude plans that are not Pareto optimal or not clinically acceptable. In the figure, a treatment plan is selected (black star), and the DVHs for that plan are shown to the right.

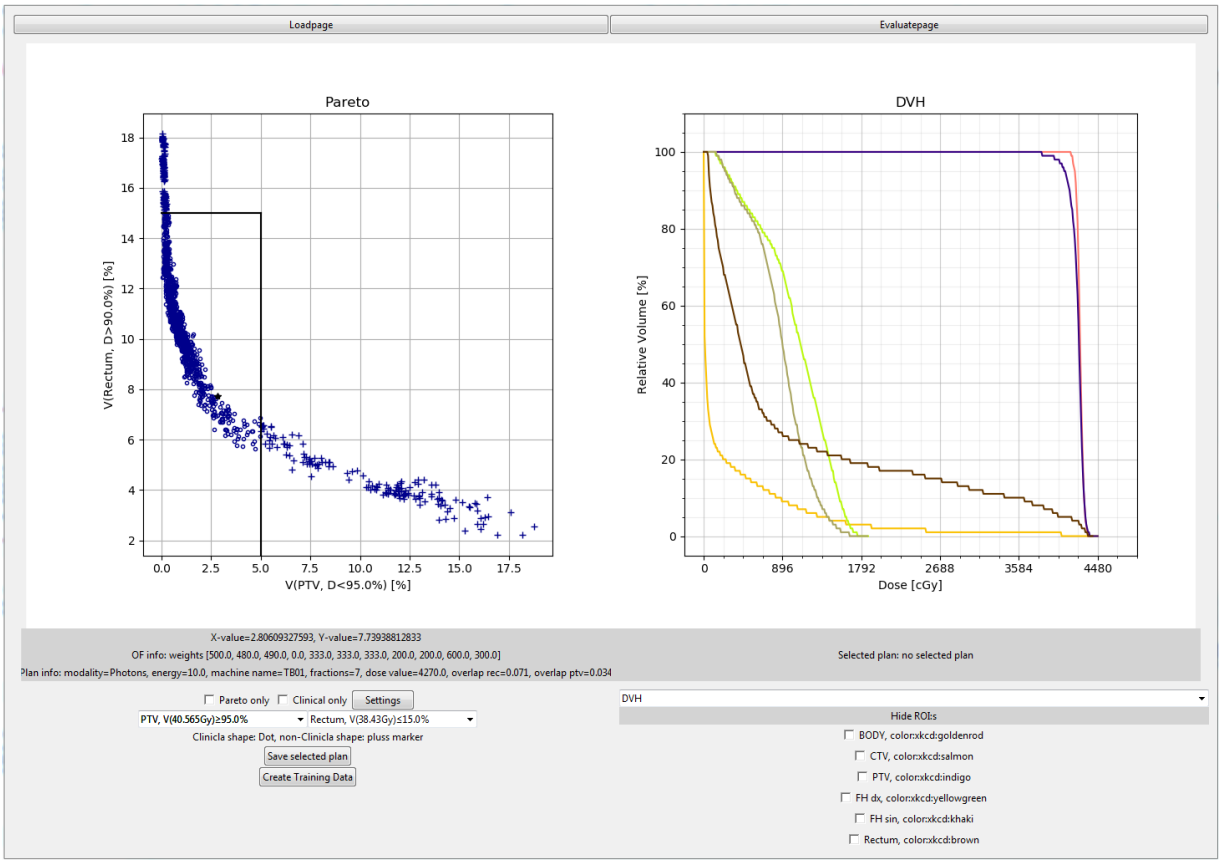


Figure 6: Image from the evaluation program. The graph to the left shows the trade-off between two clinical goals from Table 2. In this figure evaluation parameters $V_{PTV}(D < 95\%)$ and $V_{Rectum}(D > 90\%)$ is shown. Each point represents a treatment plan. The graph to the right titled DVH contains DVHs for a selected plan in the Pareto graph (marked with a star in the left graph).

9.2 Comparison with dose planner

Comparisons between a treatment plan created by a dose planner and the 1440 treatment plans generated with the treatment plan generation program are shown in *Figure 7-10*. The images are taken from the evaluation program but to better display which point represents the treatment plan produced by the dose planner, a ring (around the point) and an arrow (pointing at it) have been added to the images. The blue points in the figures are treatment plans created by the plan generation program and the green dot is the treatment plan created by the dose planner.

In the trade-off between PTV and rectum for the clinical goals $V_{PTV}(D < 95\%) = 1 - V_{PTV}(D \geq 95\%)$ and $V_{Rectum}(D > 90\%)$ is shown in *Figure 7*. Some evaluation parameters were flipped to make origin the ideal value. The clinical goals represented on the axes are the

second and third most prioritized goals from *Table 2 (*the primary goal regarding CTV coverage should always be fulfilled). Only the Pareto optimal and clinical plans are shown in this figure.

In *Figure 8-10* several other trade-offs are shown. All clinically acceptable plans are displayed in the figures.

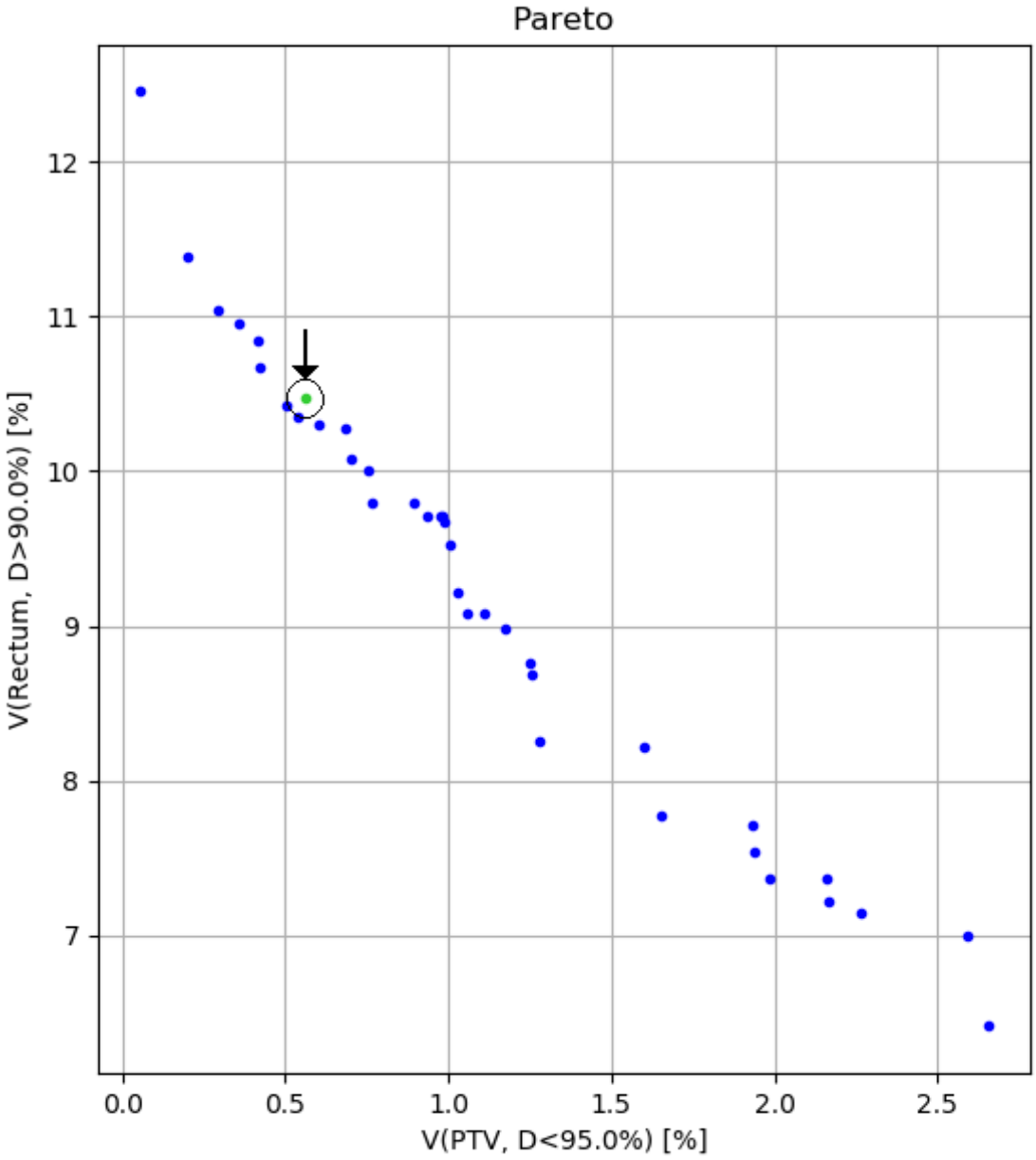
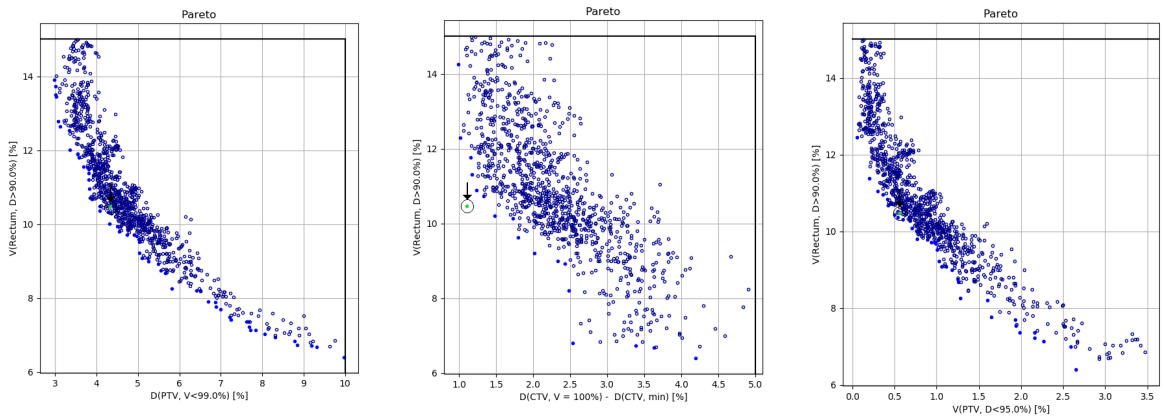
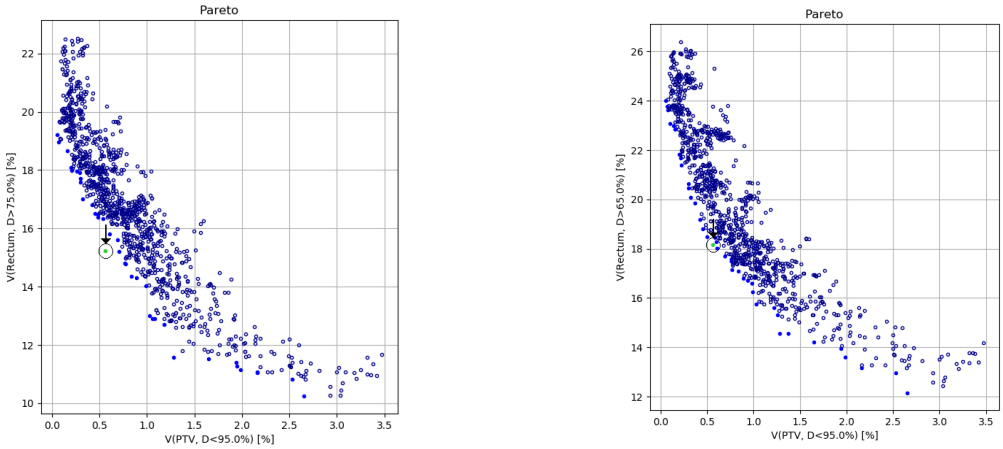


Figure 7: A comparison between automatically generated treatment plans (blue) and a treatment plan created by a dose planner (green). Only the Pareto optimal and clinical treatment plans that were automatically created are shown. The trade-off between PTV and rectum for the evaluation parameters $V_{PTV}(D < 95\%)$ *and* $V_{Rectum}(D > 90\%)$ *is shown.*



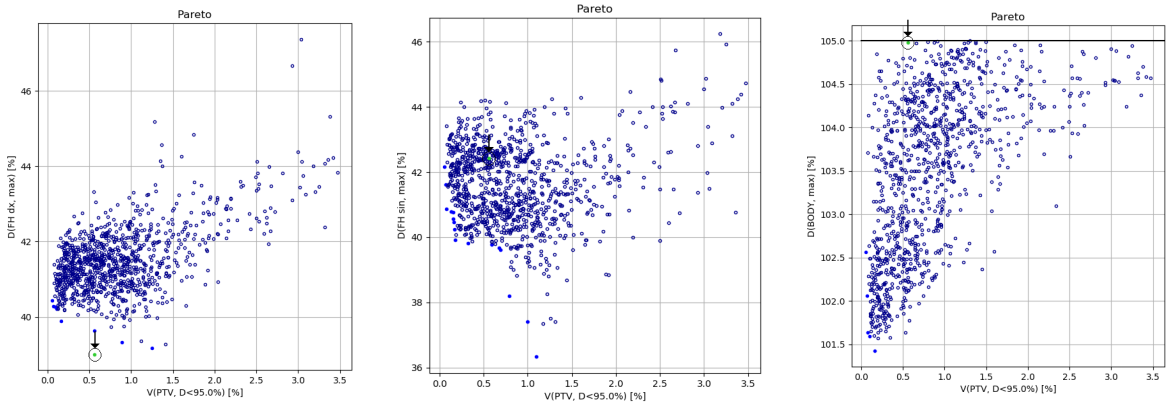
(a) (b) (c)

Figure 8: A comparison between automatically generated treatment plans (blue) and a treatment plan created by a dose planner (green). Only the Pareto optimal and clinical treatment plans that were automatically created are shown. The trade-offs between rectum and PTV or CTV for the evaluation parameters $V_{Rectum}(D > 90\%)$ *and* $D_{PTV}(D > 90\%)$ *(a),* $V_{PTV}(d < 95\%)$ *(b) or* $D_{CTV}(V_{100\%} - V_{min})$ *(c) are shown.*



(a) (b)

Figure 9: A comparison between automatically generated treatment plans (blue) and a treatment plan created by a dose planner (green). Only the Pareto optimal and clinical treatment plans that were automatically created are shown. The trade-offs between PTV and rectum for the evaluation parameters $V_{PTV}(D < 95\%)$ *and* $V_{Rectum}(D > 75\%)$ *(a) or* $V_{Rectum}(D > 65\%)$ *(b) are shown.*



(a) (b) (c)

Figure 10: A comparison between automatically generated treatment plans (blue) and a treatment plan created by a dose planner (green). Only the Pareto optimal and clinical treatment plans that were automatically created are shown. The trade-offs between PTV and femoral heads or body for the evaluation parameters $V_{PTV}(D < 95\%)$ *and* $D_{FH,dx}(Max)$ *(a),* $V_{FH,sin}(max)$ *(b) or* $V_{Rectum}(Max)$ *(c) are shown.*

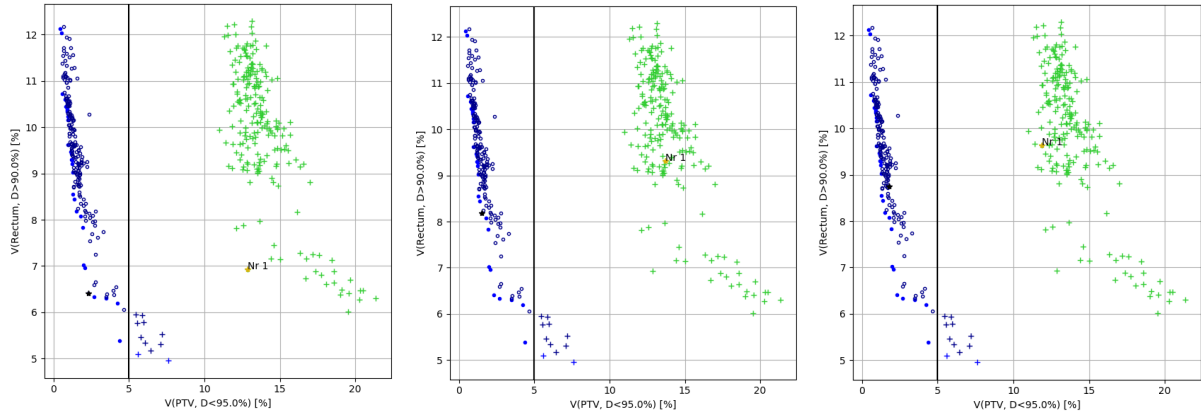
9.3 Robustness test

The option to perform a robustness test on the automatically created treatment plans was added to the program generating the treatment plans. The test was added after data from the treatment plan generation phase was saved to a file. The test was designed to move the isocenter based on prostate movement data, creating a new robustness test plan. This was followed by a collapsed cone dose calculation and saving of DVH data to a text file. The same type of data was saved for the robustness test plan as for the original plan. Multiple robustness test could be performed on the same treatment plan, meaning that many different isocenter movements could be tested for each treatment plan. The isocenter was moved back to its original position after each test. The movement options available were based on real prostate movement data collected in a parallel study, and consisted of average values, median values, or any of the actually observed values.

The robustness of a treatment plan could be evaluated in the evaluation program. This was done by loading two files, one for the original treatment plans and one for the plans with moved isocenter. When performing the evaluation both the static treatment plans and the treatment plans with the moved isocenter could be seen in the Pareto front graph (the graph showing the trade-off between a target ROI and an OAR ROI). When selecting a treatment plan in the Pareto front graph, both the original treatment plan and the corresponding treatment plans with the moved isocenter were marked (with a star). This allows the user to compare how the selected

plan was affected by the movement, compared to the other treatment plans. The isocenter movement (or movements) used in the evaluation was shown below the graph in the evaluation program.

In *Figure 11* two series from a robustness test are shown. The original treatment plans are shown in blue and the treatment plans with a moved isocenter in green. The treatment plans with a moved isocenter were moved (X, Y, Z) = (0.45cm, 0.02cm, -0.58cm). The black line represents the clinical boundary for PTV coverage. Most of the original treatment plans are clinically acceptable, while the plans with a moved isocenter are not clinically acceptable. In each of the three sub-figures one of the original treatment plans has been selected (black star), and shown together with the corresponding plan with a moved isocenter (yellow star with number).



(a) (b) (c)

Figure 11: Three screen shots of the Pareto front graph for evaluation of robustness test plans. The trade-off between PTV and rectum for the evaluation parameters $V_{PTV}(D < 95\%)$ *and* $V_{Rectum}(D > 90\%)$ *is shown. There are two series, the original plans (blue) and the plans with a moved isocenter of (X, Y, Z) = (0.45cm, 0.02cm, -0.58cm) (green). Three treatment plans are selected in the figures (black stars), two are Pareto optimal (a and b) and one is non-Pareto optimal (c). The corresponding plans with a moved isocenter is shown as yellow stars with numbers.*

10 Discussion

10.1 The software

Most of the work required was automated with the software. However, not all the work could be automated, for instance, setting up the acquisition parameters in the program that generates

the treatment plans still involves manual effort. This part can be semi-automated by the use of less specific templates, for example for hypo fractionated prostate treatments. Only a few changes are needed by the user, such as what case, treatment machine or examination to use. The thing that requires most effort is setting up the objective functions deciding how many treatment plans that should be generated. Nevertheless, the work required to create treatment plans enough to set up a Pareto front for a patient can be performed in minutes instead of days.

Possible further automation could also involve the generation of optimization ROIs based on existing ROIs. Optimization ROIs are used by the dose planner to create better treatment plans by giving the TPS a more detailed description of what is wanted. They can be used, for instance, for making more conformal treatment plans, or to better control the dose to the overlap between rectum and PTV for prostate patients. In RayStation 5, optimization ROIs can be generated by using scripting functions based on existing ROIs, and this could be included in the automation procedure.

The generation program is fast at creating and saving information from treatment plans. This is mainly due to the fact that RayStation 5 is fast at optimizing the treatment plans and calculation dose for them. The calculation time depends of course also on the computer that is used. A computer with better processors can do more calculations per second and thereby the programs will run faster. The computer used had an Intel® Xeon® processor E5-1603 (4 cores with 2.80 GHz processor base frequency) and 32-GB RAM memory. Each treatment plan took approximately 3 minutes to create and save by the generation program. To create enough treatment plans to get a good representation of the Pareto front takes 1-3 days. One needs to take into account that most of the created treatment plans will not end up on the front. Furthermore, many plans might not be clinically acceptable. If the Pareto fronts are used as comparison between treatment techniques, 3 days per front may be an acceptable optimization time. To add to this one can queue many sets of treatment plans, so one could prepare three sets that will be done after 3-9 days. While waiting for the generation program to finish the user can do other daily tasks.

There are endless amount of combination of objective functions and there is not time to test them all. The more combination that is tried, a better representation of the actual Pareto front will be obtained. If one sample to tight we will end up with similar plans and it will take longer time. If too much is demanded for an objective function the plans might end up being not clinically acceptable.

Several general improvements can be done to both the programs. In its current version, for example, if the name standard for ROIs is not followed it will fail to find the ROI it is looking for. Some cases have been solved, for instance it is no longer caps sensitive. Further improvements in this respect would be valuable from a practical point of view.

10.2 Comparison with dose planner

The dose planner's treatment plan was close to the Pareto front, especially with respect to the evaluation parameter for the PTV, $V_{PTV}(D < 95\%)$, while the rectum dose was slightly higher than the other treatment plans in the Pareto front. For these two clinical goals, the autogenerated treatment plans seemed to be as good as the dose planner's treatment plan. However, for the automatically generated treatment plans the entire trade-off between a target and an OAR for two clinical goals is given, so one could choose the most desirable plan anywhere along the Pareto front. For example, one could decrease $V_{Rectum}(D > 90\%)$ from 11% to 7% for the cost of increasing $V_{PTV}(D < 95\%)$ by 2%.

In total, there are more than two clinical goals, and the rest of them are represented in *Figure 8-10*. In these figures, all the clinically accepted treatment plans were included. This was to show all the treatment plans used to build up the Pareto front. Many treatment plans have been created, even though far from all available solutions are on the Pareto front. This was to show that treatment plans on the Pareto front in *Figure 7* do not necessarily end up on the front for other clinical goals. One could select a plan in the program and follow it through the other clinical goals to see where it ended up.

When comparing the rectum and target trade-off in *Figure 8* and *Figure 9* one can see that the dose planner's treatment plan ended up close to the Pareto front for each trade-off. This means that the plan generation program can find as good treatment plans as a dose planner when it comes to meeting the clinical goals (at least in this case). This is not very surprising since the program makes almost the same steps as a dose planner, except that it saves each change it makes, and that the changes it makes are not based on the result of the previous change. It also lets the dose planning system do all the optimizations, so that all treatment plans it finds are optimized. The user supervises which objective functions that are given to the dose planning system, similar to what a dose planner might ask for. However, this can potentially be limiting in some cases, since we are looking for something that represents all possible solutions, which we might not get if the user does not set up criteria that are wide enough.

The clinical goals for the femoral heads and the body, shown in *Figure 10,* were not pushed in the same way as the other clinical goals, so the shape of the resulting front will not look the same. This is not the same type of trade-off as between target and rectum, since we did not modify and push these goals at all.

One OAR that lay close to the prostate that could have been evaluated is the bladder. However this was not done due keep the comparison simple and that it is not is not in the HYPO-RT-PC objectives used in this thesis. HYPO Prostate cancer treatment for only prostate was chosen due to that there is mainly only rectum and PTV trade-off that is of interested, making it a simple starting case. If more organs at risks is added then would be additional trade-offs to evaluate. This would require more treatment plans to be generated to get a good representation of the Pareto-front.

The time it took the dose planner to create a treatment plan was less than a workday, while the Pareto front generation took 3 days. If the treatment planner had decided to make a Pareto front, it would have taken much longer time than the plan generation program. The total amount of work for the dose planner would also have been much more than setting up the plan generation program.

The treatment planner used two additional ROIs that were not used by the generation script. The first ROI is the overlap between PTV and rectum. This is used to get more control over an area in which the dose planning system is asked to push down the dose due to the rectum objective functions, and to increase the dose due to the PTV objective function. The second ROI is the PTV with an extra margin. This is used to create more conformal treatment plans, which is desirable. However, the conformity of the treatment plans was not compared in this test. The automatically generated treatment plans had a dose fall-off objective function stating that the dose had to decrease to 50% of the prescribed dose at 3 cm from the target.

The one of the main benefit of this program compared to multi-objective optimization is that this program generates a Pareto front instead of one treatment plan. The Pareto front have multiple uses such as comparisons between machine parameters for example energy, furthermore it gives an overview of the trade-off between target and risk organs. Another main benefit of this program is that is fully automatic and that with a few improvements it could make clinically acceptable treatment plans without the work from a doseplanner. The amount of plans that is generated as suboptimal might be able to be minimized using machine learning, however one cannot know for sure. However the time it takes to generate these plan was not

long, and this was with one normal computer. By using a serve of connected computer this time would be dramatically reduce and it could be done in a very short time. Furthermore, the user fully control how they sample the plans, they can create a template were only create 200 treatment plans is created (around 7h for the computer used in this thesis).

When it comes to creating one treatment plan for a patient a doseplanner will do a better job. The program misses things like conformity index and if the dose distribution covers the target fully (might miss one slices where the dose distribution is not good). It can't ether change the ROIs used in the optimization based on the dose distribution.

10.3 Prostate cancer case

The case that was examined in the present work was prostate cancer. This was because it had a simple case with few OAR. The only trade-off that needed to be visualized was the one between target and rectum. Due to this, only the rectum goal needed to be changed to be able to show the trade-off. If more OAR would be added, it would have taken longer time to generate enough treatment plans to represent the trade-offs.

10.4 Robust comparison

The robustness test lets the user see how a treatment plan is affected by a plan isocenter movement. To connect it with reality one can see it as if the patient is positioned, then the prostate moves, then the treatment is delivered. This does of course not fully reflect the reality, but it is simple and gives some information about how the dose distribution gets affected by prostate motion.

When evaluating the robustness test the user can click around on the treatment plans to see how they are affected by the movement. In *Figure 11(a)* one of the original treatment plans in the Pareto front is selected, which is still on the front of the cluster of treatment plans with a moved isocenter. This is what is desired. However, in *Figure 11(b)* a static treatment plan in the Pareto front ends up somewhere in the middle of the cluster of the treatment plans with moved isocenter, meaning that it is less robust compared to some of the plans around it.

Several things are not taken into account in this test. It assumes that the same movement occurs at each fraction, while in reality different movements happen for each fraction. It also assumes that only one movement occurs before the treatment had started, while it in reality might be a continuous movement through the entire treatment session.

To make it easier to evaluate one may need to include a simpler way of evaluating the effect of movements, preferably by using a single value, and one should also have a boundary beyond which a plan should no longer be desirable. The clinically acceptable boundary might be able to be used if some changes are made to the test, such as the same movement is not used for all the fractions of the treatment. For the test made this is not possible due to all plans with moved isocenter ends up not being clinically acceptable.

A good improvement to the robustness test would be to change it from one movement for the entire treatment to one movement per fraction. This would mean that all isocenter movements were performed and the dose was calculated for them all (including no movement at all). Then a movement is assigned to each fraction at random and the dose grids are summed up to give the resulting dose grid for the treatment. One could also include some probability in the selection so if a particular movement is more probable it should be selected more often. This would of course cost more time, however it would result in a more realistic test.

To make it even more realistic one could split each fraction into segments, due to the movement of the prostate might accrue during the delivery of the treatment. This causes the prostate to be located in different locations for different angles of the gantry (the prostate moves while the treatment machine is rotating around the patient). This will require even more time to calculate. Moreover, information about the speed of the movements and more detailed information about the movement needs to be collected if this were to be implemented. Currently the movement data was collected for a 30-minute interval, while the treatment session takes less than half of that time.

10.5 Machine learning

Further automation and faster implementation could potentially be gained by using machine learning. We are only interested in the plans that are Pareto optimal and clinically acceptable. If we create around 1000 treatment plans, we might be interested in less than 100 of them. It would take around 50 hours to generate 1000 plans with the treatment generation program. One could save 45 of those hours if one had a way to directly create the 100 plans we are interested in. This is for the prostate case where we only have one trade-off. For other cases with more OARs, one would need more plans, which implies that even more time might be saved.

There are several ways machine learning could be implemented. For example, one could use logistic regression algorithm to predict in advance if plans end up being Pareto optimal and clinically acceptable. The model would be fitted to labelled training data and used to make

prediction about new data. In supervised machine learning, the training data and the new data always share the same labels, which are properties of the data that is collected. An example of a label that could be used in radiotherapy would be if a plan is Pareto optimal or not. Example of features that could be used is anatomical features such as volumes of ROIs, different distances between ROIs, or overlap between ROIs (like overlap between PTV and rectum). The outcome of the evaluation parameters should also be included in the set of labelled training data.

Another way would be to use it to match the current patient with other patients in your training data set to find the most similar patient. If the patients are similar, then it might be useful to use a similar set of objective functions. The set of objective functions from the most similar patient that yielded Pareto optimal and clinical goals could then be used for the new patient. Here a k-nearest neighbour algorithm could be used. The features needed for this would also be anatomical features.

Machine learning would be able to automate the process of selecting objective function to test for the patient. Such an algorithm would find suitable objective functions based on the features of the patient.

11 Conclusions

A software has been developed and implemented that can be used for automated creation of treatment plans used for Pareto front-based dose planning. A robustness test was built-in to allow for comparison between the created plans with respect to how much a prostate motion would affect them. Features such as machine learning would be a good tool to further automate the process and to reduce generation times.

12 Future prospects

This software constitutes the initial steps towards fully automating treatment planning for external beam radiotherapy. This work can grow and evolve in many directions. The grand vision is to automate treatment planning for any diagnosis that requires radiation treatment and any treatment technique that might be used.

More features should be added to a future version of the program. The ability to generate ROIs automatically is an important step. This is an important further step towards fully automating the process. The first step for this program is to generate optimization ROIs based on already

existing ROIs. These ROIs will be based on the standard ROIs and improve the treatment plans created. One example of this could be to add a ROI with a margin around the PTV. This can be used to create a more conform plan. Another example could be to add a ROI for the overlap between PTV for the prostate and rectum. This ROI can be used to avoid getting a hot spot in that area, which is common when pushing down the dose to the rectum. Other features that could be included might be a conformity index test. This test determines the treatment plan's conformity and is a part of the decision whether a treatment plan would be accepted or not.

A robustness test for prostate movements gives the chance to determine how a dose distribution of a treatment plan is affected by prostate movements. A test could help avoid less robust treatment plans. The optimal test should be tailored to the patient that is treated. However, a step in that direction is to tailor it to a specific case. Future version of the robustness test should be closer to reality by trying to mimic the delivery of the treatment. Instead of making one isocenter movement one should split up the movement in smaller segments and connect one part of a prostate movement with a part of the treatment being delivered. The test should also be split up into fractions, where not the same movement occurs for each fraction. A new way of evaluating the test is also very useful, like a number or a grade that says something about how robust a plan is instead of comparing it to the other treatment plans.

There are many ways in which the software can be clinically useful in the future. For example, it can be used to set up a dummy run for the dose planners in the clinic. This is when all the planners in the clinic make a treatment plan for the same patient to see how they differ. The purpose is to learn from each other. These plans can be compared with a Pareto front for the patient that is easily created with the automated generation program.

This could fit in to an adaptive workflow in the future, however this depends on how that workflow looks like. If one were to receive a CT or a synthetically generated CT one could use the objectives that were used to create the Pareto front to create new treatment plans or only recalculate the dose on the new CT (if one have saved all the Pareto optimal plans). If a server is used to simultaneously create or recalculate all the treatment plans CT it would only take a few minutes all depending on computing power. It would not be plausible to do it on a normal computer (at least not if each plan takes 2 minutes and you are not abler to calculate all of them simultaneously). Then the physician would again receive a set of plans to choose from. The old Pareto front also have the bonus of being spread between having a good target coverage and low dose to OAR, this could help if there are major differences in patient anatomy and one for

example would need to lower the importance of the OAR objectives to gain sufficient target coverage.

13 Acknowledgments

Thanks to all my supervisors Hunor Benedek, André Haraldsson, Per Engström, Tommy Knöös and Crister Ceberg, who have guided and supported me through my master thesis.

Thanks to the dose planner Joakim Nilsson for helping me with the evaluation of my Program. Without the treatment plan you created I would not have had anything to compare with the quality of the treatment plans I created.

Thanks to Emilia Persson for supplying me prostate movement data. Without it, I would not be able to create the robustness test in my programs.

14 References

[1] Cancerfonden. Cancerfondsrapporten 2017[Internet]. Stockholm: Cancerfonden; 2017. [cited 2018-04-17]. Available from: http://res.cloudinary.com/cancerfonden/image/upload/v1491294283/documents/cancerfondsrapporten-2017-statistik.pdf

[2] Cancerfonden. Prostatacancer[Internet]. Stockholm: Cancerfonden; 2018 [updated 2018-06-08;cited 2018-04-17]. Available from: https://www.cancerfonden.se/om-cancer/prostatacancer

[3] Widmark A, Gunnlaugsson A, Beckman L, Thellenberg-karlsson C, Hoyer M, Lagerlund M, el al. Extreme Hypofractionation versus Conventionally Fractionated Radiotherapy for Intermediate Risk Prostate Cancer: Early Toxicity Result from the Scandinavian Randomized Phase III Trail “HYPO-RT-PC”. Int J Radiat Oncol Biol Phys. 2016; 96(5): pages 938-939.

[4] Franzén L, Widmark A. Phase III study of HYPO-fractionated Radiotherapy of Intermediate risk Localised Prostate cancer (HYPO-RT-PC). UMEÅ and SUNDSVALL: Department of Oncology and University Hospital in UMEÅ and SUNDSVALL; Version 6 (2011-12-28)

[5] Teoh M, Clark CH, Wood K, Whitaker S, Nisbet A. Volumetric modulated arc therapy: a review of current literature and clinical use in practice. BJR. 2011; 84(1007): pages 967–996.

[6] Baumann M, Krause M, Overgaard J, Debus J, Bentzen SM, Daartz J, et al. Radiationoncology in the era of precision medicine. Nat Rev Cancer. 2016;16(4): pages 234-249.

[7] Hazell I, Bzdusek K, Kumar P, Hansen C, Bertelsen A, Bertelsen A, et al. Automatic planning of head and neck treatment plans. Journal of Applied Clinical Medical Physics. 2016;vol 17 (issue 1): pages 272-282

[8] Voet PW, Dirkx ML, Breedveld S, Fransen D, Levendag PC, Heijmen BJ. Toward fully automated multicriterial plan generation: a prospective clinical study. Int J Radiat Oncol Biol Phys. 2013; 85(3): pages 866-872

[9] Voet PW1, Dirkx ML, Breedveld S, Al-Mamgani A, Incrocci L, Heijmen BJ. Fully automated volumetric modulated arc therapy plan generation for prostate cancer patients. Int J Radiat Oncol Biol Phys. 2014; 88(5): pages 1175-1179

[10] Multi-criteria optimization will change the way you plan [Internet]. RaySearchlabs: 2018[unknown;cited 2018-04-26]. Available from: https://www.raysearchlabs.com/multi-criteria-optimization-treatment-planning/

[11] Azcona JD, Li R, Mok E, Hancock S, Xing L. Automatic Prostate Tracking and Motion Assessment in Volumetric Modulated Arc Therapy With an Electronic Portal Imaging Device. INT J RADIAT ONCOL BIOL PHYS.2013; 86(4): pages 762-768.

[12] Kupelian P, Willoughby T, Mahadevan A, Djemil T, Weinstein G and Jani S. Multiinstitutional clinical experience with the Calypso System in localization and continuous, real-time monitoring of the prostate gland during external radiotherapy. INT J RADIAT ONCOL BIOL PHYS. 2007; 67(4):pages 1088-1098

[13] Amro H, Hamstra D, Mcshan D, Sandler H, Vineberg K, Hadley S, et al. The Dosimetric Impact of Prostate Rotations During Electromagnetically Guided External Beam Radiation Therapy. Int J Radiat Oncol Biol Phys. 2013; 85(1): pages 230-236.

[14] Webb S. The physical basis of IMRT and inverse planning. BJR. 2003; 76(910): pages 678-689

[15] Boyd S. Convex Optimization. Seventh printing with corrections 2009. UnitedKingdom: Cambridge University Press; 2004

[16] Petersson K, Kyroudi A, Bouris J, Ceberg C, Knöös T, Bochud F, et al. A clinical distance measure for evaluating treatment plan quality difference with Pareto front in radiotherapy. ScienceDirect. 2017 volume 3: pages 53-56

[17] Ottosson R, Engström P, Sjöström D, Behrens C, Karlsson A, Knöös T, et al. The feasibility of using Pareto fronts for comparison of treatment planning systems and delivery techniques. Acta Oncologica, 2009; 48: pages 233-237

[18] Thor M, Benedek H, Knöös T, Engström P, Behrens C, Karlsson A, et al. Introducing multiple treatment plan-based comparison to investigate the performance of gantry angle optimisation (GAO) in IMRT for head and neck cancer. Acta Oncologica, 2012; 51: pages 743-751

[19] VMAT MADE FAST AND FLEXIBLE [Internet]. RaySearchlabs: 2018[unknown;cited 2018-04-26]. Available from: https://www.raysearchlabs.com/vmat-treatment-planning/

[20] US government agency National Cancer Institute. Prostatelead [jpg-image]. 2018 [2018-11-13]. Available from: https://en.wikipedia.org/wiki/Prostate#/media/File:Prostatelead.jpg