

Large-Scale Optimization Approaches for Radiation Therapy

by

Danielle A. Ripsman

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Management Sciences

Waterloo, Ontario, Canada, 2024

© Danielle A. Ripsman 2024

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

Supervisor: Houra Mahmoudzadeh
Associate Professor, Dept. of Management Science and Engineering
University of Waterloo

Internal 1: Samir Elhedhli
Professor, Dept. of Management Science and Engineering
University of Waterloo

Internal 2: Fatma Gzara
Professor, Dept. of Management Science and Engineering
University of Waterloo

Internal-External: Ricardo Fukasawa
Professor, Dept. of Combinatorics and Optimization
University of Waterloo

External Examiner: Arkajyoti Roy
Assistant Professor, Dept. of Management Science and Statistics
University of Texas at San Antonio

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

This thesis contains several projects with contributing authors. Danielle is the lead author on the work included in each section. Input and foundational efforts are attributed accordingly.

Chapters 1, 2 and 7 are the product of independent work under the supervision of Dr. Mahmoudzadeh.

Chapter 3 is based on the paper: Ripsman, D. A., Rahimi, F., Abouee-Mehrizi, H., & Mahmoudzadeh, H. (2023). *Light Pareto robust optimization for IMRT treatment planning*. Medical Physics.

It also contains sections on CVaR from an upcoming publication with the same coauthors. Danielle was the lead author on the CVaR material from the upcoming paper, and was responsible for primary authorship, as well as generating the data and results for the published paper.

Chapter 4 includes joint supervision from Dr. Mahmoudzadeh as well as Dr. Almur Alev, and guidance from the radiation physicists at the Grand River Regional Cancer Centre (GRRCC), Dr. Ernest Osei and Dr. Johnson Darko. Danielle is the lead author on this paper.

Chapter 5 is a project undertaken in collaboration with Isaac Rudich from Polytechnique Montréal. Isaac took the lead on the implementation and leveraging his decision-diagram background for coming up with modelling techniques. Danielle was responsible for formalizing the models and took on the lead author role. The product is a joint work, under the supervision of Dr. Mahmoudzadeh, and is included in this thesis with permission.

Chapter 6 includes collaboration and supervision from Dr. Ernest Osei and Dr. Johnson Darko from the GRRCC, as well as Danielle's supervision of several undergraduate students from the University of Waterloo, including William Kwong, Gogilan Selvarajah and Mahip Singh. Danielle is the lead author on this paper.

Abstract

Radiation therapy treatment planning (RTTP) is a medically important, multistage clinical problem that poses a number of challenges in practice. This daily clinical task is full of highly-complex, large-scale operations research problems which make it difficult to both find and evaluate the highest quality plans. Modern-day planning software is often equipped with reliable but sub-optimal internal calculators and heuristics. While these heuristics may give up on some quality, they are practically and clinically viable for achieving fast results. In an ideal world, however, optimality would not need to be sacrificed in the name of clinical viability.

The goal of this thesis is to move towards the clinical viability of integrating optimal RTTP processes. There are a number of novel frameworks proposed within this document, as well as several first implementations. The largest contribution is a comprehensive look at uncertainty in radiation therapy for breast cancer treatment. In the past, uncertainty has been approached using a number of techniques, including stochastic optimization, robust optimization (RO), and plan-averaging. This thesis advances the robust field, providing an exploration of integrating faster robust approaches into both continuous and integer portions of general RO, as well as the more specific RTTP models. Promising improvements were made in terms of integration at the continuous level, but the results also suggested that the methods were less effective for integration into more complex integer models.

The thesis details a mechanism for trading off some of the conservatism characteristic of RO, for greater expected performance, without sacrificing the robustness guarantees in a light Pareto robust optimal (LPRO) framework. The document includes one of the first studies to include LPRO, and it uses the method to generate high-quality plans for radiation therapy. It also outlines the first integration of tumour-control via conditional value at risk with Pareto and light Pareto robust optimization. The LPRO method is shown to not only work effectively, but to provide meaningful treatment benefit without adding significant complexity to the underlying RO approaches.

The thesis also takes a novel, structural perspective on two notoriously difficult RTTP problems; beam angle optimization (BAO) and direct aperture optimization (DAO). The framework for BAO comes from the field of rotational geometry, and contains the steps taken to derive several potential branch and price sub problems in an effort to improve tractability. The DAO framework is inspired by the field of decision diagrams, and is the first framework to integrate aperture limitation directly into a graphical model, which is hypothesized to make the solution speed far faster and more practical for integrating into clinical planning software.

The final major study regards down-sampling in RTTP. Down-sampling is an approach that is used implicitly in planning software, and performed explicitly in most RTTP operations research papers, however, there is no agreed-upon method to perform this process. A system for comparing down-sampling approaches is introduced along with a small-scale demonstration using patient data. This study is intended to provide the basis for a larger-scale future comparison with more patient data. Finally, the thesis concludes in a proposal for future work in the field of RTTP.

Acknowledgements

I would like to begin by thanking my advisor, Dr. Houra Mahmoudzadeh. For the past eight years Houra has been everything a supervisor should be and more. She has been on top of every deadline, excited by every new idea, endlessly helpful, supportive, a mentor and a friend. Thank you so much for everything, Houra.

I would also like to thank the University of Waterloo, the Faculty of Engineering, the Department of Management Science and Engineering, and the Natural Sciences and Engineering Research Council of Canada for their financial support throughout my degree.

To my committee, Dr. Elhedhli, Dr. Gzara, Dr. Fukasawa and Dr. Roy, thank you for taking the time to read my thesis and provide your insightful comments and suggestions.

To my mom and dad, Arlene and Colin, thank you for your patience and unconditional support. I know you went to great lengths to avoid asking “are you done yet?” every time we spoke, and you did well most of the time. Mommy, you are a copy-editor beyond compare, thank you so much for reading all of this, you are the best.

To David and Ryan, thanks for always providing a listening ear and moral support as I bumbled through graduate programs, you guys are amazing brothers, I am so lucky to have you both. To Nana, I think you listened to me complain and rant more than anyone else on the planet. Thanks for always being there, every day, to listen and chat.

To all the undergraduate students I had the pleasure of researching with, particularly Charlotte, Karina and William, you truly made my experience so much richer. And I cannot forget Dr. Ernest Osei and Dr. Johnson Darko, who took me on as a research student between degrees and despite their very busy schedule, always made the time to help me with my research, and make me feel comfortable at the GRRCC. You truly changed how I see and study radiation therapy, and I cannot thank you both enough.

Finally, to my friends and colleagues, running and climbing buddies, you made the degree and city so much more fun. Isaac, working with you this past year has made so extremely excited about the possibilities for operations research. Fuat, Stoyan and Kiefer, you are great colleagues and friends. Laura and Nicole, you made my days off feel like an adventure. To all my K-12 friends and those who I’ve met since, thanks for always being just a phone call away. Chris, Mark and Monica, you’ve kept me grounded since undergrad. I hope you keep doing so forever, because I need you for the next steps too.

Dedication

*To my grandfather, **Dr. Bernard Rubenstein**.
I know how proud you always were and would have been today.
I am so excited to have a PhD now, just like you.*

Table of Contents

List of Figures	xiii
List of Tables	xvi
List of Abbreviations	xvii
List of Symbols	xix
1 Introduction	1
1.1 Radiation Therapy Treatment Planning	2
1.2 Thesis Contributions	7
1.3 Contribution to Clinical Data Acquisition	10
1.4 Thesis Outline	11
2 Closed-Form Cutting-Plane Methods Under Polyhedral Uncertainty	12
2.1 Introduction	13
2.2 Background	16
2.3 Cutting Plane for RLO	18
2.3.1 Classical CP Algorithm	18
2.3.2 Closed-Form Cutting-Plane Sub Problem	20
2.3.3 Generalizing the Budget of Uncertainty	23
2.4 Cutting-Plane Procedures for RMIO	24

2.4.1	Current Literature on CP for RMIO	26
2.4.2	The Bertsimas et al. (2016) Repair Heuristic	28
2.4.3	Alternative Implementation for CP in RMIO	29
2.5	Application to Radiation Therapy	31
2.5.1	Robust linear radiation therapy (RLO)	32
2.5.2	Robust mixed integer radiation therapy (RMIO)	33
2.5.3	Warm starting the RMIO	35
2.6	Numerical Results	36
2.6.1	RLO model results	37
2.6.2	RMIO model results	40
2.7	Conclusions	44
3	Light Pareto Robust Optimization with CVaR Extensions	46
3.1	Introduction	47
3.2	Basic PRO and LPRO Models	49
3.3	Extended CVaR Models	53
3.4	Methods and Materials	58
3.4.1	Clinical Data and Processing	58
3.4.2	Breathing Pattern Simulation	59
3.4.3	Proposed Breathing Pattern Generation Algorithm	60
3.5	Results	61
3.5.1	Basic Model Nominal Results	62
3.5.2	Basic Model Random Breathing Simulations	65
3.5.3	Extended CVaR Model	70
3.6	Discussion	76
3.7	Conclusions	77

4	A Geometrical Approach to Beam Angle Optimization	78
4.1	Problem Definition	80
4.2	Methods	82
4.2.1	<i>2-Dimensional</i> Beam Rotation Approach	82
4.2.2	<i>2-Dimensional</i> Patient Rotation Approach	85
4.2.3	Extensions to the 3 rd Dimension	87
4.2.4	Model	91
4.3	Solution Approaches	92
4.3.1	Pricing Problem	93
4.3.2	Beam Rotation-Based Pricing	94
4.3.3	Patient-Rotation-Based Pricing	98
4.3.4	Heuristic Angle-Generation Approach	100
4.4	Results	103
4.4.1	Solving the Sample Problem	104
4.4.2	Solving Larger Sample Problems	104
4.5	Conclusions	105
5	An Exploration of Network-Based Direct Aperture Optimization	107
5.1	Introduction	108
5.2	Problem Preliminaries	111
5.3	A minimum-cost flow formulation	114
5.3.1	Aperture enforcement and splitting	116
5.3.2	Cutting plane binary relaxations	122
5.3.3	Problems with decoherence and merging	124
5.3.4	Decoherence detection heuristic	126
5.3.5	Super paths for decoherence prevention	129
5.3.6	Node-splitting for decoherence prevention	133
5.4	Heuristic Warm Start Approach	135
5.5	Conclusions	136

6	A Framework for Comparing Voxel Down-Sampling Approaches	137
6.1	Background Literature	139
6.2	Sampling Techniques	141
6.2.1	Integer Down-Sampling	142
6.2.2	k -means Sampling	142
6.2.3	k -Neighbour Sampling	144
6.2.4	Beamlet-Based Sampling	144
6.3	Proposed Layered Approach to Sampling	145
6.4	Voxel Normalization Strategies	147
6.5	Results	148
6.5.1	Patient Sampling Visualizations	148
6.5.2	Numerical Results	151
6.6	Conclusions	158
7	Future Research Directions	159
7.1	Parameter Selection in Radiation Therapy	159
7.1.1	Inverse Methods	162
7.1.2	Future Steps and Applications to RTTP	164
7.2	Extensions to the Warm Start Methodology	166
7.3	Future Light Pareto DRO Approaches to RTTP	168
7.3.1	Intra-Fractional Motion and RO	168
7.3.2	DRO Version of the Intra-Fractional Problem	170
7.3.3	Inter-Fractional Motion and DRO	175
7.4	Future Work in BAO	177
7.5	Future Work in DAO	179
7.6	Future Work in Sampling	180
8	Conclusions	181
	References	183

List of Figures

1.1	4D CT scan depiction	2
1.2	Labelled components of linear accelerator	3
1.3	Image of beamlets and voxels	4
1.4	Representation of IMRT collimator	6
1.5	Map of contributions in context of treatment process	7
1.6	Workflow to accommodate and validate new planning methods	11
2.1	RMIO branching solution methodologies	25
2.2	Literature CP implementation methods	27
2.3	Repair heuristic possible outcomes	29
2.4	Proposed CP approach	30
2.5	Boxplot of the SP runtimes for both CP models	39
3.1	Ideal DVH diagram	53
3.2	Five-phase breathing cycle	59
3.3	DVHs of the CTV and heart, nominal breathing	66
3.4	Scatter plots of D99% of CTV dose and the average heart dose	69
3.5	CVaR tumour DVH for RO, PRO and LPRO	72
3.6	CVaR heart DVH for RO, PRO and LPRO	73
3.7	DVH with nominal realized breathing for uncertainty set	74
3.8	Gain function	76

4.1	2D gantry rotation depiction	81
4.2	Three 2D beamlet rotation abstractions	82
4.3	Rotating 2D beamlets	83
4.4	Beamlet depiction	83
4.5	Plane rotation	84
4.6	Rotating the patient-voxel grid by $\theta = -45^\circ$	85
4.7	Voxel v rotation summary	86
4.8	Renderings of the 3D beam components	87
4.9	Beam inclusion in rotation	96
4.10	Initial angle offset	97
4.11	Single beamlet rotation signs	97
4.12	Examples of pre-processing a set of angle critical points	101
4.13	Random trial 1 output	105
5.1	Column generation for aperture selection	108
5.2	A simple 3×3 beam.	109
5.3	Network representation of all the nodes from a 3×3 beam	110
5.4	Network representation with 3 beam angles	112
5.5	Single-angle network without vertical discontinuities	113
5.6	Transformed legal aperture, and rejected disconnected aperture	113
5.7	Undesirable aperture properties due to leaf motion	114
5.8	Infeasible graph recommendation for two apertures	118
5.9	3-way+ split at node	118
5.10	Abstract decoherence example	124
5.11	Concrete representation of the decoherence shown on 4×3 graph	126
5.12	Single-node decoherence example	127
5.13	Super path depiction	131
5.14	Legal apertures violate super path sum	131

5.15	Superpath elimination technique	132
5.16	Node splitting approach for decoherence	133
5.17	Extra splitting for multi-decoherence	134
5.18	Network warm start depiction	136
6.1	CT vs. planning voxel granularity	137
6.2	Integer- n sampling	138
6.3	Layered approach to sampling	146
6.4	Single patient plan image	148
6.5	Integer sampling example	149
6.6	Every down-sampling approach except integer	150
6.7	DVHs of extreme sampling rates	156
6.8	All DVH diagrams for the $n = 10$ sampling rate	157
7.1	Two IO examples	162
7.2	IO infeasible example	165
7.3	Proposed branch-and-bound approach	167
7.4	Phase-based uncertainty	169
7.5	PMF around time spent in finite phases with uncertainty	169
7.6	Ambiguity set based on a polyhedral uncertainty	170
7.7	Issues with random points and summations	170
7.8	DVH plots for the DRO, PDRO and LPDRO model results	174
7.9	Uncertain $\tilde{D}_{v,b}$ for inter-fractional motion	176
7.10	Water phantom PDD	177
7.11	A PDD taken using CERR's dose engine	178
7.12	A PDD taken using matRad's dose engine	178

List of Tables

2.1	Summary metrics for the three RLO models	37
2.2	Divergence of the two CP algorithms	40
2.3	Summary metrics for the RC, CP, CPn1 and CPnAll models	42
2.4	Warm start only summary metrics	43
2.5	Summary metrics for the RC and CP models with warm start	43
3.1	Nominal aggregate dose values for each model	63
3.2	Dose values at random breathing realizations	68
3.3	Time (mins) to run CVaR	70
3.4	Dose metrics for nominal and extreme breathing patterns	75
4.1	Result changes at various A -values, random trial 1	105
6.1	Full plan dose outcomes	152
6.2	Run times (s) of the various sampling methods	153
6.3	Dose metrics in Gy for the various sampling methods	155
7.1	5 five-phase $\tilde{\mathbf{p}}$ making up $\mathcal{D}_1$	173
7.2	Dose metrics for the DRO models	174

List of Abbreviations

- BAO** beam angle optimization 4, 5, 8, 9, 11, 78–80, 107, 141, 159
- CG** column generation 5, 93, 102, 108–111, 136
- CP** cutting-plane 13–15, 18, 23–26, 28–30, 36, 44, 50, 70
- CT** computed tomography 2, 80, 106, 137, 148, 149
- CTV** clinical target volume 2, 3, 6, 49–52, 62, 64, 65, 91, 100, 140, 141, 148, 152, 180
- CVaR** conditional value-at-risk 48, 49, 54, 57, 62, 70, 71, 76, 171–173, 179, 180
- DAO** direct aperture optimization 5, 8, 9, 11, 32, 33, 35, 77, 107–111, 113, 116, 136, 139, 141, 159, 168, 179, 181
- DRO** distributionally robust optimization 47, 168–175
- DVH** dose volume histogram 53, 65, 75, 76, 154, 156
- FMO** fluence map optimization 4, 5, 8, 9, 32, 46, 48, 49, 78, 79, 81, 91, 107, 109, 115, 135, 137, 139–141, 159, 171
- IMRT** intensity-modulated radiation therapy 3, 5, 31, 36, 46–50, 77
- IO** inverse optimization 160–162, 165
- LP** linear programming 13, 15–18, 24, 30, 31, 39, 61, 160, 167
- LPDRO** light Pareto distributionally robust optimization 168, 173, 175
- LPRO** light Pareto robust optimization 48, 49, 54, 55, 57, 77, 168, 173

LRO light robust optimization 48

LS leaf sequencing 4, 6, 78, 79, 107, 109, 139

MILP mixed integer linear programming 5, 9, 44, 108, 139, 166

OAR organ at risk 2, 3, 6, 81, 91, 100, 141, 148, 151

OR operations research 1, 47, 161, 162, 165, 181, 182

PDD percentage depth dose 177–179

PDRO Pareto distributionally robust optimization 173, 175, 176

PMF probability mass function 50, 169, 170

PRO Pareto robust optimization 48, 49, 51, 54, 55, 173

RC robust counterpart 13–15, 24, 25, 36, 44, 50, 55, 61, 70

RLO robust linear optimization 13–16, 18, 24, 30, 34, 37, 40, 41, 43, 44

RMIO robust (linear) mixed integer optimization 13–16, 18, 24, 29, 30, 37, 40–42, 44

RO robust optimization 8, 32, 46–49, 54, 57, 59, 60, 76, 77, 168–171, 173, 175, 176, 179, 180

RTTP radiation therapy treatment planning 1–3, 7–9, 12, 16, 20, 31, 44, 48, 49, 53, 78, 121, 135, 137, 139, 158, 159, 161, 165, 166, 179–182

SO stochastic optimization 47

VaR value-at-risk 53

VMAT volumetric modulated radiation therapy 5, 110, 179, 182

List of Symbols

Global symbols for all chapters

Sets

$\mathcal{T}$	set of target organs
$\mathcal{H}$	set of healthy organs
$\mathcal{B}$	set of beamlets
$\mathcal{V}$	set of voxels
Θ	set of all angles
$\mathcal{U}$	general uncertainty set
$\mathcal{K}$	set of breathing phases

Constants

$D_{v,b}/D_{v,b,k}$	dose from beamlet b to voxel v at phase k
L_v	prescribed dose for voxel v

Variables

w_b	intensity of each beamlets b
-------	--------------------------------

Chapter 1

Introduction

The [radiation therapy treatment planning \(RTTP\)](#) problem is an interesting [operations research \(OR\)](#) application, as it is both clinically relevant, and composed of a number of large-scale, mathematically diverse sub-problems. Seeing as radiation therapy, in the extreme, may have life-and-death ramifications, or at the very least, profound quality-of-life impacts, a lot of work has and continues to go into improving the quality and scope of automation used in the planning process ([Romeijn and Dempsey, 2008](#); [Ehrgott et al., 2010](#); [Cho, 2018](#)). In clinic, [RTTP](#) is regarded as a difficult, multistage process that blends technology with human decision-making in an effort to optimize the resulting treatment for cancer patients ([Bortfeld, 2006](#)). Modern clinical software is advanced, but typically heuristic-based and therefore, non-optimal, requiring expert knowledge and iterative adjustment to create plans with desirable metrics. In an ideal world, optimal models would be used throughout the planning process, with human intervention being reserved for intuitive trade-off evaluation. However, in order for these tools to be integrated into a clinical environment, they must be rapid, reliable and account for many diverse features.

This thesis explores a number of modelling approaches, at various intervention stages of the [RTTP](#) process, in an attempt to get closer to future clinical integration. It explores

issues relating to down-sampling, motion uncertainty, trade-offs in terms of conservativeness, beam angle selection, exploiting graph structure for aperture selection, and finally, several recommendations for future work. In addition to its clinical importance, [RTTP](#) provides a good basis for exploring optimization methodologies, as nearly every stage of the process may be modeled as a unique, large-scale optimization problem.

This chapter is broken down as follows, Section [1.1](#) introduces the preliminaries and describes the [RTTP](#) process. Section [1.2](#) outlines the thesis contributions both in general, and within the [RTTP](#) space, specifically. Section [1.3](#) briefly touches on the challenges of acquiring data for [RTTP](#). Section [1.4](#) provides the structure for the remainder of the thesis.

1.1 Radiation Therapy Treatment Planning

Radiation therapy treatment delivery typically begins with [computed tomography \(CT\)](#) imaging. The [CT](#) captures a 3D image of the patient’s tumour or lesion, often referred to as the [clinical target volume \(CTV\)](#), as well as any surrounding [organ at risk \(OAR\)](#)s. If breathing motion impacts the treatment site, several of these 3D images may be captured at various phases in the breathing cycle, resulting in a 4D-CT scan, as depicted in [Figure 1.1](#). Patient organs are then delineated and labelled, often through a manual process ([van Dijk et al., 2020](#)), mathematically resulting in a 3D grid of structure-assigned units called voxels.

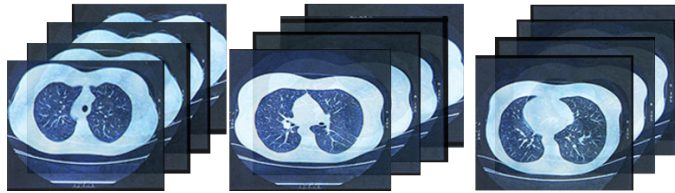


Figure 1.1: Depiction of a 4D CT scan with 3 breathing phases. Original image courtesy of [The Johns Hopkins University, The Johns Hopkins Hospital and Johns Hopkins Health System \(2024\)](#).

Once a set of labelled voxels has been determined, the remainder of the [RTTP](#) process centres on finding a strategy to destroy all cells in the [CTV](#), while keeping the critical [OARs](#) intact. The methods employed at this stage depend on the radiation delivery modality at hand. The focus of this thesis will be on the commonly-used modality of external-beam radiation therapy called [intensity-modulated radiation therapy \(IMRT\)](#) (see book by [Webb \(2015\)](#) for a comprehensive review on the topic). As an overview, an [IMRT](#)-enabled device is built on a machine called a linear accelerator, as shown in [Figure 1.2](#). The linear accelerator generates beams of radiation through an opening called a collimator, which contains dynamic metal parts which then modulate the shape of the beam. The collimator sits on a rotating arm called a gantry, that typically allows for 360° of clearance to rotate around the patient. The treatment delivery part of the process generally occurs while the patient is lying immobilized on a bench. The treatment is typically delivered over several days or weeks, in sessions called fractions.

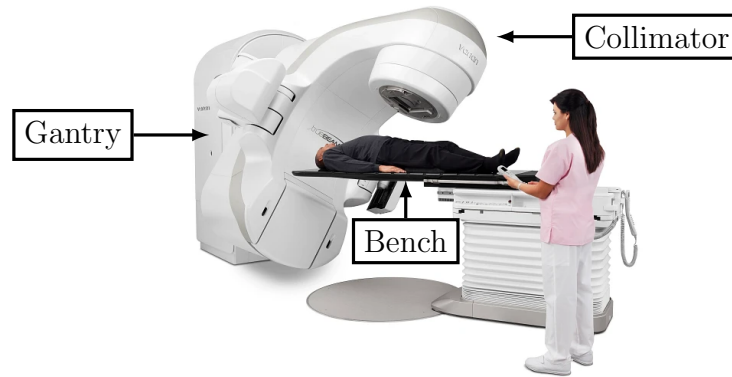


Figure 1.2: Labelled components of a linear accelerator. Original image courtesy of [Varian Medical Systems, Inc. \(2024\)](#).

Perhaps the most important piece of the input for the [RTTP](#) decision problems discussed in this thesis is the information regarding how dose travels throughout the patient body. This information may be captured in a matrix referred to as the *dose-influence matrix*. The dose-influence matrix describes how dose from each portion of the beam,

called a beamlet, impacts each patient voxel (see Figure 1.3 for a visual depiction of the relationship between the two components in 3-dimensions).

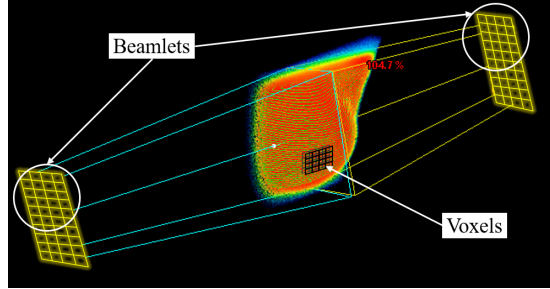


Figure 1.3: The voxels and beamlets noted on a radiation therapy treatment plan.

The dose-influence matrix is typically calculated using a convolution or Monte Carlo simulation engine (Hurkmans et al., 1995; Dorje, 2014; Pandu et al., 2022) and in general, it is assumed to scale linearly with beamlet intensity. This has shown to be a good assumption, even though it does not fully account for interplay between neighbouring-beamlet intensities, which may impact the dose in practice.

The rotating gantry and programmable contouring of the beam shape allow radiation physicists to design customized treatment plans for every individual patient. These capabilities have revolutionized radiation therapy over the past few decades, but at the cost of dramatically increasing planning complexity (Webb, 2015). Treatment planning has become so complex that in practice, it is often broken down into 3 distinct steps (Romeijn and Dempsey, 2008):

Step 1: **beam angle optimization (BAO)**. A finite set Θ of gantry angles are chosen.

Step 2: **fluence map optimization (FMO)**. An idealized plan is generated at the intersection of the $|\Theta|$ angles.

Step 3: **leaf sequencing (LS)**. An (often approximate) algorithm translates the **FMO** results into a mechanically deliverable plan for the collimator.

Note, the above steps refer to a specific and common mode of **IMRT** delivery called step-and-shoot. Continuous methodologies such as **volumetric modulated radiation therapy (VMAT)** use “path selection” rather than angle selection, as they sweep through multiple angles in a continuous motion ([Mahn timer et al., 2017](#)). There is a lot of active research that deals with combining stages 2 and 3 using either heuristic, **mixed integer linear programming (MILP)** or **column generation (CG)** approaches ([Shepard et al., 2002](#); [Li et al., 2003](#); [Milette, 2008](#); [Broderick et al., 2009](#); [Romeijn et al., 2005](#); [Men et al., 2007](#); [Carlsson, 2008](#); [Salari and Unkelbach, 2013](#); [Mahn timer et al., 2017](#); [Ripsman, 2018](#); [Ripsman et al., 2022](#)). These combined approaches are typically referred to as **direct aperture optimization (DAO)**. There has similarly been work on combining steps 1 and 2 ([Lee et al., 2003](#); [Aleman et al., 2009](#); [Bertsimas et al., 2013](#)).

Step 1 is considered a hard optimization problem, and while a number of **BAO** methods have been proposed in the literature, in clinic, general practice entails simply selecting equidistant beams when possible (although some clinics have region-specific heuristics of their own) ([Romeijn and Dempsey, 2008](#); [Bertsimas et al., 2013](#); [Amit et al., 2015](#)). This practice occurs despite modern studies showing opportunities for plan improvement when any form of optimization is applied to the **BAO** process ([Nazareth et al., 2009](#); [Dias et al., 2014](#); [Sadeghnejad-Barkousaraie et al., 2021](#)).

Step 2 is arguably the most studied stage of the process, with a wealth of literature ranging from linearizing nearly every type of radiation dose requirement, to accounting for biological aspects like tumour control and complication probability, to addressing clinical concerns like motion and setup uncertainty ([Romeijn et al., 2003, 2006](#); [Chu et al., 2005](#); [Chan et al., 2014b](#); [Romeijn and Dempsey, 2008](#); [Mahmoudzadeh et al., 2015](#)). In this thesis, the the following simple, linear **FMO** model is considered for most applications, and

more application-specific dose features are integrated on an as-needed basis,

$$\begin{aligned}
& \min_{\mathbf{w} \in \mathbb{R}_+^{|\mathcal{B}|}} \sum_{s \in \mathcal{H}, \mathcal{T}} \sum_{v \in \mathcal{V}_s} \sum_{b \in \mathcal{B}} \frac{D_{v,b}}{|\mathcal{V}_s|} w_b \\
& \text{s.t.} \sum_{b \in \mathcal{B}} D_{v,b} w_b \geq L_v \quad \forall v \in \mathcal{V}_{\mathcal{T}}.
\end{aligned} \tag{FMO}$$

The decision variable w_b indicates the intensity of each beamlets $b \in \mathcal{B}$ (which encompasses all angles in Θ). The input dose-influence matrix for each beamlet b in relation to each voxel v is denoted $D_{v,b}$. Sets $\mathcal{H}$ and $\mathcal{T}$ represent healthy organs (**OARs**) and target organs (**CTVs**), respectively. The model minimizes the expected dose to all organs, $s \in \mathcal{H} \cup \mathcal{T}$, subject to delivering the clinically prescribed dose L_v to all **CTV** voxels, $v \in \mathcal{V}_{\mathcal{T}}$.

The output of model (**FMO**), is unlikely to be feasible for the delivery portion of the collimator, as depicted in Figure 1.4. As such, algorithms associated with step 3, or the **LS** portion of the process, are used to process the optimized solution into a deliverable treatment plan. The deliverability is determined by whether the set of variable intensities w_b , may be recreated by a finite set of apertures within the space between the two opposing sets of tungsten leaves. This problem grows increasingly difficult as more beam angles are incorporated into the problem and/or multiple apertures are allowed to form at each beam-angle.

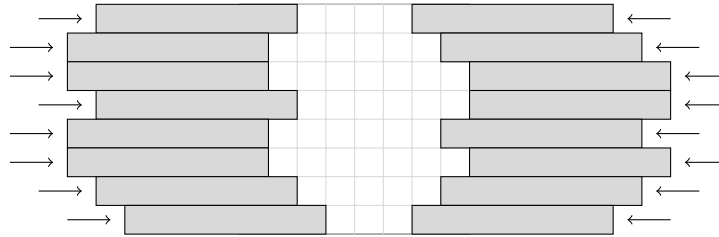


Figure 1.4: Representation of the delivery component of an IMRT collimator. The grid represents the beamlets, and the beam is modulated via covering leaves, which may not be broken or misaligned.

1.2 Thesis Contributions

The studies conducted in this thesis examine structural features of elements of the [RTTP](#) process, with the goal of improving upon them individually. Ideally, these improvements will facilitate more efficient combined approaches down the road. There are five areas with proposed interventions, which are outlined in the order of [RTTP](#) relevance in Figure 1.5.

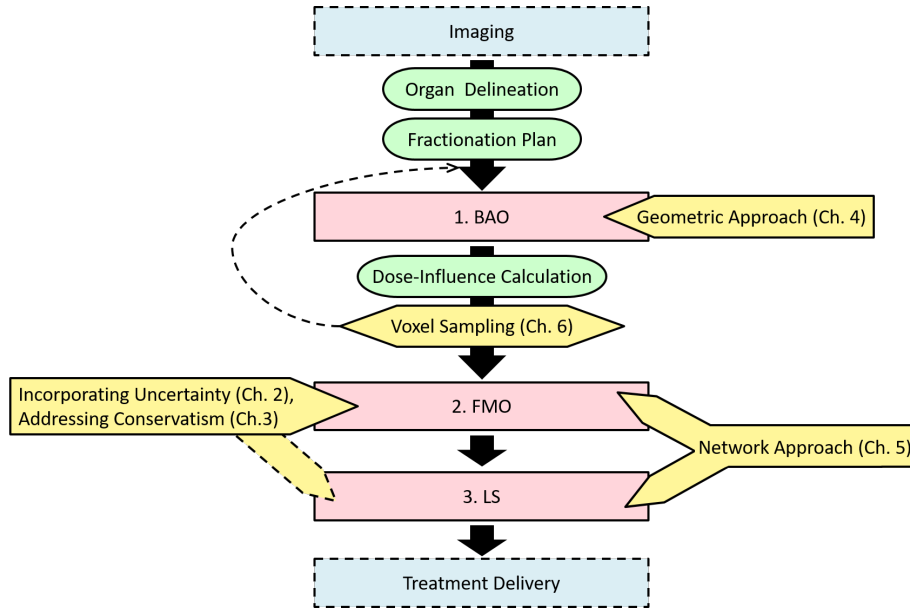


Figure 1.5: Map of contributions in the context of the [RTTP](#) process. Angular yellow items are contributed work, pink boxes are treatment planning steps, green rounded items are intermediate procedures and blue non-solid boxes are clinical patient-facing procedures.

Within the thesis, the chapters have been sorted in the sequence of immediate impact. Chapters 2 and 3 present work from three different complete projects, with proof-of-concept studies; each of which may be immediately integrated into [RTTP](#) planning software. Chapters 4 and 5 represent frameworks that have high potential for impact, but are large undertakings that will require continued development. Chapter 6 provides a road map for a large-scale study, currently demonstrated with a single set of patient data. Finally, Chapter 7 contains proposals for future work.

Chapter 2 provides a comprehensive look at the uncertainty in radiation therapy for breast cancer treatment. It does so by first generalizing the uncertainty into a subclass of polyhedral uncertainty, then by providing guidance on how to efficiently integrate this subclass into optimization models. This work contributes to the field of [robust optimization \(RO\)](#) through an exploration of cutting-plane approaches in both continuous and integer problems. Promising improvements are made in terms of integrating these methods into continuous models, specifically the [FMO](#) model for [RTTP](#). While the results from directly integrating the cuts into the mixed integer models were not as promising, they led to the development of high-quality warm-starts for the [DAO](#) problem that integrate the continuous robust techniques instead.

Chapter 3 details a mechanism for trading off some of the conservatism characteristic of [RO](#), for greater expected performance, without sacrificing the robustness guarantees in a light Pareto robust optimal framework. The chapter includes one of the first studies to include light Pareto robust optimization, and it uses the method to generate high-quality [RTTP](#) solutions. It also outlines the first integration of tumour-control via conditional value at risk with Pareto and light Pareto robust optimization. The proposed methods are shown to not only work effectively, but to provide meaningful treatment benefit without adding significant complexity to the underlying models.

The thesis also takes a novel, structural perspective on two notoriously difficult [RTTP](#) problems; beam angle optimization ([BAO](#)) and direct aperture optimization ([DAO](#)). Chapter 4 presents the [BAO](#) framework, which uses a rotational geometric approach to derive sub-problems for a branch-and-price column-generation methodology, in order to choose a finite set of appropriate beam angles for a given patient plan. It is shown that by switching the perspective of the rotation, the complexity of modelling can be significantly reduced, although the problem remains large and non-linear. Further complexity reductions come in the form of a final proposed heuristic-based approach, that acts as a more memory-intensive alternative to solving either of the non-linear pricing sub-problems.

Chapter 5’s DAO framework, on the other hand, aims to solve two problems with existing methods for formulating DAO models. The first problem is that MILP models are too slow to effectively solve the problem, as re-demonstrated in Chapter 2. The second problem is that column generation methods do not allow for direct aperture control; they may generate an intractable number of apertures and will be biased towards generating increasingly small apertures. The proposed method is the first framework to integrate aperture limitation directly into a network model, which is hypothesized to make the solver faster and more practical for integration into clinical planning software.

The last major study in Chapter 6 focuses on down-sampling in RTTP. Down-sampling is an approach used to reduce the complexity in RTTP. It is used implicitly in most planning software, and explicitly in much of the planning literature. There is, however, no agreed-upon method for performing this process. A system for comparing common down-sampling approaches is introduced along with new proposed sampling methods and updates to existing methods. This study is intended to provide the framework for a future larger, comparative, multi-patient study. As a standalone project, however, it gives some preliminary indications of the relative strength of a number of different common approaches, and a resulting set of planning recommendations.

In summary, this document contributes:

1. A rapid, closed-form cutting-plane solution approach for classes of robust uncertainty.
2. A light Pareto robust approach to reducing FMO conservatism under uncertainty.
3. The first conditional value-at-risk implementation of light Pareto robust optimization.
4. A novel geometric approach that moves towards BAO tractability.
5. A novel DAO framework for integrating aperture decisions into a network structure.
6. A comprehensive, unified comparison of new and existing voxel sampling approaches.
7. A number of exciting future research directions in the field of optimization and RTTP.

1.3 Contribution to Clinical Data Acquisition

Several parts of this study use breathing motion data from the Princess Margaret Cancer Centre in Toronto. These breast cancer datasets were created using 4D-CT scanners, and RayStation (version 3.99.0.8; RaySearch Laboratories AB, Stockholm, Sweden). Voxels were tracked using a hybrid deformable image registration technique discussed in [Weistrand and Svensson \(2015\)](#), [Kadoya et al. \(2016\)](#), and [RaySearch Laboratories \(2017\)](#). The sampled datasets used in the experiments in this study have been previously made available as supplementary material for the manuscript [Ripsman et al. \(2022\)](#). They are used to conduct all experiments involving motion uncertainty.

For the remaining projects however, it was desirable to generate more data and a more flexible workflow, for cancer centres that do not have access to academic versions of the software. At many hospitals, including our partner hospital in the Kitchener-Waterloo region, the patient location and structural information is stored in a flexible DICOM format, which stands for digital imaging and communications in medicine. The DICOMs of concern are all single-phase, as the local hospital does not perform 4D CT scans, so with the help of the radiation physicists and a number of undergrads over many semesters, we experimented with software capable of making this slightly simpler form of data accessible. Figure 1.6 shows the workflow that we selected. We used open-source software and custom [MATLAB \(2022\)](#) scripting to communicate with Eclipse, the hospital’s clinical software.

To choose a data-generation software, we investigated open-source radiation therapy options. While up-and-coming software implementations like Portpy ([Jhanwar et al., 2023](#)) are being rolled out, the current implementations do not allow for integration with new hospital data, nor do they facilitate dose-influence matrix-calculation at this time.

Older softwares like CERR (a computational environment for radiotherapy research) ([Deasy et al., 2003](#)) were experimented with at early phases of the research, due to their exemplary data-handling in MATLAB, however their dose-calculation engines seemed to

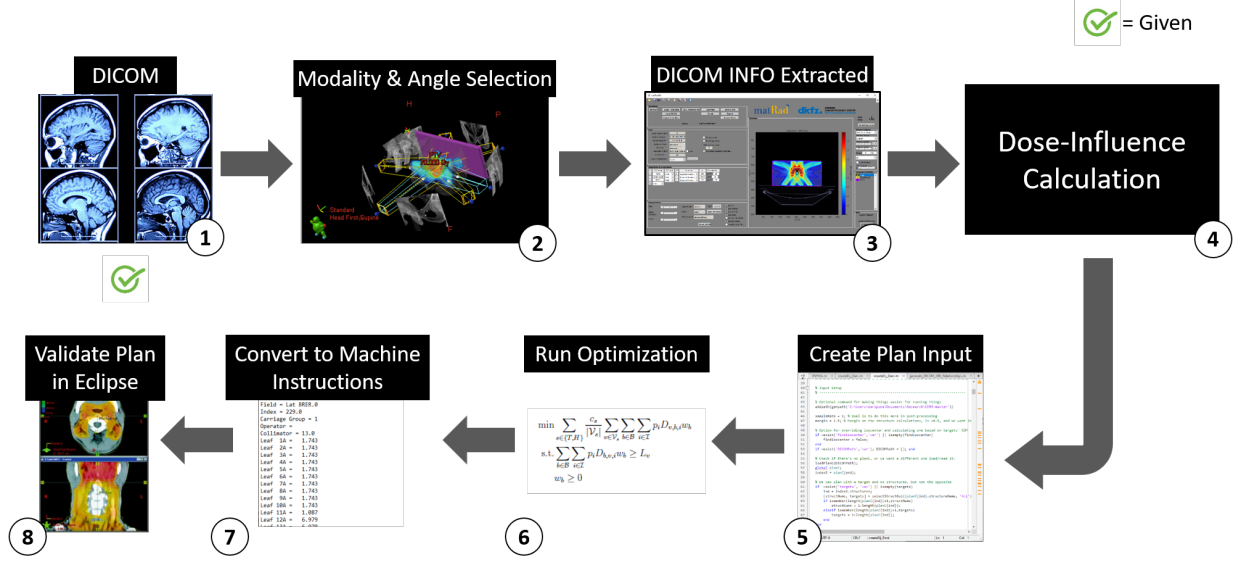


Figure 1.6: Workflow to accommodate and validate new planning methods

lag by modern standards. As a result, the data extraction and dose-influence calculation steps above were done in matRad (Wieser et al., 2017), which thanks to the help of a team of several undergraduate students over a number of semesters, we were able to verify had a comparable dose engine to that of clinical software.

Finally, we worked on a validation pipeline, translating optimized output into machine input, where the local system was Eclipse.

1.4 Thesis Outline

Chapter 2 describes the robust cutting-plane approach for handling uncertainty, Chapter 3 presents the work on light Pareto robust optimization. Chapters 4 and 5 present the novel BAO and DAO frameworks, respectively. Chapter 6 provides the down-sampling framework and small-scale study and Chapter 7 contains the proposal for future work. Finally, the thesis is concluded in Chapter 8.

Chapter 2

Closed-Form Cutting-Plane Methods Under Polyhedral Uncertainty

A major assumption within the basic [FMO](#) model, is that the dose-influence matrix is known and constant throughout the treatment. For cases with breathing motion, e.g., if the target region borders on the patient's thoracic cavity, which is the case in lung, liver and breast cancer, the dose intensity matrix becomes a continuous function of the time t within a breathing cycle, $D_{v,b}(t)$. This potentially hysterical patient breathing motion is difficult, both to predict and to model tractably. [Chan et al. \(2006\)](#) and [Bortfeld et al. \(2008\)](#) show that this form of motion can be effectively captured using 4D planning, which may in turn be protected against using robust linear optimization. In this chapter, the specific form of robust uncertainty found in this type of [RTTP](#) is characterized and a cutting-plane solution methodology is proposed. The methodology is then demonstrated using [RTTP](#) breast cancer patient data.

The chapter is organized as follows, in [Section 2.1](#) the problem is introduced. In [Section 2.2](#), the problem is stated in its most general form. The tools and derivations for the proposed cutting-plane methodology, and its relationship to past work are then explained in

Section 2.3 (continuous) and Section 2.4 (integer). The application of intensity-modulated radiation therapy is re-introduced and the model-compatibility is shown in Section 2.5. An analysis of the methodologies using real-world radiation therapy treatment planning data is then given in Section 2.6. Finally, the chapter is concluded in Section 2.7.

2.1 Introduction

A [robust linear optimization \(RLO\)](#) with polyhedral uncertainty is commonly applied to [linear programming \(LP\)](#) models that have uncertain parameters with linear interdependencies. [RLO](#) has previously been shown to have a linear [robust counterpart \(RC\)](#), meaning it can be solved tractably ([Ben-Tal and Nemirovski, 1999](#)). Despite its tractability, however, the size of the [RC](#) reformulation can make it cumbersome to solve for larger problem instances. This issue becomes even more pronounced when the problem has integer variables, i.e., in [robust \(linear\) mixed integer optimization \(RMIO\)](#). In the case of [RMIO](#), the polynomial number of additional variables and constraints added by the [RC](#) framework has an even more detrimental impact on runtimes ([Fischetti and Monaci, 2012](#); [Bertsimas et al., 2016](#)).

An alternative to the [RC](#) reformulation approach is a two-stage [cutting-plane \(CP\)](#) method. In the [CP](#) method for [RLO](#), a subset of the uncertain constraints are generated, and then iteratively, the most pessimistic set of violated constraints is added and the problem is re-solved, until no further violations are found ([Mutapcic and Boyd, 2009](#)). For [RMIO](#), the same conceptual approach applies, however, a cutting-plane schema must now be integrated into the original branch-and-bound solution method (now referred to as branch-and-cut), which leaves some flexibility in terms of implementation. Methods applied to addressing this integration tend to come from the roots of [Kelley \(1960\)](#) and [Gomory \(1963\)](#), whereas general modern methods for speeding up mixed integer programming (MIP) solvers, such as disjunctive cuts and in-and-out methods (e.g., [Ben-Ameur](#)

and Neto (2007); Pessoa et al. (2013); Fischetti and Salvagnin (2010); Poirrier and Yu (2019)) have fewer benefits under the polyhedral robust framework. This is due to the fact that all cuts generated are already guaranteed to be effective polyhedral corner points, and performance quality relies on dispatching them efficiently.

Both Fischetti and Monaci (2012) and Bertsimas et al. (2016) approach the question of efficiency in the RLO and RMIO frameworks, using RC vs. CP solution methodologies. Both authors focus on a specific type of polyhedral uncertainty originally introduced in Bertsimas and Sim (2004), called budget of uncertainty (or cardinality-constrained uncertainty). They also similarly observe that this particular class of uncertainty has a closed-form representation at the sub problem level, meaning CP can be implemented with even higher efficiency than the general case. The two studies go on to do a thorough comparison of the methods, with Bertsimas et al. (2016) doing more experimentation regarding when to add cuts in the solution procedure.

Noteably, Fischetti and Monaci (2012) and Bertsimas et al. (2016) come to different conclusions in their work, with Fischetti and Monaci (2012) determining:

- For RLO: CP outperforms the RC method.
- For RMIO: RC is preferable to CP.

On the other hand, after running even more experiments and controlling for different factors, Bertsimas et al. (2016) find that:

- For RLO: CP is faster for the majority of cases, however, there is no clear overall winner, as RC sometimes far outperforms CP in certain cases.
- For RMIO: no clear winner is found.

Ultimately, Bertsimas et al. (2016) propose a hybrid solver, that runs both methods in parallel, returning the fastest. Note that the work of Bertsimas et al. (2016) also investigates ellipsoidal methods, which may be reformulated as a second-order cone programming

problem. Interested readers are encouraged to look at the results of their study, as the comparisons and implementations are more complex than that of the linear models.

The present study revisits the comparison between the two solution methods for [RLO](#) and [RMIO](#) for several reasons. The first is that minor implementation details, when overlooked can have a major impact on the problem. For example, the class of problems that do not require a secondary set of optimizations is broader than the budget of uncertainty, and yet many practical applications of these formulations continue to solve a secondary [LP](#), at every iteration (eg., [Gavranović and Buljubašić \(2011\)](#); [Mahmoudzadeh et al. \(2016\)](#)). This introduces unnecessary numerical instability into the problem. In order to combat this issue, the form of polyhedral uncertainty sets that fall under this category, are explicitly characterized and then applied to a real-world example of radiation therapy.

The second purpose for revisiting this topic is to propose more minimalist branch-and-cut schemes, in contrast to the more heavy-handed heuristic approaches proposed by [Bertsimas et al. \(2016\)](#) and [Fischetti and Monaci \(2012\)](#), in order to determine whether or not the methods were simply becoming over-constrained.

Finally, the experimentation is brought back to a real-world radiation therapy implementation, similar to that proposed in [Ripsman et al. \(2022\)](#). Since both of the original comparative studies' authors concluded that the [RMIO RC](#) primarily outperformed [CP](#) in terms of finding infeasible cases, treatable patients with satisfiable lower-bound constraints were chosen, so that these past experiments may be repeated with no chance of infeasibility. The [RC](#) and [CP](#) methods are also integrated into a fast warm-start solution approach, that is again, simpler than that introduced in past literature. More formally, this chapter makes the following contributions:

1. A derivation and proof of correctness for a closed-form [CP](#) sub problem for a class of polyhedral uncertainty.
2. An equivalence is shown between budget-of-uncertainty and a subset of the high-

lighted polyhedral uncertainty class.

3. An alternate branching scheme is proposed for the [RMIO](#) problem.
4. An application of the novel approaches to the [RTTP](#) process is shown, and a simple, fast, robust warm start is proposed.

2.2 Background

In this section a foundation is built for the modelling methods used in the rest of the chapter. Note that calligraphic notation is used for sets, bold capital letters for matrices, bold lower-case letters for vectors (which are oriented column-wise), and tilde for uncertain variables.

[RLO](#) is characterized similarly to a general [LP](#) model, i.e., $\min_{\mathbf{x}} \{\mathbf{c}^T \mathbf{x} \mid \mathbf{A} \mathbf{x} \leq \mathbf{b}, \mathbf{x} \in \mathbb{R}^n\}$. The only difference comes from uncertainty in elements of the coefficient matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, often denoted by $\tilde{\mathbf{A}} \in \mathbb{R}^{m \times n}$ ([Bertsimas et al., 2011](#)). Note, for simplicity, $\mathbf{A}$ will be used to denote the nominal coefficients, around which the values of $\tilde{\mathbf{A}}$ will vary. Let $\mathcal{I}$ represent the complete set of rows $\{1, \dots, m\}$ of $\tilde{\mathbf{A}}$, such that $\tilde{\mathbf{a}}_i$ is the i^{th} row of $\tilde{\mathbf{A}}$, and let $\mathcal{J}$ represent the complete set of columns, $\{1, \dots, n\}$. Assuming $\tilde{\mathbf{A}}$ is in an uncertainty set $\mathcal{U}$, a general [RLO](#) model may be expressed as follows,

$$\begin{aligned} & \underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} && \mathbf{c}^T \mathbf{x} \\ & \text{subject to} && \tilde{\mathbf{a}}_i^T \mathbf{x} \leq b_i \quad \forall \tilde{\mathbf{a}}_i \in \mathcal{U}_i, i \in \mathcal{I}. \end{aligned} \tag{RLO}$$

When [RLO](#) problems have polyhedral uncertainty sets, $\mathcal{U}_i$ takes the form,

$$\mathcal{U}_i = \{\tilde{\mathbf{a}}_i \in \mathbb{R}^n \mid \mathbf{U}_i \tilde{\mathbf{a}}_i \leq \mathbf{t}_i\}, \tag{2.1}$$

where $\mathbf{U}_i$ and $\mathbf{t}_i$ are the constraint matrix and right-hand values, respectively, that govern

the uncertainty around each row-parameter $\tilde{\mathbf{a}}_i$. Note that in constructing the [RLO](#) model, the parametric uncertainty in each row i is assumed to be row-wise separable. As such, the robust solution, optimized for the worst-case realization of $\tilde{\mathbf{a}}_i$ for a given $\mathbf{x}$, is exactly the solution to the following [LP](#), $\forall i \in \mathcal{I}$,

$$\begin{aligned} & \underset{\tilde{\mathbf{a}}_i \in \mathbb{R}^n}{\text{maximize}} && \mathbf{x}^T \tilde{\mathbf{a}}_i \\ & \text{subject to} && \mathbf{U}_i \tilde{\mathbf{a}}_i \leq \mathbf{t}_i. \end{aligned} \tag{2.2}$$

Conceptually, the goal of model (2.2) is to find the worst-case realization of $\mathbf{a}_i^T \mathbf{x}$ for each row i of the original model ([RLO](#)), while ensuring that they still satisfy the less than b_i requirement. Note, this formulation still holds for deterministic [LPs](#), in which rows without uncertainty will have singular uncertainty sets, while columns with no uncertainty will have strict parametric equalities, rather than parameter ranges.

Taking the dual of (2.2), with variables $\mathbf{y}_i$ for each $i \in \mathcal{I}$ and working them into the original model has been shown, e.g., [Bertsimas et al. \(2011\)](#), to lead to the following complete linear robust counterpart model,

$$\begin{aligned} & \underset{\{\mathbf{x}, \mathbf{y}_i\} \in \mathbb{R}}{\text{minimize}} && \mathbf{c}^T \mathbf{x} \\ & \text{subject to} && \mathbf{t}_i^T \mathbf{y}_i \leq b_i && \forall i \in \mathcal{I}, \\ & && \mathbf{U}_i^T \mathbf{y}_i - \mathbf{x} = 0 && \forall i \in \mathcal{I}, \\ & && \mathbf{y}_i \geq 0 && \forall i \in \mathcal{I}. \end{aligned} \tag{2.3}$$

This formulation guarantees that $\mathbf{x}$ is chosen such that the worst possible realization of $\mathbf{U}_i$ is still feasible, since $\mathbf{t}_i^T \mathbf{y}_i \geq \max_{\tilde{\mathbf{a}}_i} \{\tilde{\mathbf{a}}_i^T \mathbf{x} \mid \mathbf{U}_i \tilde{\mathbf{a}}_i \leq \mathbf{t}_i\}$, but is smaller than b_i , $\forall i \in \mathcal{I}$.

The cost of implementing this robust counterpart reformulation is an increase in the number of variables and constraints. Whereas a fixed or nominal [LP](#) has an $m \times n$ coefficient matrix $\mathbf{A}$, the robust counterpart model (2.3) adds $|\mathbf{t}_i|$ new variables for each of the $i \in \mathcal{I}$

new constraints, as well as $m \times n$ new equality constraints, resulting in a much larger $m(n+1) \times [\sum_{i \in \mathcal{I}} |\mathbf{t}_i| + n]$ parameter matrix. Note, notation $|\cdot|$ is used to capture the length of a vector.

The **RMIO** model is almost identical to (**RLO**), as only variable domains are affected. Thus, equations (2.2) or (2.3) remain true, and the domain may simply be partitioned as follows, i.e., $\mathbf{x} = \mathbf{x}^1 \cup \mathbf{x}^2$, with $\mathcal{J}^1 := \{1, \dots, n_1\}$ and $\mathcal{J}^2 := \{1, \dots, n_2\}$, such that $\mathcal{J} = \mathcal{J}^1 \cup \mathcal{J}^2$, for the following **RMIO** model,

$$\begin{aligned} & \text{minimize} \quad \mathbf{c}^T \mathbf{x} \\ & \text{subject to} \quad \tilde{\mathbf{a}}_i^T \mathbf{x} \leq b_i \quad \forall \tilde{\mathbf{a}}_i \in \mathcal{U}_i, i \in \mathcal{I}, \\ & \quad \mathbf{x} \in \mathbb{Z}^{n_1} \times \mathbb{R}^{n_2}. \end{aligned} \tag{RMIO}$$

2.3 Cutting Plane for RLO

In this section, the **CP** methods for solving the **RLO** and **RMIO** problems are outlined. First, a classical approach to **RLO** using **CP** for a general polyhedral uncertainty set is described in Section 2.3.1. Then, in Section 2.3.2, the class of polyhedral uncertainty under study is explicitly characterized, and the closed form **CP** algorithm is proposed. Finally, in Section 2.3.3, the ubiquitous budget-of-uncertainty approach of **Bertsimas and Sim (2004)**, is shown to belong to the examined class of polyhedral uncertainty.

2.3.1 Classical CP Algorithm

The following is a standard **LP**-based solution algorithm used in **CP** approaches for solving the generic polyhedral model (**RLO**), with uncertainty set (2.1). The logic behind the algorithm is divide-and-conquer: when (**RLO**) has the uncertainty set $\mathcal{U}_i$ substituted by a finite subset of its values, i.e., $\mathcal{U}_i^q \subset \mathcal{U}_i$ at each row i , the resulting $\mathbf{x}^q$ will provide a

lower bound for the actual optimal objective function value. If $\mathbf{x}^q$ does not violate any constraints in the full problem, it is, by definition, optimal.

The master problem is defined as follows:

$$\begin{aligned} & \underset{\mathbf{x}^q \in \mathcal{R}}{\text{minimize}} \quad \mathbf{c}^T \mathbf{x}^q \\ & \text{subject to} \quad \tilde{\mathbf{a}}_i^T \mathbf{x}^q \leq b_i \quad \forall \tilde{\mathbf{a}}_i \in \mathcal{U}_i^q, \quad i \in \mathcal{I}. \end{aligned} \tag{MP}$$

The sub problem for each row $i \in \mathcal{I}$ is similarly defined as:

$$z_i^{sub} = \underset{\tilde{\mathbf{a}}_i \in \mathcal{U}_i^q}{\text{maximize}} \quad \mathbf{x}^{qT} \tilde{\mathbf{a}}_i. \tag{2.4}$$

Since the problem is row-wise separable, (2.4) can be aggregated in a single $m \times [n + m]$ optimization model, where the violations $\mathbf{v}$ of constraint $\tilde{\mathbf{a}}_i^T \mathbf{x}^q \leq b_i$ are maximized for all $i \in \mathcal{I}$ rows simultaneously, i.e.,

$$\begin{aligned} & \underset{\mathbf{v} \in \mathcal{R}, \tilde{\mathbf{a}}_i \in \mathcal{U}_i^q}{\text{maximize}} \quad \sum_{i=1}^m v_i \\ & \text{subject to} \quad \mathbf{x}^{qT} \tilde{\mathbf{a}}_i - v_i \geq b_i \quad \forall i \in \mathcal{I}. \end{aligned} \tag{SP}$$

Based on these formulations, Algorithm 1 is used to find $\mathbf{x}^*$. Note, $\mathcal{U}_i^0$ is typically chosen

Algorithm 1 General RLO CP Solution Algorithm

```

0: choose valid subset  $\mathcal{U}_i^0 \subset \mathcal{U}_i \quad \forall i \in \mathcal{I}$ 
1: for all  $q \in \{0, \dots, \infty\}$  do
2:     Solve MP  $\rightarrow \mathbf{x}^q$ 
3:     Solve SP  $\rightarrow v_i, \tilde{\mathbf{a}}_i \quad \forall i \in \mathcal{I}$ 
4:     if  $|v_i > 0 \quad \forall i \in \mathcal{I}| = 0$  then
5:         return  $\mathbf{x}^q$ 
6:     else
7:          $\mathcal{U}^{q+1} \leftarrow \mathcal{U}^q \cup \{\tilde{\mathbf{a}}_i \mid v_i > 0 \quad \forall i \in \mathcal{I}\}.$ 

```

to be the nominal parameter vector $\mathbf{a}_i$ for each row i . Empirically, Algorithm 1 tends to

converge in a relatively limited number of iterations, while in the worst-case, it generates $n! \times |\mathcal{I}|$ constraints, over an equally large number of iterations (Mahmoudzadeh et al., 2016).

2.3.2 Closed-Form Cutting-Plane Sub Problem

Within the space of polyhedral uncertainty, a common class of problems has the row-wise uncertainty, $\mathcal{U}_i$, bound by a single linking constraint. These single-connecting constraint polyhedral uncertainty problems have a wide range of applications, including in the RTTP domain as discussed in Section 2.5. With this class structure in mind, this section proposes a closed-form solution methodology, based on a continuous knapsack reformulation.

Consider an uncertainty set of the form,

$$\mathcal{U}_i = \{\tilde{\mathbf{a}}_i \in \mathbb{R}^n \mid \underline{\mathbf{u}}_i \leq \tilde{\mathbf{a}}_i \leq \bar{\mathbf{u}}_i; \mathbf{u}_i^T \tilde{\mathbf{a}}_i \leq t_i\}. \quad (2.5)$$

Based on this uncertainty definition, formulation (2.2) can be written as, $\forall i \in \mathcal{I}$,

$$\begin{aligned} & \underset{\tilde{\mathbf{a}}_i \in \mathbb{R}^n}{\text{maximize}} && \mathbf{x}^T \tilde{\mathbf{a}}_i \\ & \text{subject to} && \mathbf{u}_i^T \tilde{\mathbf{a}}_i \leq t_i, \\ & && \underline{\mathbf{u}}_i \leq \tilde{\mathbf{a}}_i \leq \bar{\mathbf{u}}_i. \end{aligned} \quad (2.6)$$

Recall that the general uncertainty matrix, $\mathbf{U}_i$, has arbitrary dimension $|\mathbf{t}_i| \times n$ for each row $\tilde{\mathbf{a}}_i$. The single connecting-constraint class limits that uncertainty matrix to a single connecting vector $\mathbf{u}_i$, or a dimension of $1 \times n$ for each row $\tilde{\mathbf{a}}_i$. Furthermore, the parameters may each be separately bounded above and below by vectors $\bar{\mathbf{u}}_i$ and $\underline{\mathbf{u}}_i$, respectively. The robust counterpart model (2.3) under this class of polyhedral uncertainty has a dimensionality of $m(n+1) \times [m+n+2mn]$ (unless the uncertain parameters have no bounds, i.e., $\bar{\mathbf{u}}_i = \infty$ and $\underline{\mathbf{u}}_i = -\infty$, then the size is $m(n+1) \times [m+n]$).

Moreover, if $\sum_j \underline{\mathbf{u}}_i$ is subtracted from t_i , and the lower bound is replaced with $\mathbf{0}$, this forms a continuous knapsack problem. As such, to provide a closed-form solution to the [SP](#), step 2 of the iterative algorithm can be replaced with a sequencing problem. The form and correctness of this algorithm can be shown as follows.

Proposition 1. *There exists a greedy, ordered allocation of uncertainty in parameters $\tilde{\mathbf{a}}_i$, for each row $i \in \mathcal{I}$, that always maximizes the sub problem (SP).*

Proof. Let the ordering of parameters be $\mathbf{o}_i$ in each row i , such that the value over proportional weight of each element of $\frac{\mathbf{x}}{\mathbf{u}}$ is monotonically decreasing. This can be done by mapping each variable, $j \in \mathcal{J} \mapsto o_i(j) \in \mathcal{J}$ such that

$$\frac{x_{i,o_i(j)}}{u_{i,o_i(j)}} \geq \frac{x_{i,o_i(j)+1}}{u_{i,o_i(j)+1}} \quad \forall o_i \in \{1, \dots, |\mathcal{J}| - 1\}, \forall i \in \mathcal{I}.$$

Without loss of generality, assume $\mathbf{u} > 0$, as negatives may be flipped and any indices where $u_{i,j} = 0$ may be separately taken to be $\max\{x_j \underline{u}_{i,j}, x_j \bar{u}_{i,j}\}$, and then neglected from the sorting procedures mentioned below. Next, it may be shown by contradiction that allocating the largest allowable values of $\tilde{\mathbf{a}}_i$ by ordering $\mathbf{o}_i$ satisfies (SP). Suppose proportion $\tilde{\mathbf{a}}_i^*$, results in the maximum objective function value, z_i^{sub*} , but does not have the maximum values of $\tilde{\mathbf{a}}_i$ allocated in order $\mathbf{o}_i$. Let $\tilde{\mathbf{a}}_i'$ be a valid, identical vector to $\tilde{\mathbf{a}}_i^*$, except in indices $o_i(j)$ and $o_i(j) + 1$, where,

$$\tilde{\mathbf{a}}_{i,o_i(j)}' = \tilde{\mathbf{a}}_{i,o_i(j)}^* + \delta, \text{ and}$$

$$\tilde{\mathbf{a}}_{i,o_i(j)+1}' = \tilde{\mathbf{a}}_{i,o_i(j)+1}^* - \delta,$$

i.e., a larger value is allocated to an element that is earlier in ordering $\mathbf{o}_i$ for $\tilde{\mathbf{a}}_i'$. The associated objective function value is:

$$z' = z_i^{sub*} + \delta \left(\frac{x_{i,o_i(j)}}{u_{i,o_i(j)}} - \frac{x_{i,o_i(j)+1}}{u_{i,o_i(j)+1}} \right),$$

which, in cases where the indices correspond to non-equal elements of $\frac{\mathbf{x}}{\mathbf{u}}$, i.e., $\frac{x_{i,o_i(j)}}{u_{i,o_i(j)}} > \frac{x_{i,o_i(j)+1}}{u_{i,o_i(j)+1}}$, we have $z' > z_i^{sub*}$, violating the initial assumption. Note: for equal elements of $\frac{\mathbf{x}}{\mathbf{u}}$, $z' = z_i^{sub*}$, i.e., the order does not matter. $\square$

Based on this proposition, Algorithm 2 is proposed for generating $\tilde{\mathbf{a}}_i$.

Algorithm 2 Closed-form (SP) Solver

- | | |
|--|--|
| 1: set $f \leftarrow \mathbf{u}_i^T \underline{\mathbf{u}}_i$ | ▷ Calculate the floor of the solution space in row i |
| 2: set $T \leftarrow \min\{\mathbf{u}_i^T \bar{\mathbf{u}}_i, t_i\}$ | ▷ Calculate the upper bound on row i (feasibility) |
| 3: set $d \leftarrow T - f$ | ▷ Calculate the difference |
| 4: for all columns $\iota \in \mathcal{J}$ do | |
| 5: $j \leftarrow o_i^{-1}(\iota)$ | ▷ Find the ι^{th} item in the ordering |
| 6: $\delta \leftarrow \min\{d, u_{i,j}(\bar{u}_{i,j} - \underline{u}_{i,j})\}$ | ▷ Assign maximum allowable δ |
| 7: $\tilde{a}_{i,j} \leftarrow \underline{u}_{i,j} + \delta/u_{i,j}$ | ▷ Update the optimal proportion vector |
| 8: $d \leftarrow d - \delta$ | ▷ Update the differential |
-

Proposition 2. *Algorithm 2 generates a feasible uncertainty vector, i.e., $\tilde{\mathbf{a}}_i \in \mathcal{U}_i$.*

Proof. The proportion vector $\tilde{\mathbf{a}}_i$ meets: (1) upper bound, (2) lower bound and (3) constraint criteria of set $\mathcal{U}_i$, by construction:

(1) Upper bound $\tilde{\mathbf{a}}_{i,j} \leq \bar{u}_{i,j} \forall j \in \mathcal{J}$ is enforced by Step 7 as follows:

$$\begin{aligned}
 \underline{u}_{i,j} + \delta/u_{i,j} &\leq \underline{u}_{i,j} + u_{i,j}(\bar{u}_{i,j} - \underline{u}_{i,j})/u_{i,j} \\
 &= \underline{u}_{i,j} + (\bar{u}_{i,j} - \underline{u}_{i,j}) \\
 &= \bar{u}_{i,j}.
 \end{aligned}$$

(2) Similarly, Step 7 forces the constructed $\tilde{\mathbf{a}}_i$ to exceed set $\mathcal{U}_i$'s lower bound in any feasible problem, since by definition, at every iteration, d can only be reduced by d or a positive value smaller than d (since $u_{i,j} \geq 0$, and $\underline{u}_{i,j} \leq \bar{u}_{i,j}$) $\implies \delta/u_{i,j} \geq 0$ at all times, meaning $\tilde{a}_{i,j} \geq \underline{u}_{i,j}$.

(3) Finally, the constructed $\tilde{\mathbf{a}}_i$ respects set $\mathcal{U}_i$'s constraint: $\mathbf{u}_i^T \tilde{\mathbf{a}}_i \leq t_i$, as Step 1 enforces

that $d \leq t_i - \mathbf{u}_i^T \underline{\mathbf{u}}_i$, while Steps 6-8 ensure that $\mathbf{u}_i^T \tilde{\mathbf{a}}_i \leq \mathbf{u}_i^T \underline{\mathbf{u}}_i + d$, therefore, t_i can never be exceeded. $\square$

Proposition 3. *The constructed $\tilde{\mathbf{a}}_i$ from Algorithm 2 is equivalent to that produced in the CP optimization step 2.*

Proof. Algorithm 2 takes the maximum slack in the system d and iteratively allocates it to the elements in $\tilde{\mathbf{a}}_i$, by ordering o_i . It is therefore guaranteed to find the worst-case $\tilde{\mathbf{a}}_i$ at each iteration. $\square$

These three propositions taken together show that the use of Algorithm 2 replaces the need for optimization in the sub problem in the typical CP Algorithm for this select class of polyhedral uncertainty. Note that the bottle-neck in this algorithm is simply a sorting procedure that may be efficiently implemented in $O(n \log n)$ time.

2.3.3 Generalizing the Budget of Uncertainty

The following section shows that the budget of uncertainty is contained within the class of polyhedral uncertainty considered in this paper. As an overview, the budget of uncertainty requires that the uncertainty around each of row i 's parameters, $\underline{\mathbf{u}}_i \leq \tilde{\mathbf{a}}_i \leq \bar{\mathbf{u}}_i$, has only Γ_i parameters differing from their nominal values (where Γ_i may or may not be an integer). As such, the worst-case realization of $\tilde{\mathbf{a}}_i$ can be found using the following model,

$$\begin{aligned}
& \underset{\tilde{\mathbf{a}}_i, \mathbf{h}_i^+, \mathbf{h}_i^- \in \mathbb{R}^n}{\text{maximize}} && \mathbf{x}^T \tilde{\mathbf{a}}_i \\
& \text{subject to} && \tilde{\mathbf{a}}_i \leq \mathbf{a}_i + \mathbf{h}_i^+ (\bar{\mathbf{u}}_i - \mathbf{a}_i), \\
& && \tilde{\mathbf{a}}_i \geq \mathbf{a}_i - \mathbf{h}_i^- (\mathbf{a}_i - \bar{\mathbf{u}}_i), \\
& && \mathbf{1}^T \mathbf{h}_i^+ + \mathbf{1}^T \mathbf{h}_i^- \leq \Gamma_i, \\
& && \mathbf{0} \leq \mathbf{h}_i^-, \mathbf{h}_i^+ \leq \mathbf{1},
\end{aligned} \tag{2.7}$$

where variables $\mathbf{h}_i^+$ and $\mathbf{h}_i^-$ represent the positive and negative deviations from nominal $\mathbf{a}_i$, within row budget, Γ_i . The value of each $\tilde{\mathbf{a}}_i$, however, can be shown to be determined by $\mathbf{h}_i^+$ and $\mathbf{h}_i^-$ at optimality, as calculated by the following formula $\tilde{\mathbf{a}}_i = \mathbf{a}_i + (\bar{\mathbf{u}}_i - \mathbf{a}_i) \circ \mathbf{h}_i^+ - (\mathbf{a}_i - \underline{\mathbf{u}}_i) \circ \mathbf{h}_i^-$. Note, notation $\circ$ is used to indicate element-wise multiplication.

Due to this equivalence, the model may also be expressed as the following single-constraint LP model,

$$\begin{aligned} & \underset{\mathbf{h}_i^+, \mathbf{h}_i^- \in \mathbb{R}^n}{\text{maximize}} && [\mathbf{x} \circ (\bar{\mathbf{u}}_i - \mathbf{a}_i)]^T \mathbf{h}_i^+ - [\mathbf{x} \circ (\mathbf{a}_i - \underline{\mathbf{u}}_i)]^T \mathbf{h}_i^- \\ & \text{subject to} && \mathbf{1}^T \mathbf{h}_i^+ + \mathbf{1}^T \mathbf{h}_i^- \leq \Gamma_i, \\ & && \mathbf{0} \leq \mathbf{h}_i^+, \mathbf{h}_i^- \leq \mathbf{1}. \end{aligned} \tag{2.8}$$

Formulation (2.8) corresponds to the single-constraint model (2.6), with $\mathbf{x}$ replaced by $[\mathbf{x} \circ (\bar{\mathbf{u}}_i - \mathbf{a}_i); \mathbf{x} \circ (\mathbf{a}_i - \underline{\mathbf{u}}_i)]$, $\tilde{\mathbf{a}}_i$ by $[\mathbf{h}_i^+; \mathbf{h}_i^-]$, $\mathbf{u}_i$ and $\bar{\mathbf{u}}_i$ by $\mathbf{1}$, and $\underline{\mathbf{u}}_i$ by $\mathbf{0}$. All models with budget of uncertainty style constraints may therefore be classified as a single-connecting-constraint polyhedral problem.

This relationship is important, as it alludes to the wide variety of potential applications of the proposed algorithm. The budget of uncertainty problem, which is a more restrictive formulation, appears in fields as diverse as scheduling, power systems, healthcare, resource planning, and more (Gorissen et al., 2015; Dai et al., 2016; Chen et al., 2018; Yan and Kung, 2018; Polinder et al., 2019; Mahmoudzadeh et al., 2020; Dehghani Filabadi and Mahmoudzadeh, 2022).

2.4 Cutting-Plane Procedures for RMIO

The CP approach for the RMIO problem differs from the RLO, in that it will be solved many times, throughout a branch-and-cut framework. Unlike the RC implementation which automatically encodes all the robustness information, violations may be found all

throughout the solution tree, and it is at the discretion of those implementing the model to determine where and when these cuts are necessary. See Figure 2.1 for a depiction of the two solution methodologies. Note, a given solver may automatically employ additional heuristic cuts, e.g., Gomory cuts for the RC method, making it also technically a branch-and-cut method. These cuts, however, are for efficiency only, not a requirement for solving the problem.

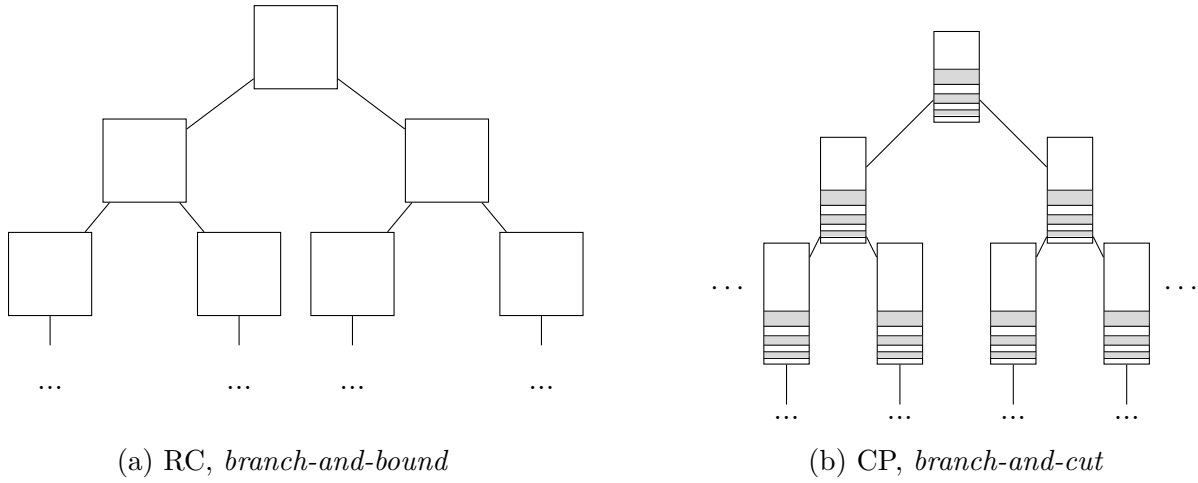


Figure 2.1: RMIO branching solution methodologies. Note the shaded grey region represents solver time, as the CP procedure is iterative. The number of constraints may get longer at each iteration, but unlike RC, the width (i.e., number of variables) in the problem stays the same.

The RC method benefits from having the entire solution space encoded from the start, meaning infeasibility may be detected (in the entire problem or within a particular branch) earlier than in the CP method. Conceptually, if a nominal model is considered, where all constraints are known, the RC should have roughly the same or a smaller tree size, in terms of breadth and depth, seeing as fewer solutions are admissible. This is not guaranteed however, as the added constraints, may also expose deeper integer corner points, and the RC problem has more variables and constraints compared to the nominal model.

This section first details the current state-of-the-art in this field, in Section 2.4.1, a

comment on the heuristic proposed in [Bertsimas et al. \(2016\)](#) is made in Section 2.4.2, and finally, an additional solution approach is proposed in Section 2.4.3.

2.4.1 Current Literature on CP for RMIO

The general branch-and-cut procedure used to find $\mathbf{x}^*$ is detailed in Algorithm 3. For the most part, it is like a branch-and-bound procedure, however, there is a cut set $\mathcal{C}$, that may or may not be referenced at different points throughout the algorithm. Non-standard additions to the algorithm are indicated by the dashed lines.

Algorithm 3 General Branch-and-Cut CP Solution Algorithm

```

1:  $N \leftarrow \text{RMIO}^0$ , where  $\text{RMIO}^0$  uses a valid subset  $\mathcal{U}_i^0 \subset \mathcal{U}_i \forall i \in \mathcal{I}$ .  $\triangleright$  Typically nominal
2:  $\mathbf{x}^* \leftarrow \text{NULL}$ ,  $z^* \leftarrow \infty$ 
3:  $\mathcal{C} \leftarrow$  any initial CP constraints
4: while  $N \neq \text{NULL}$  do
5:    $N_{\text{curr}} = N.\text{next}()$ .
6:   if !infeasible( $N_{\text{curr}}$ ) then
7:     Solve  $N_{\text{curr}}$  to obtain  $\mathbf{x}$  and  $z$ .
8:     if  $z \geq z^*$  then
9:       Continue.  $\triangleright$  Node pruned
10:    Optional: Check for cut violations in  $\mathcal{C}$ 
11:    [Add new violations to  $\mathcal{C}$  (Algorithms 1)  $\triangleright$  Fischetti and Monaci \(2012\)]
12:    if  $\mathbf{x}^1$  is all integer then
13:      [Add new violations to  $\mathcal{C}$  (Algorithm 2)  $\triangleright$  Bertsimas et al. \(2016\)]
14:      Check for cut violations in  $\mathcal{C}$ 
15:      if no violations in  $\mathcal{C}$  then
16:         $\mathbf{x}^* \leftarrow \mathbf{x}$ ,  $z^* \leftarrow z$ 
17:      else
18:        [Optional repair algorithm.  $\triangleright$  Bertsimas et al. \(2016\)]
19:        Continue.  $\triangleright$  Node pruned
20:    else
21:      Add branched elements to  $N$ 

```

Both [Fischetti and Monaci \(2012\)](#) and [Bertsimas et al. \(2016\)](#) start with an empty

cut set $\mathcal{C}$. Fischetti and Monaci (2012) appear to run step 11 on all nodes, regardless of integrality, whereas Bertsimas et al. (2016) only run a violation check (Algorithm 2) on each uncertain constraint at nodes where condition 12 is met (i.e., $\mathbf{x}^1$ is integer), with the exception of at the root node, where they sometimes run the full Algorithm 1 to guide the search process (here they specify that they run the model “until the fractional solution is feasible with respect to all uncertain constraints”, which is taken to mean the cuts are added to the model itself, since cut set $\mathcal{C}$ is only checked at incumbents). This implies that visually, the Fischetti and Monaci (2012) method looks more like Figure 2.1b, whereas the Bertsimas et al. (2016) method looks like Figure 2.2a if cuts are added to the root node or Figure 2.2b if they are not. In their article, Bertsimas et al. (2016) conclude that root cuts are beneficial, however, they note that the benefit primarily stems from infeasibility detection, rather than convergence advantages.

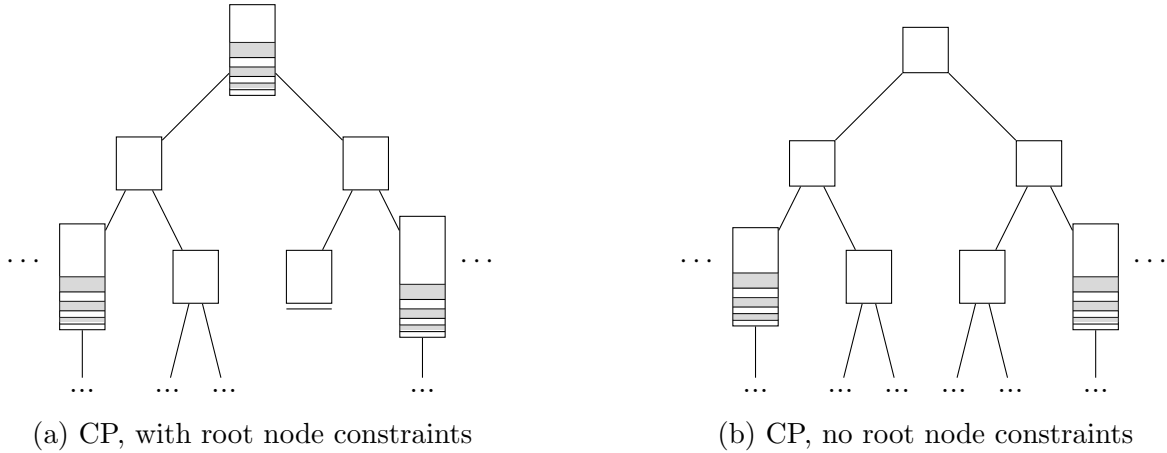


Figure 2.2: Two of the CP implementation methods proposed by Bertsimas et al. (2016). The no root constraint problem does not grow as fast, but it also may not prune results as quickly.

Note that in all diagrams, the growing cut sets $\mathcal{C}$, are represented as part of the node, as they are added and (usually) performed on open nodes. In practice, however, they are implemented as lazy constraints, which are stored in a separate list $\mathcal{C}$ by both CPLEX

and Gurobi (as of April, 2024). This counter-intuitively increases overall efficiency since only a single problem instance must be stored in memory, and referenced for feasibility. This feature, in turn, makes keeping $\mathcal{C}$ as small as possible extremely important (Gurobi Optimization, LLC, 2024).

2.4.2 The Bertsimas et al. (2016) Repair Heuristic

Bertsimas et al. (2016) propose a heuristic to repair an incumbent solution that is found to be integral but infeasible to the previously unknown robust elements of the uncertainty set. In this heuristic, the integral set of variables $\mathbf{x}^1$ is fixed to the values found in the node, and the rest of the model is solved to find $\mathbf{x}^2$ using the typical CP process (i.e., Algorithm 1). The algorithm may be summarized as follows:

Algorithm 4 Bertsimas et al.’s Repair Heuristic

- | | |
|--|---|
| 1: input: $\mathbf{x}, z$ at current node N ; $\mathbf{x}^1 \in \mathbb{Z}^{n_1}$. | ▷ Integer solution violates uncertainty |
| 2: fix $\mathbf{x} := \mathbf{x}^1$ | |
| 3: run Algorithm 1, solving for $\mathbf{x} \in \mathbf{x}^2$ | ▷ Reduced variable CP algorithm |
| 4: if feasible then | |
| 5: report $\mathbf{x}, z$ to solver | ▷ Potentially updating the incumbent |
-

The idea is that solving this model will get to an incumbent faster. In their experimentation, however, they found that not including the heuristic was typically beneficial for polyhedral uncertainty sets.

It pays to examine what each node might look like before it gets passed to Algorithm 4. In Figure 2.3 all 4 possible node-scenarios are shown, given that $x_1 \in \mathbf{x}^1$ and $x_2 \in \mathbf{x}^2$. If an otherwise feasible integral solution node fails to accomodate uncertainty constraints, the node either becomes infeasible when the constraints are added (Figure 2.3b), the current integer value remains feasible and optimal (Figure 2.3c), or the extra cuts expose more integral points deeper within the solution space, either making the current point suboptimal (Figure 2.3d), or infeasible (Figure 2.3e).

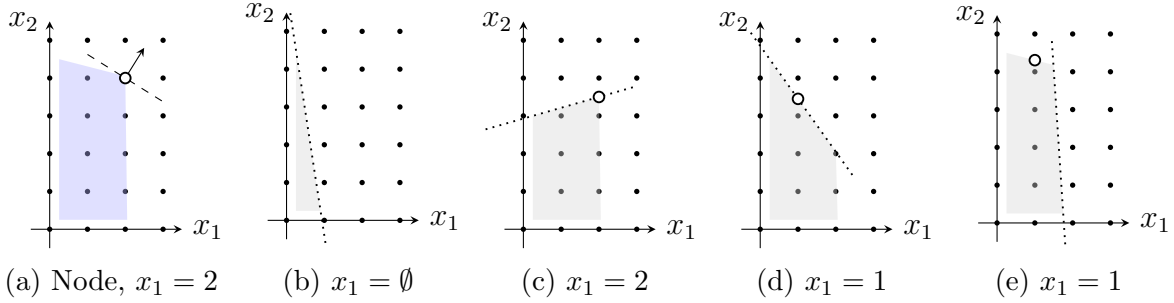


Figure 2.3: Repair heuristic possible outcomes. Given initial node outcome (a) and assuming x_1 must be integer, the image depicts the three possible outcomes of a rejected node: (b) no feasible solution, (c) the same integral values are feasible and optimal, (d) the same integral values are feasible but not optimal, and (e) a new integral value.

In terms of the branch-and-cut algorithm, Algorithm 4 only helps find the case that the solution remains optimal, as in Figure 2.3c, without ruling out the case of a better solution, as in Figure 2.3d. If the solutions are very close, like Bertsimas et al. (2016) predicted, the branch-and-cut algorithm should rapidly find the new solution as well, once all cutting planes have been added to set $\mathcal{C}$ (although more nodes may be needed to get back to integrality). The addition of Algorithm 4 does not prevent cuts from being added to $\mathcal{C}$, and in cases where it takes longer to get back to the initial integer feasible solutions, these cuts may have revealed more promising facets of the decision space, as in Figure 2.3d, meaning it is potentially more efficient to explore those regions earlier. In other words, since this area of the decision space is already being explored by the branch-and-cut algorithm, it is understandable, if not expected, that the extra time spent duplicating and rerunning the model would generally have diminishing returns.

2.4.3 Alternative Implementation for CP in RMIO

Given that the previous sections have established the most efficient way to run CP for RMIO is by adding as few cuts as possible to $\mathcal{C}$, the following implementation suggestions are proposed at the root node.

1. Do not add any additional cuts to the main model, only populate $\mathcal{C}$.
2. Rather than running the full [RLO CP](#), run the first separation, i.e., iteration of [Algorithm 1](#) only, to expose initial infeasibility concerns (adding those to $\mathcal{C}$), as illustrated in [Figure 2.4](#).

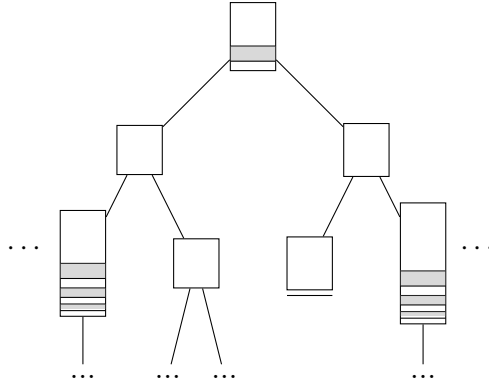


Figure 2.4: CP implementation with a single iteration at the the root node.

These features may provide a stronger start for the model, seeing as,

1. Modern solvers are optimized to handle lazy constraints in an intelligent way.
2. [Algorithm 1](#)'s convergence relies on the fact that higher violations are typically penalized in earlier iterations. Thus, more general and relevant constraints should appear in this first iteration, and time and memory can be saved by not exploring further, while ideally ruling out infeasibility via the worst-case violations.

This single, lazy root-node setup is hypothesized to outperform in situations where a feasible solution cannot be guaranteed, whereas no root cuts, i.e., [Figure 2.2b](#) might be beneficial for problems where a solution is known to exist.

Note that for problems with a warm start, a set of cuts *must* be added to $\mathcal{C}$ at the root node as in [Figure 2.4](#). Rather than being related to the [LP](#) relaxation of the [RMIO](#),

however, these cuts must indicate the worst-case realizations of uncertainty based on all realized constraints in the warm start. In other words, a user must apply Algorithm 2 to every constraint when a warm start has been input to populate $\mathcal{C}$, or else the branch-and-cut algorithm may be infeasible. A user could choose to input additional constraints from the LP relaxation into the cut set as well, as in Figure 2.2a, however, this will increase the size of an already populated initial $\mathcal{C}$.

2.5 Application to Radiation Therapy

RTTP, and more specifically IMRT, provides a real-world example of uncertainty that may be modelled using the single-connecting constraint form of polyhedral uncertainty set. Breathing motion can be effectively captured using this form of uncertainty set, which captures the unknown proportion of time spent in each phase of the breathing cycle throughout treatment (between inhale and exhale). Suppose a patient has $|\mathcal{K}|$ distinct breathing phases and these phases are incorporated into the dose influence matrix, $D_{v,k,b}$. Then, the uncertain proportion of time $\tilde{p}_{v,k}$ spent in each phase, at each voxel, has the following polyhedral uncertainty set,

$$\mathcal{P}_v = \{\tilde{\mathbf{p}}_v \in \mathbb{R}^{|\mathcal{K}|} \mid \underline{p}_{v,k} \leq \tilde{p}_{v,k} \leq \bar{p}_{v,k} \quad \forall k \in \mathcal{K}, \mathbf{1}^T \tilde{\mathbf{p}}_v = 1\} \quad \forall v \in \mathcal{V}, \quad (2.9)$$

where $\underline{\mathbf{p}}_k \geq \mathbf{0}$ and $\bar{\mathbf{p}}_k \leq \mathbf{1}$, for all phases k . Note, when $\mathbf{p}_v$ is certain, it is denoted with no tilde and referred to as the nominal probability. The uncertainty set, on the other hand, mirrors equation (2.5) almost exactly, only differing in an equality, vs., an inequality constraint, since a realized breathing cycle must always sum to 1. It is trivial to show, however, that replacing the equation with $-\mathbf{1}\tilde{\mathbf{p}}_v \leq -1$ will result in an identical uncertainty set in the final optimization problem, seeing as the inequality constraint must always be tight. This uncertainty set can then be used to apply the methods discussed in previous sections to RTTP.

In this section, the impact of **RO** on the simple **FMO** version of the problem is discussed (Section 2.5.1), followed by an overview of the more realistic **DAO** approach with **RO** (Section 2.5.2), then a simple warm-start heuristic for **DAO** is provided (Section 2.5.3).

2.5.1 Robust linear radiation therapy (RLO)

Give the uncertainty sets for each voxel, $\mathcal{P}_v$, the robust uncertain motion is integrated into the original (**FMO**) model as follows,

$$\begin{aligned} \min_{\mathbf{w} \in \mathbb{R}_+^{|\mathcal{B}|}} \quad & \sum_{s \in \mathcal{H}, \mathcal{T}} \sum_{v \in \mathcal{V}_s} \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} p_{v,k} \frac{D_{v,b,k}}{|\mathcal{V}_s|} w_b \\ \text{s.t.} \quad & \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} \tilde{p}_{v,k} D_{v,b,k} w_b \geq L_v \quad \forall \tilde{\mathbf{p}}_v \in \mathcal{P}_v, v \in \mathcal{V}_{\mathcal{T}}. \end{aligned} \quad (2.10)$$

In terms of uncertainty, model (2.10) is visibly similar to model (**RLO**), however, to take on the exact format, some minor reformulation is still needed. A set of auxiliary variables defined as $d_{v,k} = \sum_{b \in \mathcal{B}} D_{b,v,k} w_b$ is added to the formulation, as follows,

$$\begin{aligned} \min_{\mathbf{w} \in \mathbb{R}_+^{|\mathcal{B}|}} \quad & \sum_{s \in \mathcal{H}, \mathcal{T}} \sum_{v \in \mathcal{V}_s} \sum_{k \in \mathcal{K}} \frac{p_{v,k}}{|\mathcal{V}_s|} d_{k,v} \\ \text{s.t.} \quad & \sum_{k \in \mathcal{K}} \tilde{p}_{v,k} d_{k,v} \geq L_v \quad \forall \tilde{\mathbf{p}}_v \in \mathcal{P}_v, v \in \mathcal{V}_{\mathcal{T}}, \end{aligned} \quad (2.11a)$$

$$\sum_{b \in \mathcal{B}} D_{b,v,k} w_b - d_{k,v} = 0 \quad \forall k \in \mathcal{K}, v \in \mathcal{V}_{\mathcal{T}}. \quad (2.11b)$$

This modified model serves to disentangle the uncertain and certain constraints, leading to the desirable format where each constraint is either singular (constraints 2.11b) or has a polyhedral uncertainty set around the entire coefficient (constraints 2.11a). The variable equivalence, with model (**RLO**) then becomes $\mathbf{x} = w_b \cup d_{v,k}$, with an associated cost vector of $\mathbf{c} = \left[\mathbf{0}_b; \frac{1}{|\mathcal{V}_{\mathcal{H}}|} [\mathbf{p}_{\times|\mathcal{V}_{\mathcal{H}}|}]; \frac{1}{|\mathcal{V}_{\mathcal{T}}|} [\mathbf{p}_{\times|\mathcal{V}_{\mathcal{T}}|}] \right]$, where $\mathbf{0}_{dim}$ is a 0-vector of length dim , and $\mathbf{v}_{\times rep}$ is vector $\mathbf{v}$ repeated rep times.

2.5.2 Robust mixed integer radiation therapy (RMIO)

In considering the subsequent steps or constraints to account for mechanical delivery considerations, a brief overview is given, then interested readers are directed to [Ripsman et al. \(2022\)](#) or [Romeijn et al. \(2005\)](#) for much more detailed models and descriptions. In general, realistic treatment delivery is concerned with the beamlet $b \in \mathcal{B}$ location, as well as the dose delivered to target and healthy tissue. For a machine-compatible DAO model, a deliverable IMRT treatment must break down into a set of deliverable apertures, wherein, all active beamlets within a given aperture receive the same dose (recall Figure 1.4).

Since the beam can be delivered from multiple angles, the beam may be broken down by a triplet identifying the row, column and angle $\{q, n, \theta\}$ of each beamlet. If the problem is simplified by assuming only one aperture is available per angle, the linear model (2.11) can be modified by associating binary variables w_b^1 , with each of the existing linear variables w_b . This modification will allow for the enforcement of dose uniformity and adjacency constraints via the binary variables. Further, the beamlets in each angle $\theta \in \Theta$ can be partitioned into their respective angles using the notation $\mathcal{B}_\theta$. Rows and columns may be represented by sets $\mathcal{Q}$ and $\mathcal{N}$, respectively, where $\mathcal{Q} := \{1, \dots, |\mathcal{Q}|\}$ and $\mathcal{N} := \{1, \dots, |\mathcal{N}|\}$. The small DAO model looks as follows,

$$\min_{\mathbf{w}, \mathbf{d}, \mathbf{f} \in \mathbb{R}_+, \mathbf{w}^1, \mathbf{l}, \mathbf{r} \in \mathbb{Z}_2} \sum_{s \in \mathcal{H}, \mathcal{T}} \sum_{v \in \mathcal{V}_s} \sum_{k \in \mathcal{K}} \frac{p_{v,k}}{|\mathcal{V}_s|} d_{k,v} \quad (2.12a)$$

$$\text{s.t.} \quad \sum_{k \in \mathcal{K}} \tilde{p}_{v,k} d_{k,v} \geq L_v \quad \forall \tilde{\mathbf{p}}_v \in \mathcal{P}_v, v \in \mathcal{V}_{\mathcal{T}}, \quad (2.12b)$$

$$\sum_{b \in \mathcal{B}} D_{b,v,k} w_b - d_{k,v} = 0 \quad \forall k \in \mathcal{K}, v \in \mathcal{V}_{\mathcal{T}}, \quad (2.12c)$$

$$w_b \leq f_{\theta} \quad \forall b \in \mathcal{B}_{\theta}, \theta \in \Theta, \quad (2.12d)$$

$$w_b \leq M w_b^1 \quad \forall b \in \mathcal{B}_{\theta}, \theta \in \Theta, \quad (2.12e)$$

$$w_b \geq f_{\theta} - M(1 - w_b^1) \quad \forall b \in \mathcal{B}_{\theta}, \theta \in \Theta, \quad (2.12f)$$

$$l_{q,n+1,\theta} \geq l_{q,n,\theta} \quad \forall n \in \mathcal{N} \setminus \{|\mathcal{N}|\}, q \in \mathcal{Q}, \theta \in \Theta, \quad (2.12g)$$

$$r_{q,n,\theta} \geq r_{q,n+1,\theta} \quad \forall n \in \mathcal{N} \setminus \{|\mathcal{N}|\}, q \in \mathcal{Q}, \theta \in \Theta, \quad (2.12h)$$

$$w_{q,n,\theta}^1 = -1 + l_{q,n,\theta} + r_{q,n,\theta} \quad \forall n \in \mathcal{N}, q \in \mathcal{Q}, \theta \in \Theta. \quad (2.12i)$$

Constraints (2.12b–2.12c) are from the [RLO](#) model, constraints (2.12d–2.12f) are either/or constraints that enforce uniform dose to the apertures, with the value for M being selected as a constant based on a clinical dose upper-bound. Constraints (2.12g–2.12i) enforce continuity within the left and right leaves, which means only the beamlets in between the two may be active. $\mathbf{l}$ and $\mathbf{r}$ are auxiliary binary variables created for this purpose, with the exact same dimensions as $\mathbf{w}$ and $\mathbf{w}^1$, indexed over rows, columns and angles.

These three extra binary variable-sets make even this very stripped-down version of the model quite large to solve, seeing as each adds hundreds of additional decisions into the model. Note that the robust elements of the model stay the same as in the [RLO](#) case, since the relationship to uncertainty remains between the continuous variables and their coefficients.

2.5.3 Warm starting the RMIO

Similar to [Ripsman et al. \(2022\)](#), a warm start is recommended for the model. Unlike that manuscript, a simplified approach is taken, reducing their 3-step process to 2-steps. In step 1, model (2.12) is linearized so only constraints (2.12b)–(2.12d) are needed and a penalty term is simultaneously added to the objective function, to get the resultant $\mathbf{w}^s$. Note, there is no benefit in keeping the remaining linearized constraints (2.12e)–(2.12i) in the model, since, in the absence of binary variables, $\mathbf{l}$, $\mathbf{r}$ and $\mathbf{w}^1$ do not restrict the domain of $\mathbf{w}^s$ in any meaningful way. This is because it is expected that $M \gg f_\theta$ and therefore, fractional values of the binary variables would give near-absolute freedom to $\mathbf{w}^s$ (e.g., $\mathbf{l}, \mathbf{r} = \mathbf{0.75} \implies \mathbf{w}^1 = \mathbf{0.5}$).

Again, in contrast to [Ripsman et al. \(2022\)](#), all variable values in the DAO model are not assigned to their integer values, rather, a robust and integer-feasible $\mathbf{w}^s$ vector is found and modern solvers are assumed to correctly fix the $\mathbf{w}^1$, $\mathbf{r}$ and $\mathbf{l}$ integer variables given the $\mathbf{w}^s$ vector on their own.

The warm-start model retains the original objective function (2.12a) alongside tuning parameter α and max dose parameter (per angle) f_θ . Part 1 of the warm-start process then becomes,

$$\begin{aligned} \min \quad & \alpha \sum_{\theta \in \Theta} f_\theta + (1 - \alpha) \cdot \text{2.12a} \\ \text{s.t.} \quad & \text{(2.12b)} - \text{(2.12d)}. \end{aligned} \tag{2.13}$$

In step 2, any gaps are filled in using a heuristic. Since the robustness is on a $\geq$ constraint, the coefficients are positive and the value of $\mathbf{w}^s$ is always the same or increased, there is no fear of constraint violation in a secondary algorithm, only objective-function-degradation. The full warm-start algorithm is detailed in Algorithm 5.

Recall that for feasibility, i.e., to ensure that the solver will not tighten the solution past allowable bounds in the warm start, the worst-case uncertainty realization from Algorithm 2 for every constraint containing w_b^s must be added either to the root or to cut

Algorithm 5 Simple warm start heuristic

```

1: choose  $\alpha$  value in  $(0,1)$ 
2: solve model (2.13) using either an RC, SP or Algorithm 1  $\rightarrow w_b^{s1}, f_\theta^{s1}$ 
3:  $w_b^s \leftarrow \mathbf{0}$ 
4: for angle  $\theta \in \Theta$  do
5:   for row  $q \in \mathcal{Q}$  do
6:      $n_1 \leftarrow \min_n \{w_{q,n,\theta}^{s1} > 0\}, n_2 \leftarrow \max_n \{w_{q,n,\theta}^{s1} > 0\}$ 
7:     for column  $n \in \{n_1, \dots, n_2\}$  do ▷ if no active beamlets, range is empty
8:        $w_{q,n,\theta}^s \leftarrow f_\theta^{s1}$ 
9: return  $w_b^s$ 

```

set $\mathcal{C}$ before running a branch-and-cut solver.

2.6 Numerical Results

The following section uses [IMRT](#) treatment data to compare the various algorithm approaches. The datasets are the four-dimensional computed tomography scans of three left-sided breast cancer patients, provided by a local cancer centre. The healthy organ considered was the heart, while the target region was the full breast region. Each target region dataset was down-sampled, i.e., every 32nd voxel was included; a process that both saves on memory and makes the data more comparable to clinical granularity ([Mahmoudzadeh et al., 2015](#)). These datasets are available for download through the Supplemental Material of [Ripsman et al. \(2022\)](#).

For each patient, $|\mathcal{K}| = 5$ breathing phases were considered, with associated uncertainty bounds $\underline{\mathbf{p}} = [0.025, 0.025, 0.025, 0.025, 0.4]$ and $\bar{\mathbf{p}} = [0.225, 0.225, 0.225, 0.225, 0.6]$, based on clinical guidelines. The nominal proportion $\mathbf{p}$ is the mean of the upper and lower bounds on the uncertainty set. Note that because the resulting elements of $\mathbf{p}$ sum to one, they may be used to populate $\mathcal{P}^0$ for each voxel in the first phase of all [CP](#) heuristics. Violations in all [CP](#) algorithms were detected at a 1e-5 level.

The results for the **RLO** version of the problem, i.e., model (2.11), are shown in Section 2.6.1, while the computational results for the **RMIO** version of the problem are presented in Section 2.6.2

2.6.1 RLO model results

In the following section, three model variants are compared: RC, gCP and cfCP. The first variant, RC, is the robust counterpart model (2.3), the second, gCP, is the general (optimization-based SP) cutting-plane algorithm detailed in Section 2.3.1, and the third, cfCP, is the proposed cutting-plane algorithm with the closed-form sub problem described in Algorithm 2. Each model was implemented for the radiation therapy treatment problem using C++ and CPLEX 22.1.1. Every algorithm was run using equal memory allotment for a given patient on the remote Graham cluster on Compute Canada¹.

Patient	Model	Rows	Columns	Iterations	Avg. Run Time (s)		$\mathbf{z}^*$
		First, Last			Full Model	Sub Problem	
A	RC	6300	7820	-	464.11	-	46.9526
	gCP	1050, 2317	1520	8	48.26	14.78	
	cfCP	1050, 2321	1520	8	38.44	10.01	
B	RC	13776	16168	-	4966.76	-	64.4719
	gCP	2296, 5436	2392	11	2738.92	60.12	
	cfCP	2296, 5427	2392	9	2626.24	38.11	
C	RC	19008	21124	-	2142.82	-	56.5723
	gCP	3168, 5441	2116	8	2690.27	53.74	
	cfCP	3168, 5418	2116	9	2522.50	44.55	

Table 2.1: Summary metrics for the RC, gCP and cfCP models run on clinical datasets. The fastest average (of five) runtimes are indicated with bold text. All 3 plans for each patient were run on the same node for fairness for each of the five runs.

Table 2.1 summarizes the metrics for each of the three solution methodologies, applied

¹www.computecanada.ca

to each patient case-study. The patients are listed from smallest (A) to largest (C) number of voxels in the target region, which translates to the number of constraints in the robust problem. Patient A was run on 4 GB of memory, patient B on 12 GB and Patient C on 16 GB for all 3 trials – each being the minimum allowable for the RC case. While the general and closed-form CP plans only need 4 GB to run for patients A and B and 8 GB for C, they were run in sequence on the same cluster node, with the same memory allocation to ensure a fair comparison.

In order to report consistent runtimes in the table, all solution methodologies were run five times for each patient and then averaged, in order to control for cluster speed fluctuations. Since there is no stochasticity between runs, each iteration of a given algorithm gives identical outputs, with the only possible variation being run-time. Similarly, RC, gCP and cfCP all return plans with an identical objective function value, z^* for a given patient. The plans themselves may differ, however, due to the existence of multiple optimal solutions.

At first glance, the major discrepancy between the three model types lies in the overall speed of both CP methods, relative to that of the RC reformulation. Moving from the RC formulation to the gCP had nearly a $10\times$ speed up for patient A, and more than $10\times$ improvement for the cfCP. For patient B, the speed-up was closer to $2\times$ for both gCP and cfCP, whereas for patient C, the CP algorithms actually ran 5-10 minutes slower than the RC. They are all on similar orders, however, and it likely just means that the master problem was slightly more difficult to solve for that particular problem instance.

The closed form cfCP outperformed the general CP algorithm for every problem instance, both in terms of total run time and isolated total sub problem runtime. The distinction between SP runtimes becomes even more clear when a given cfCP iteration of the SP is compared to the corresponding gCP iteration, as in the box plot in Figure 2.5. Throughout the five runs of each algorithm, the SP times are shown to have no overlap in a given iteration, as the cfCP SP runs significantly faster. The plots also show the difference in number of iterations for the two models as well, which is interesting, considering the

same problem is being solved at each iteration.

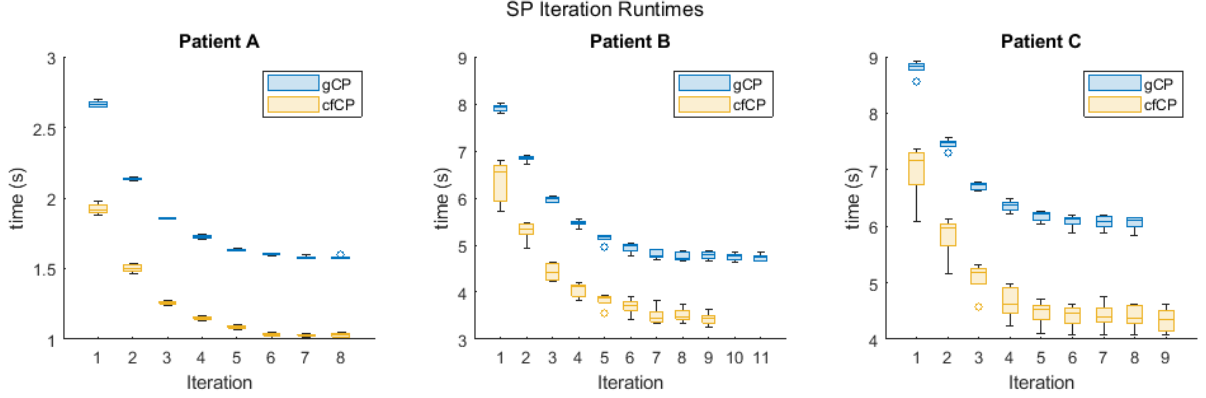


Figure 2.5: Boxplot of the SP runtimes for both CP models, for each patient.

The cause of the discrepancy between the number of iterations may be due to multiple optimal solutions for a given sub problem (this would occur if multiple phases have equal weightings, meaning there are multiple “worst”-violating constraints, each leading to a different master problem formulation). The second explanation could be precision from within the solver in the case of gCP, where the sub problem is an optimization. Solvers only guarantee a certain degree of precision overall, and individual variables from within a solver’s output are being examined to a fairly high degree of precision. Since the cfCP is exact, a higher degree of precision can be expected, as nothing is lost in its return.

The difference between the two algorithms is further highlighted in Table 2.2, where the number of constraints added at each iteration is indicated for the two CP models. The asterisk represents where the solution converged to four decimal places, and the vertical line in each row of the table shows where the number of constraints started differing between the two algorithms. This could either be where multiple optimal solutions are first encountered, or a rough indication of where the precision of the LP approach in the gCG model starts to suffer. Note that the discrepancy between algorithm convergence and solution convergence represent opportunities for even faster run-times given better tuning. E.g., if violations were only detected up to the $1e-4$ level, this would have likely led to

faster runtimes for both CP algorithms, at the cost of very minimal violation allowance.

Case	Model	Constraints Added By Iteration												Total
		0	1	2	3	4	5	6	7	8	9	10	11	
A	gCP	1050	619	326	174	86	39	19*	4	0				2317
	cfCP	1050	619	326	171	92	47	8*	2	0				2321
B	gCP	2296	1229	895	516	280	143	55	18*	2	1	1	0	5436
	cfCP	2296	1229	895	516	279	136	59	13*	4	0			5427
C	gCP	3168	1158	620	285	138	53	16*	3	0				5441
	cfCP	3168	1158	620	285	116	53	13*	3	2	0			5418

Table 2.2: Divergence of the two CP algorithms as the violations get closer to 0. A vertical line is placed to the left of the divergence, and an asterisk is used to indicate the iteration at which the optimal solution was found to 4 decimal places.

These results make a strong argument for the use of the cfCP model for solving robust IMRT planning problems (and by extension, using the closed-form approach to solve single-constraint [RLO](#) problems in general). In addition to being a simple algorithm to implement, the closed-form SP approach has stronger guarantees of correctness, requires fewer computational resources and may provide more stable convergence for the [RLO](#). While in some cases, the RC model may outperform both CP processes in terms of run time, the lower demands on memory, and lack of reformulation requirement remain present with CP approaches. Relative to the gCP algorithm, cfCP is clearly the preferable candidate as well.

2.6.2 RMIO model results

Incorporating the basic delivery constraints of an extended [RMIO](#) model (2.12), leads to much longer runtimes than the several-hour [RLO](#) version of the problem. Since apertures are equated to angles for this implementation, it is relevant to note that the case study at hand is a tangential breast cancer case, meaning that $|\Theta| = 2$ opposing angles are used to

provide full coverage of the target region. The M in model (2.12) is set to 1,000,000 Gy, which far exceeds the prescribed dose of 42.4 Gy.

Four model variants are run: RC, CP, CPn1, CPnAll. RC is once again the robust counterpart version of the model, CP is the cutting plane method with no additional root node constraints (Figure 2.2b), CPn1 is the proposed single iteration at the root node (Figure 2.4), and finally, CPnAll is the full first-node RLO CP implementation version (Figure 2.2a). Once again C++ and CPLEX (22.1.1) are used to implement these models, this time, controlling for CPU fluctuations by running each model variant with a cap of 63,000 seconds, on the same dedicated 32-thread Compute Canada cluster node. CPLEX was allowed to run on its default settings, running as many of its typical heuristics as possible (some of which are automatically switched off due to the use of lazy callbacks). Any root node cuts were run in a duplicate linear model and then pre-loaded into the RMIO as lazy constraints. This process takes only a few minutes on the large cluster nodes, and is thus considered a pre-processing step and not counted in the total runtime.

When all four models were run using the lazy callback setting, in the 17.5 hours allotted, no RC model was able to return an incumbent for any of the three patient cases. This shows that disabling dual reductions, pre-solve reductions and repeat pre-solve have a significant impact on both methods, in terms of finding a solution. When the pre-solve settings were re-instituted for the RC models, the RC model found an incumbent solution for all three patients, and failed to move out of this local optima for the remainder of the solution time. The results are reported in Table (2.3).

These outputs seem to indicate that the pre-solve processes are extremely important for getting a good incumbent, with only patient C finding a better solution in the allotted time using the proposed single-iteration of cuts approach (Figure 2.4), i.e., CPn1, while the rest performed best with the RC with pre-solve setup.

In terms of the CP setup, the single node CPn1 performed best for patients B and C, but lost out to no starting lazy constraints, i.e., the plain CP run for patient A. Overall,

Patient	Model	Pre-solve	Rows	Columns	Final $ \mathcal{C} $	z_{Final}	Reported Gap
A	RC	No	15,260	12,382	-	-	-
	RC	Yes	15,260	12,382	-	70.84	33.72%
	CP	No	10,010	6,082	23	79.41	41.06%
	CPn1	No	10,010	6,082	638	80.60	41.96%
	CPnAll	No	10,010	6,082	1284	80.65	41.92%
B	RC	No	27,944	23,346	-	-	-
	RC	Yes	27,944	23,346	-	102.39	37.03%
	CP	No	16,464	9,570	11	684.72	90.77%
	CPn1	No	16,464	9,570	1,240	420.98	84.99%
	CPnAll	No	16,464	9,570	3,119	492.64	87.17%
C	RC	No	31,520	27,474	-	-	-
	RC	Yes	31,520	27,474	-	95.5755	40.81%
	CP	No	15,680	8,466	34	100.48	44.12%
	CPn1	No	15,680	8,466	1,192	90.45	37.92%
	CPnAll	No	15,680	8,466	2,281	108.21	48.11%

Table 2.3: Summary metrics for the RC, CP, CPn1 and CPnAll models run on clinical datasets.

this indicated that in the current version of CPLEX, the best option for the radiation therapy [RMIO](#) remains the RC approach.

Next, the warm-start approach (Algorithm 5) was applied to each patient. Following the warm start, each model was allowed to run for another 17.5 hours, as in the previous iteration. The warm start results are reported in Table 2.4. Note, $\alpha = .5$ was used to obtain a well-balanced solution for all trials. One observation from this table, is that in a maximum of roughly 10 minutes for the RC model, under 3 minutes for the CP, objective function values better than any of the 17.5 hour runs above were obtained. In other words, the heuristic warm start was found to dominate the original models with no warm starts.

The performance of the RC and CP methods given a warm-start is reported in table 2.5. Note in the CP case, the discussed cuts from the warm-start solution were added to prevent CPLEX from tightening the problem beyond the true robust feasible region, rather than

Patient	Iterations		Total Runtime (s)		z_{WS}	$z_{BestNoWS}$	Improvement
	RC	cfCP	RC	cfCP			
A	-	6	94.0	36.0	60.18	70.84	15.05%
B	-	7	446.0	166.0	101.70	102.39	0.67%
C	-	6	619.0	148.0	88.00	90.45	2.71%

Table 2.4: Warm start summary metrics (no subsequent iterations) run on all clinical datasets. Note, for RC we ran the RC counterpart [RLO](#) constraints in Step 1, and for all CP methods, we ran the cfCP constraints in Step 1, yielding the same objective function value, but potentially different solutions.

the cuts from Table 2.3. These cuts were the worst-case uncertainty realization from the warm start output, i.e., the worst uncertainty realization for each voxel constraint, given the chosen warm start w_b^s , regardless of whether there was slack in the constraint or not. When run without these cuts, the solver failed, as it produced a tightened start solution that violated robustness guarantees at the root. A decision that could be made however, was whether to add the constraints directly on the node (CP-root) or adding them to the cut set $\mathcal{C}$ (CPn1- $\mathcal{C}$).

Patient	Model	Final $ \mathcal{C} $	$z_{Initial}$	Reported Gap	z_{Final}	Reported Gap
A	RC	-	59.93	21.66%	59.93	21.65%
	CP-root	0	59.95	21.83%	59.95	21.82%
	CPn1- $\mathcal{C}$	1050	59.95	22.11%	59.95	21.90%
B	RC	-	100.41	35.79%	100.41	35.79%
	CP-root	0	100.41	36.20%	100.41	36.20%
	CPn1- $\mathcal{C}$	2296	100.41	37.05%	100.05	36.82%
C	RC	-	87.16	35.09%	87.16	35.09%
	CP-root	0	87.15	35.24%	87.15	35.24%
	CPn1- $\mathcal{C}$	3168	87.15	35.60%	87.15	35.54%

Table 2.5: Summary metrics for the RC and CP models (with constraints added to the root as well as the lazy set) run on clinical datasets, with the warm start.

For testing, the RC model was given an RC warm start, whereas a CP warm start was given to the CP problems. However, the major takeaway is there was barely any movement in the solution beyond slight tightening of the warm start for any solution. Patient B had slight improvement in the CPn1- $\mathcal{C}$ model, but even this was quite marginal. This indicates that the MILP model is still quite inefficient for this problem, and if possible, other approaches should be taken. If not, multiple days were needed to run these algorithms to convergence, regardless of whether RC or CP was used. For the models at hand, there is no clear winner between the two, but since the lazy constraints are presently much more difficult-to-implement than the RC, the RC approach with a warm start seems like the correct method for this type of large-scale problem instance.

Overall, for the present instance of uncertainty in RTTP, this study’s findings were in agreement with Fischetti and Monaci (2012). For RLO, CP is preferable to the RC due to its ease of implementation and potential speed and memory improvements. In terms of RMIO, the RC approach is recommended, due to the complexity and lack of apparent benefit when it comes to implementing CP.

2.7 Conclusions

An exact cutting-plane sub problem was introduced for a class of robust linear optimization problems with polyhedral uncertainty sets. This algorithm was shown to generalize previously studied budget-of-uncertainty algorithms, and to work rapidly on linear real-world radiation therapy treatment models. The streamlined algorithm outperformed the general optimization sub problem across all linear problem instances and outperformed most of the robust counterpart instances as well.

The more compact cutting plane implementation struggled with integer instances of the problem. When compared with existing robust counterpart implementations, the cutting plane approaches benefited from fewer solver features in the current CPLEX implemen-

tation, and therefore often under-performed. A warm-start methodology that integrated the closed-form cutting plane sub problem was found to outperform both the robust counterpart and the cutting plane approach. However, when the warm start solution was fed into any algorithm, CPLEX failed to improve on the solution over the course of many hours. For that reason, it was concluded that there was no clear winner in terms of robust counterpart versus cutting-plane integer implementations, but it is highly recommended that users look for their own strong warm start, in the case of large-scale mixed integer models.

Future directions related to this work could include refocusing efforts on the warm-start heuristic and using it as a means to explore the solution space, without requiring mixed integer formulations at all. It would also be interesting to revisit this topic with other solvers, such as Gurobi, or CPLEX in future years, as the quality and speed of both improve greatly on a regular basis. Finally, the methods at hand could be applied to a benchmarking study, similar to the one performed in [Bertsimas et al. \(2016\)](#). The radiation therapy problem explored in this chapter is extremely large, and it is possible that the more light-weight implementations proposed in this study would go further with more manageable problem instance sizes, such as examples in the scheduling or resource allocation domains.

Chapter 3

Light Pareto Robust Optimization with CVaR Extensions

This chapter focuses on the integration of three efficient optimization methods into the [FMO](#) framework for [IMRT](#). The first method is Pareto robust optimization, which improves upon [RO](#) solutions by also optimizing for non-worst-case scenarios. The second method is light robustness which allows for a trade-off between optimality at worst-case and the plan quality in non-worst-case scenarios. The final method is conditional value-at-risk, which is shown to benefit from the light and Pareto additions to [RO](#), while also allowing for radiation-dose considerations to be integrated into the [FMO](#) modelling.

The chapter is broken down as follows, in [Section 3.1](#) the three mathematical methods above are motivated and contextualized, and the chapter's contributions are highlighted. In [Section 3.2](#), the model derivations are made for the basic robust [FMO](#) case. In [Section 3.3](#), the conditional value-at-risk extensions are derived. In [Section 3.4](#) the data and testing methods are outlined. [Section 3.5](#), contains the numerical results and [Section 3.6](#) a discussion of those results. Finally, the chapter is concluded in [Section 3.7](#).

3.1 Introduction

The majority of [IMRT](#) literature in the field of [OR](#) contains two well-known approaches for handling data uncertainty. The first is [stochastic optimization \(SO\)](#) and the second is [RO](#) ([Kåver et al., 1999](#); [Chan et al., 2006](#); [Bortfeld et al., 2008](#); [Unkelbach et al., 2018](#)). [SO](#) uses known underlying distributions to immunize against uncertainty, either pessimistically (worst-case) or for more stylized probabilistic goals, which balance uncertainty mitigation with conservatism. The drawback of [SO](#) is that it requires an awareness of the underlying distribution of a given phenomenon. [RO](#), on the other hand, accounts for both data uncertainty and distributional uncertainty, albeit pessimistically ([Soyster, 1973](#)). Note, that [distributionally robust optimization \(DRO\)](#) is typically seen as a middle-ground between [SO](#) and [RO](#), however, it still requires some distributional information, and effectively collapses back into an [RO](#) approach in set-based uncertainty cases ([Delage and Ye, 2010](#)).

In [IMRT](#), [RO](#) models have been shown to immunize plans against patient setup, dose calculation and organ location uncertainty ([Chu et al., 2005](#); [Olafsson and Wright, 2006](#); [Chan et al., 2006](#)). It has also been used to adaptively reduce conservatism from uncertainty realizations between decision-making periods ([Nohadani and Roy, 2017](#); [Böck et al., 2017](#); [Roy et al., 2022](#); [Ten Eikelder et al., 2022](#)). In relation to breathing motion, it has been well-documented (including in the previous chapter) as an effective method for capturing the phase-based patterns within cancer treatments, using polyhedral uncertainty sets ([Chan et al., 2006](#); [Bortfeld et al., 2008](#); [Chan et al., 2014b](#); [Mahmoudzadeh et al., 2015, 2016](#); [Ripsman et al., 2022](#)). For a comprehensive review of the robust radiotherapy planning literature, the reader is referred to [Unkelbach et al. \(2018\)](#).

While [RO](#) is beneficial for uncertainty mitigation, it also has a key drawback. [RO](#) solutions are often criticized for being overly conservative, as they are only optimized for the worst-case scenario. Oftentimes, the worst-case is unlikely to occur and it is unclear what the performance in non-worst-case realizations would be, or what benefits are being traded off for such extreme guarantees.

Pareto robust optimization (PRO) is a theoretical approach for improving the performance of RO solutions in non-worst-case scenarios Iancu and Trichakis (2014). A PRO solution has the same worst-case performance as an RO solution but may have better performance in other scenarios. PRO is a good fit for FMO problems with uncertainty. In breast cancer IMRT, for instance, while it is important to guarantee that the prescribed target dose reaches the entire target region in the worst-case breathing scenario, such a worst case may be unlikely to occur and it is equally important to deliver a dose that is also optimized for other breathing scenarios (Rahimi, Fahimeh, 2022). In other words, while PRO addresses the issue of completely neglecting non-worst-case behaviour, the issue of being overly conservative for the worst-case largely remains with this approach.

To address the issue of conservatism in RO solutions, light robust optimization (LRO) is a methodology that aims to improve the non-worst case behaviour of a set of secondary objectives. It accomplishes this trade-off through a two-phased optimization approach in which the primary objective is relaxed in a controlled manner (Fischetti and Monaci, 2009). Inspired by this concept, this chapter introduces light Pareto robust optimization (LPRO) solutions for IMRT that further improve the average performance of the PRO treatment plans by slightly giving up on the optimality of plans under the worst case. Unlike the LRO methods, which typically relax robust constraints as well as the primary objective function value, the LPRO model discussed in this work maintains robustness, only relaxing the optimality of the original robust plan by a negligible amount (Fischetti and Monaci, 2009; Cacchiani and Toth, 2012; Schöbel, 2014).

In RTTP, analyzing the trade-off between worst-case optimality and average case performance can be a valuable decision-making tools for clinicians. Moreover, the guarantee that LPRO will not sacrifice robustness in the constraints means clinicians can be more confident in their selected treatment plan. It also increases the importance of accuracy in the constraints, which is why the final set of models in this chapter integrate robust tumour control via conditional value-at-risk (CVaR) constraints into the radiation LPRO

models. These CVaR constraints tend to mirror real-world clinical dose-criteria requirements for RTTP plans by allowing for control of average tail loss of the dose distribution (Mahmoudzadeh, 2015). They have well-behaved mathematical properties, due to their convex, linear structure and have previously been shown to be tractable for dose control in general FMO models (Romeijn et al., 2003, 2006) as well as being extensible to the RO framework (Mahmoudzadeh et al., 2015). They have not previously been integrated into the LPRO framework for RTTP, however.

This is some of the first work to apply the concept of PRO in IMRT and analyze the trade-off between the optimality of robust solutions under the worst case and the treatment quality in the average case of uncertainty. The benefits of the proposed two-stage PRO and LPRO models over the more conventional single-stage RO approach from the literature are demonstrated in this work. It is also the first work to unify all of these elements into the CVaR framework. This chapter demonstrates a proof of concept of the proposed LPRO model for FMO in IMRT using the clinical database of left-sided tangential breast cancer patients with breathing motion uncertainty. Using all of these approaches, the trade-off between the optimality of plans in the worst-case scenario and the plan quality in average non-worst-case scenarios is quantified and discussed.

3.2 Basic PRO and LPRO Models

The model under consideration is a variant of the (RLO) model, this time with the goal of purely minimizing the normalized radiation received by the heart, while still ensuring that every voxel in the CTV receives at least the prescribed dose, L_v , for all $v \in \mathcal{T}$. Therefore, the robust FMO problem with uncertainty in the CTV position, can be written as follows (Bortfeld et al., 2008). The model is referred to as the (RO) model.

$$\begin{aligned}
(\mathbf{RO}) : \quad & \underset{w_b}{\text{minimize}} \quad \frac{1}{|\mathcal{H}|} \sum_{v \in \mathcal{H}} \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} p_{v,k} w_b \\
& \text{subject to} \quad \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} \tilde{p}_{v,k} w_b \geq L_v \quad \tilde{\mathbf{p}}_v \in \mathcal{P}_v, v \in \mathcal{T} \\
& w_b \geq 0, \quad \forall b \in \mathcal{B}
\end{aligned} \tag{3.1}$$

The **(RO)** model ensures that all **CTV** voxels receive the prescribed dose regardless of the breathing **probability mass function (PMF)**. Note that since treatments are typically delivered over multiple sessions or fractions, uncertainty is only considered in the constraints (which must hold for each individual fraction) while the objective function term can be taken as an expectation. This is a standard practice in robust optimization methods for **IMRT** with uncertain constraints. For more details, the reader is referred to the paper by [Bortfeld et al. \(2008\)](#).

Given the same polyhedral uncertainty set around breathing motion studied in the previous chapter, but repeated here for completeness,

$$\mathcal{P}_v = \{\tilde{\mathbf{p}}_v \in \mathbb{R}^{|\mathcal{K}|} \mid \underline{p}_{v,k} \leq \tilde{p}_{v,k} \leq \bar{p}_{v,k} \quad \forall k \in \mathcal{K}, \mathbf{1}^T \tilde{\mathbf{p}}_v = 1\} \quad \forall v \in \mathcal{T},$$

the **(RO)** model, may be reformulated using dual variables, as in the previously introduced **RC** approach, or solved using either of the **CP** approaches provided in Chapter 2. The **(RO)** dual is made explicit, as follows; let $\mathbf{r}$ and $\mathbf{q}$ be the dual variables of the polyhedral set for each uncertain constraint. The robust counterpart of the **(RO)** model is then exactly,

$$\begin{aligned}
& \underset{\mathbf{w}, \mathbf{q}, \mathbf{r}}{\text{minimize}} && \frac{1}{|\mathcal{H}|} \sum_{v \in \mathcal{H}} \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} p_{v,k} w_b && (3.2) \\
& \text{subject to} && \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} [p_{v,k} - \underline{p}_{v,k}] D_{v,k,b} w_b + \sum_{k \in \mathcal{K}} \underline{p}_{v,k} q_v - \sum_{k \in \mathcal{K}} r_{v,k} \geq L_v \quad \forall v \in \mathcal{T}, \\
& && (\bar{p}_{v,k} + \underline{p}_{v,k}) q_v - r_{v,k} \leq \sum_{b \in \mathcal{B}} D_{v,k,b} (\bar{p}_{v,k} + \underline{p}_{v,k}) w_b \quad \forall v \in \mathcal{T}, k \in \mathcal{K}, \\
& && q_v \text{ free}, \quad \forall v \in \mathcal{T} \\
& && r_{v,k} \geq 0, \quad \forall v \in \mathcal{T}, k \in \mathcal{K} \\
& && w_b \geq 0, \quad \forall b \in \mathcal{B}.
\end{aligned}$$

Given the solution to the initial **(RO)**, secondary objectives may be considered. While minimizing the heart dose is a very important for patient quality of life, a secondary goal for the planner might be to reduce overdose to the **CTV**, as a measure of overall toxicity. Let $\mathcal{W}^{\text{RO}}$ be the set of all **(RO)** solutions. Then a **PRO** solution may be defined as an **(RO)** solution $\hat{w}$ that cannot be dominated (Iancu and Trichakis, 2014) by any other **(RO)** solutions in terms of the tumour overdose in all scenarios. That is, $\hat{w}$ is a **PRO** solution if $\nexists w \in \mathcal{W}^{\text{RO}}$ such that

$$\begin{aligned}
\sum_{v \in \mathcal{T}} \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} p_{v,k} w_b &\leq \sum_{v \in \mathcal{T}} \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} p_{v,k} \hat{w}_b && \forall \mathbf{p}_v \in \mathcal{P}_v, \\
\sum_{v \in \mathcal{T}} \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} \hat{p}_{v,k} w_b &< \sum_{v \in \mathcal{T}} \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} \hat{p}_{v,k} \hat{w}_b && \text{for some } \hat{\mathbf{p}}_v \in \mathcal{P}_v.
\end{aligned}$$

Let z^{RO} denote the optimal expected heart dose for the robust solution in the **(RO)** model. To find a **PRO** solution, a second-stage model must be solved that searches across all robust optimal solutions of **(RO)**. This model will find plans that have the same optimal heart dose of z^{RO} at worst-case, but a lower **CTV** dose in some scenarios. The **(PRO)** model is thus similar to the **(RO)** model but enforces the original **(RO)** objective as a

hard constraint while minimizing the [CTV](#) dose as the new objective,

$$\begin{aligned}
(\mathbf{PRO}) : \quad & \underset{w_b}{\text{minimize}} \quad \sum_{v \in \mathcal{T}} \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} \hat{p}_{v,k} w_b \\
& \text{subject to} \quad \frac{1}{|\mathcal{H}|} \sum_{v \in \mathcal{H}} \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} p_{v,k} w_b \leq z^{\text{RO}}, \\
& \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} \tilde{p}_{v,k} w_b \geq L_v \quad \forall \tilde{\mathbf{p}}_v \in \mathcal{P}_v, v \in \mathcal{T}, \\
& w_b \geq 0, \quad \forall b \in \mathcal{B},
\end{aligned} \tag{3.3}$$

where $\hat{p}$ is a non-worst-case breathing pattern within the relative interior of the uncertainty set. See [Rahimi, Fahimeh \(2022\)](#) for further details.

These **(PRO)** solutions, however, may over-protect the worst-case while not allowing any room for improving non-worst-case scenarios. For that reason, [Rahimi, Fahimeh \(2022\)](#) show that some of the optimality of the solutions under the worst case may be traded off, in order to gain a better quality of solutions in the average case. If ϵ is a loss parameter capturing the amount of dose that may be added to the objective of the worst-case model (i.e., the amount of average heart dose). Then, the light PRO model with an ϵ loss level (**LPRO** ϵ) can be written as:

$$\begin{aligned}
(\mathbf{LPRO}\epsilon) : \quad & \underset{w_b}{\text{minimize}} \quad \sum_{v \in \mathcal{T}} \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} \hat{p}_{v,k} w_b \\
& \text{subject to} \quad \frac{1}{|\mathcal{H}|} \sum_{v \in \mathcal{H}} \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} p_{v,k} w_b \leq Z^{\text{RO}} + \epsilon \\
& \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} \tilde{p}_{v,k} w_b \geq L_v \quad \forall \tilde{\mathbf{p}}_v \in \mathcal{P}_v, v \in \mathcal{T} \\
& w_b \geq 0, \quad \forall b \in \mathcal{B}
\end{aligned} \tag{3.4}$$

The **(LPRO** ϵ) model may be substituted for **(PRO)**, relaxing the optimal heart dose under worst-case breathing by ϵ , and allowing the optimization model to improve the outcomes

for non-worst-case scenarios. This has the clinical interpretation of allowing an increase in the total dose delivered to the heart voxels during the worst-case breathing realization, in order to decrease the total dose delivered to the tumour in non-worst-case scenarios. By adjusting ϵ , different levels of conservatism may be considered and the trade-off between optimality at worst-case and average non-worst-case plan quality can be observed.

3.3 Extended CVaR Models

In [RTTP](#), treatment plan quality is frequently assessed using a [dose volume histogram \(DVH\)](#). This metric is typically depicted as a cumulative plot that shows the percentage of the prescribed dose delivered to a patient, versus the percentage of the volume of patient organ receiving at least that dosage. An ideal DVH is depicted in [Figure 3.1](#).

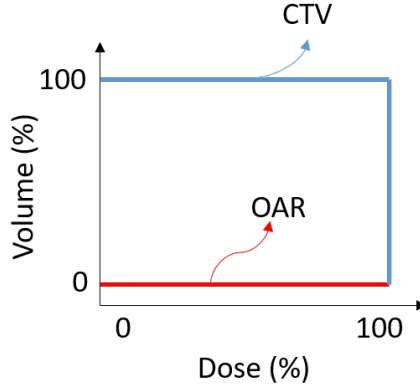


Figure 3.1: Ideal DVH diagram. Exactly 100% of the target volume receives 100% of the prescribed dose, and 0% of the OAR receives any dose. Image from [Ripsman \(2018\)](#).

A clinician typically evaluates a plan based on specific points within the [DVH](#), i.e., no more than $100 \times (1 - \beta)\%$ of a structure may receive more than some quantity of dose. In order to directly enforce this [DVH](#) behaviour, [value-at-risk \(VaR\)](#) can be used as a loss function that captures these dose-requirements (see [Romeijn et al. \(2006\)](#) and [Chan et al. \(2014b\)](#) for more details). While the resulting [VaR](#) loss function is non-linear, a bound on

the loss function that controls the average of all losses exceeding the β -VaR level (i.e., the tail end of the distribution) is called **CVaR**, and may be expressed linearly.

Past **CVaR** models have been shown to work well for radiation therapy (Romeijn et al., 2006) and extend to **RO** for radiation (Chan et al., 2014b). This section extends that work to the **PRO** and **LPRO** frameworks. As in the basic tumour coverage approach, the **CVaR** formulations use minimization of average heart dose under the expected uncertainty scenario as the primary **RO** objective. Unlike the basic models, the robust **CVaR** tumour dose is controlled using constraints on the upper and lower tails of the dose distribution, rather than simple coverage constraints.

The $(\mathbf{RO}^C)$ model allowing for robust upper and lower dose-volume criteria on the tumour looks as follows,

$$\begin{aligned}
(\mathbf{RO}^C) : \quad & \underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{|\mathcal{H}|} \sum_{v \in \mathcal{H}} \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} p_{v,k} w_b \\
& \text{subject to} \quad \bar{d}_v \geq \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} \tilde{p}_{v,k} w_b - \bar{\zeta} \quad \forall v \in \mathcal{T}, \tilde{\mathbf{p}}_v \in \mathcal{P}_v, \\
& \quad \underline{d}_v \geq \underline{\zeta} - \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} \tilde{p}_{v,k} w_b \quad \forall v \in \mathcal{T}, \tilde{\mathbf{p}}_v \in \mathcal{P}_v, \\
& \quad \bar{\zeta} + \frac{1}{(1-\beta)|\mathcal{T}|} \sum_{v \in \mathcal{T}} \bar{d}_v \leq U, \\
& \quad \underline{\zeta} - \frac{1}{(1-\beta)|\mathcal{T}|} \sum_{v \in \mathcal{T}} \underline{d}_v \geq L, \\
& \quad w_b \geq 0 \quad \forall b \in \mathcal{B}.
\end{aligned} \tag{3.5}$$

The $(\mathbf{RO}^C)$ model allows for control of how the target dose deviates from the desired value, as an uncertainty distribution within the constraints. Here, β is the tail of the tumour dose distribution, $\bar{\zeta}$ and $\underline{\zeta}$ enforce the upper and lower β -CVaR values respectively, $\bar{d}_v$ and $\underline{d}_v$ track the dose to the voxels that exceed or fall below the $\bar{\zeta}$ and $\underline{\zeta}$ values, respectively, and U and L are the upper and lower dose thresholds.

As in the basic problem, the **RC** of the model is taken as follows:

$$\begin{aligned}
& \underset{\mathbf{w}, \underline{d}, \bar{\zeta}, \bar{\mathbf{r}}, \underline{\mathbf{q}}, \underline{\mathbf{q}}}{\text{minimize}} && \frac{1}{|\mathcal{H}|} \sum_{v \in \mathcal{H}} \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} p_{v,k} w_b && (3.6) \\
& \text{subject to} && \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} [p_{v,k} - \underline{p}_{v,k}] D_{v,k,b} w_b + \sum_{k \in \mathcal{K}} \underline{p}_{v,k} \bar{q}_v + \sum_{k \in \mathcal{K}} \bar{r}_{v,k} \leq \bar{\zeta} + \bar{d}_v \quad \forall v \in \mathcal{T}, \\
& && \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} [p_{v,k} - \underline{p}_{v,k}] D_{v,k,b} w_b + \sum_{k \in \mathcal{K}} \underline{p}_{v,k} \underline{q}_v - \sum_{k \in \mathcal{K}} \underline{r}_{v,k} \geq \underline{\zeta} - \underline{d}_v \quad \forall v \in \mathcal{T}, \\
& && (\bar{p}_{v,k} + \underline{p}_{v,k}) \bar{q}_v + \bar{r}_{v,k} \geq \sum_{b \in \mathcal{B}} D_{v,k,b} (\bar{p}_{v,k} + \underline{p}_{v,k}) w_b \quad \forall v \in \mathcal{T}, k \in \mathcal{K}, \\
& && (\bar{p}_{v,k} + \underline{p}_{v,k}) \underline{q}_v - \underline{r}_{v,k} \leq \sum_{b \in \mathcal{B}} D_{v,k,b} (\bar{p}_{v,k} + \underline{p}_{v,k}) w_b \quad \forall v \in \mathcal{T}, k \in \mathcal{K}, \\
& && \bar{\zeta} + \frac{1}{(1-\beta)|\mathcal{T}|} \sum_{v \in \mathcal{T}} \bar{d}_v \leq U, \\
& && \underline{\zeta} - \frac{1}{(1-\beta)|\mathcal{T}|} \sum_{v \in \mathcal{T}} \underline{d}_v \geq L, \\
& && w_b \geq 0 \quad \forall b \in \mathcal{B}, \\
& && \bar{q}_v, \underline{q}_v \text{ free} \quad \forall v \in \mathcal{T}, \\
& && \bar{r}_{v,k}, \underline{r}_{v,k} \geq 0 \quad \forall v \in \mathcal{T}, k \in \mathcal{K}.
\end{aligned}$$

Then, considering z^{RO} as the optimal objective value of model **(RO^C)** and worst-case loss values of ϵ , a **PRO** ($\epsilon = 0$) or **LPRO** ($\epsilon > 0$) solution can be found by moving the original objective function to the constraints, bounded by $z^{\text{RO}} + \epsilon$:

$$\frac{1}{|\mathcal{H}|} \sum_{v \in \mathcal{H}} \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} p_{v,k} w_b \leq z^{\text{RO}} + \epsilon$$

and providing the new objective function, given $\hat{\mathbf{p}}$ is in the relative interior of $\mathcal{P}$,

$$\min \quad \bar{\zeta} + \frac{1}{(1-\beta)|\mathcal{T}|} \sum_{v \in \mathcal{T}} \max \left\{ 0, \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} \hat{p}(m) w_b - \bar{\zeta} \right\} -$$

$$\hat{\underline{\zeta}} - \frac{1}{(1-\beta)|\mathcal{T}|} \sum_{v \in \mathcal{T}} \max \left\{ 0, \hat{\underline{\zeta}} - \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} \hat{p}(m) w_b \right\}.$$

The maximization can once again be handled in the constraints, for the complete model, using new variables $\bar{\hat{d}}_v$ and $\underline{\hat{d}}_v$:

$$\begin{aligned}
(\mathbf{LPRO}\epsilon^C) : \quad & \underset{\mathbf{w}, \mathbf{d}, \bar{\mathbf{d}}, \bar{\zeta}, \underline{\zeta}, \bar{\mathbf{r}}, \underline{\mathbf{r}}, \bar{\mathbf{q}}, \underline{\mathbf{q}}}{\text{minimize}} \quad \left[\bar{\zeta} + \frac{1}{(1-\beta)|\mathcal{T}|} \sum_{v \in \mathcal{T}} \bar{\hat{d}}_v \right] - \left[\underline{\zeta} - \frac{1}{(1-\beta)|\mathcal{T}|} \sum_{v \in \mathcal{T}} \underline{\hat{d}}_v \right] \\
& \text{subject to} \quad \bar{\hat{d}}_v \geq \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} \hat{p}_{v,k} w_b - \bar{\zeta} \quad \forall v \in \mathcal{T}, \\
& \quad \underline{\hat{d}}_v \geq \underline{\zeta} - \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} \hat{p}_{v,k} w_b \quad \forall v \in \mathcal{T}, \\
& \quad \bar{\zeta} + \frac{1}{(1-\beta)|\mathcal{T}|} \sum_{v \in \mathcal{T}} \bar{\hat{d}}_v \leq U, \\
& \quad \underline{\zeta} - \frac{1}{(1-\beta)|\mathcal{T}|} \sum_{v \in \mathcal{T}} \underline{\hat{d}}_v \geq L, \\
& \quad \frac{1}{|\mathcal{H}|} \sum_{v \in \mathcal{H}} \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} p_{v,k} w_b \leq z^{\text{RO}} + \epsilon, \\
& \quad \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} [p_{v,k} - \underline{p}_{v,k}] D_{v,k,b} w_b + \sum_{k \in \mathcal{K}} \underline{p}_{v,k} \bar{q}_v + \sum_{k \in \mathcal{K}} \bar{r}_{v,k} \leq \bar{\zeta} + \bar{\hat{d}}_v \quad \forall v \in \mathcal{T}, \\
& \quad \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} [p_{v,k} - \underline{p}_{v,k}] D_{v,k,b} w_b + \sum_{k \in \mathcal{K}} \underline{p}_{v,k} \underline{q}_v - \sum_{k \in \mathcal{K}} \underline{r}_{v,k} \geq \underline{\zeta} - \underline{\hat{d}}_v \quad \forall v \in \mathcal{T}, \\
& \quad (\bar{p}_{v,k} + \underline{p}_{v,k}) \bar{q}_v + \bar{r}_{v,k} \geq \sum_{b \in \mathcal{B}} D_{v,k,b} (\bar{p}_{v,k} + \underline{p}_{v,k}) w_b \quad \forall v \in \mathcal{T}, k \in \mathcal{K}, \\
& \quad (\bar{p}_{v,k} + \underline{p}_{v,k}) \underline{q}_v - \underline{r}_{v,k} \leq \sum_{b \in \mathcal{B}} D_{v,k,b} (\bar{p}_{v,k} + \underline{p}_{v,k}) w_b \quad \forall v \in \mathcal{T}, k \in \mathcal{K}, \\
& \quad \bar{\zeta} + \frac{1}{(1-\beta)|\mathcal{T}|} \sum_{v \in \mathcal{T}} \bar{\hat{d}}_v \leq U, \\
& \quad \underline{\zeta} - \frac{1}{(1-\beta)|\mathcal{T}|} \sum_{v \in \mathcal{T}} \underline{\hat{d}}_v \geq L, \\
& \quad w_b \geq 0 \quad \forall b \in \mathcal{B}, \\
& \quad \bar{q}_v, \underline{q}_v \text{ free} \quad \forall v \in \mathcal{T}, \\
& \quad \bar{r}_{v,k}, \underline{r}_{v,k} \geq 0 \quad \forall v \in \mathcal{T}, k \in \mathcal{K}.
\end{aligned} \tag{3.7}$$

The gain of the model is a concept introduced in [Rahimi, Fahimeh \(2022\)](#) for **LPRO**, to describe the non-worst-case gain in the secondary objective relative to the loss in the primary model ϵ . It is extended to these **CVaR** models based on the total tightness of the overdose constraints,

$$\bar{\zeta} + \frac{1}{(1-\beta)|\mathcal{T}|} \sum_{v \in \mathcal{T}} \max \left\{ 0, \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} p_{v,k} w_b - \bar{\zeta} \right\},$$

and the underdose constraints,

$$\underline{\zeta} - \frac{1}{(1-\beta)|\mathcal{T}|} \sum_{v \in \mathcal{T}} \max \left\{ 0, \underline{\zeta} - \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} p_{v,k} w_b \right\}.$$

Leading to the following gain function,

$$f(\epsilon; \hat{\mathbf{p}}_v) = \left[\bar{\zeta} + \frac{1}{(1-\beta)|\mathcal{T}|} \sum_{v \in \mathcal{T}} \bar{d}_v - \underline{\zeta} + \frac{1}{(1-\beta)|\mathcal{T}|} \sum_{v \in \mathcal{T}} \underline{d}_v \right] - \left[\bar{\hat{\zeta}} + \frac{1}{(1-\beta)|\mathcal{T}|} \sum_{v \in \mathcal{T}} \bar{\hat{d}}_v - \hat{\underline{\zeta}} + \frac{1}{(1-\beta)|\mathcal{T}|} \sum_{v \in \mathcal{T}} \hat{\underline{d}}_v \right].$$

This can be simplified to the decrease in the second term (total overdose and underdose for a given non-worst-case $\hat{\mathbf{p}}_v$), seeing as the first term of the gain function is constant, resulting in the equation,

$$f(\epsilon; \hat{\mathbf{p}}_v)^{\text{LR}} = \left[\bar{\hat{\zeta}} + \frac{1}{(1-\beta)|\mathcal{T}|} \sum_{v \in \mathcal{T}} \bar{\hat{d}}_v - \hat{\underline{\zeta}} + \frac{1}{(1-\beta)|\mathcal{T}|} \sum_{v \in \mathcal{T}} \hat{\underline{d}}_v \right].$$

This characterizes the improvement in the tightness of constraints in (**LPRO** ϵ^C), as a function of the change in ϵ departure from the original **RO** objective function value.

3.4 Methods and Materials

This section outlines the clinical data transformations and methods used for implementing the models proposed in the previous sections. Overview of patient datasets and sources of uncertainty are detailed in Section 3.4.1. The breathing motion simulation method used to generate random motion scenarios for testing the proposed models is discussed in Section 3.4.2, followed by an algorithm designed to generate random breathing patterns in Section 3.4.3.

3.4.1 Clinical Data and Processing

The data consists of a five-patient cohort prescribed whole breast IMRT. A tangential IMRT treatment setup with two opposing beams was selected to treat each of the patients, in accordance with clinical protocols (Purdie et al., 2011, 2014). Within a treatment planning system (Pinnacle, Philips Healthcare, Fitchburg, WI), the clinical target volume (CTV) in the breast was defined as the volume irradiated by the tangential beams at the 55% isodose level, with the left lung and heart excluded, and then contracted 0.5 cm in the axial plane and 1.0 cm in the superior-inferior direction. All patients were given a prescribed dose of 42.4 Gy to the whole breast over a treatment with 16 fractions.

In order to simulate each patient’s breathing motion, previously established positioning and immobilization techniques were used within a clinical CT scanner (Brilliance, Philips Medical Systems, Milpitas, CA). This resulted in a 4D-CT image with ten phases, capturing a complete breathing cycle for each patient. The grid of voxels that made up each of the patients’ organs was tracked over the ten phases of breathing motions using a hybrid deformable registration modeling method (Weistrand and Svensson, 2015). This was done using the 4D-CT datasets in RayStation (version 3.99.0.8, RaySearch Laboratories AB, Stockholm, Sweden).

3.4.2 Breathing Pattern Simulation

In order to evaluate the plans, the dose accumulation of both the target and the heart was considered under both expected (nominal) and random scenarios. Under the expected or nominal scenario, it was assumed that 50% of the dose is accumulated during the exhale phase, when the heart is closest to the breast, while the rest of the breathing time is divided equally among the remaining phases. See the red points in Figure 3.2 for a five-phase example. Note that these phase allocations are in line with past RO literature (Mahmoudzadeh et al., 2015), as well as previous clinical studies (Rouabhi et al., 2019; Balter et al., 1998). Using the phase allocations, alongside the dose-influence matrix at a given phase, the accumulated dose to each voxel may be calculated for the nominal model.

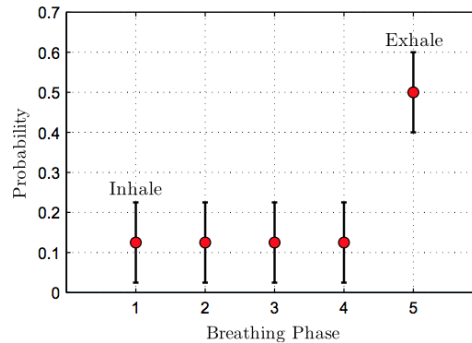


Figure 3.2: Five-phase breathing cycle. The red points are expected (nominal) values, the bars represent the uncertainty sets. Image from Chan et al. (2014b) with permission.

To evaluate the plan in random non-nominal scenarios, a set of simulated reasonable breathing patterns must be generated. To do so, the bounds on the uncertainty set were set to $\pm 10\%$ of the nominal pattern and each breathing pattern was required to sum to 1, or 100% of a breathing cycle. This uncertainty region is represented by the bars in Figure 3.2. Clinically, it is seen as a reasonable uncertainty set, allowing for reasonable departures from the expected breathing pattern, without creating impossible scenarios, e.g., a breathing cycle only accounting for 80% of the time.

3.4.3 Proposed Breathing Pattern Generation Algorithm

An algorithm for generating feasible polyhedral breathing sets (or any polyhedral uncertainty realization) was designed for the purpose of running experiments with the RO models. Algorithm 6 generates feasible realizations of any single-constraint polyhedral uncertainty set $\mathcal{U}$ and is guaranteed to terminate in a finite number of iterations. The MATLAB (2022) implementation of this pseudocode is available for download at: <https://www.mathworks.com/matlabcentral/fileexchange/119993-genuncertaintyrealizations>.

Algorithm 6 Randomly Generate n Feasible Breathing Patterns

Require: $n, \underline{\mathbf{p}}, \bar{\mathbf{p}}$

```

1:  $nPhases \leftarrow \text{length}(\underline{\mathbf{p}})$ 
2:  $totalSlack \leftarrow 1 - \text{sum}(\underline{\mathbf{p}})$  ▷ Required total amount to add to get to 1
3:  $\delta \leftarrow \bar{\mathbf{p}} - \underline{\mathbf{p}}$  ▷ Max that can be added to each phase
4: Br  $\leftarrow \text{rep}(n, \delta)$  ▷ Generate  $n$  copies (rows) of delta
5: for  $i \in 1 \dots n, j \in 1 \dots nPhases$  do
6:   Br( $i, j$ )  $\leftarrow \mathbf{Br}(i, j) \times \text{rand.unif}(0,1)$  ▷ Allocate realization randomly
7: for  $i \in 1 \dots n$  do
8:   Br( $i, :$ )  $\leftarrow \mathbf{Br}(i, :) \times totalSlack / \text{sum}(\mathbf{Br}(i, :))$  ▷ Adjust sum to 1
9: for each row  $i$  where an element Br( $i, j$ ) exceeds  $\delta(j)$  do
10:  while  $\delta - \mathbf{Br}(i, :)$  has values  $\leq 0$  do
11:    let j be the set of  $\leq 0$  indices
12:    Br( $i, \mathbf{j}$ )  $\leftarrow \delta(\mathbf{j})$ 
13:    let j' be the remaining indices
14:     $totalSlack' \leftarrow totalSlack - \text{sum}(\delta(\mathbf{j}))$ 
15:    Br( $i, \mathbf{j}'$ )  $\leftarrow \mathbf{Br}(i, \mathbf{j}') \times totalSlack' / \text{sum}(\mathbf{Br}(i, \mathbf{j}'))$  ▷ Re-adjust to 1

```

Output: **Br**

Using this algorithm, 50 random breathing phases were generated within the selected uncertainty set, for testing the models.

3.5 Results

The optimization models were tested on a database of five patient datasets. These datasets were each down-sampled to include every 32nd target voxel, from a $1 \times 1 \times 2 \text{ mm}^3$ resolution to an effective $4 \times 4 \times 4 \text{ mm}^3$ resolution, in order to make the data granularity more comparable to the clinical systems (Mahmoudzadeh et al., 2015). The heart voxels were not down-sampled, to maintain maximal sensitivity to overdose, however, all those that received zero or negligible dose ($\leq 0.000001 \text{ Gy/MU}$) from the beam were pruned from the decision space during optimization for computational efficiency, and subsequently re-added for the diagrams and dose-volume calculations. Note that patients 1, 2, 3, 4, and 5 correspond to patients B, C, D, A, and E, in Ripsman et al. (2022).

The uncertainty set was constructed from the first $|\mathcal{K}| = 5$ breathing phases considered for each patient, mapped from inhale to exhale, capturing the full range of breathing motion. The phases were assigned the uncertainty bounds $\underline{\mathbf{p}} = [0.025, 0.025, 0.025, 0.025, 0.4]$ and $\bar{\mathbf{p}} = [0.225, 0.225, 0.225, 0.225, 0.6]$, while the nominal PMF of $\mathbf{p} = [0.125, 0.125, 0.125, 0.125, 0.5]$, as discussed in Section 3.4.2. The nominal p , which is in the relative interior of the uncertainty set, was used in the objective function for all models.

Since all introduced models are LPs, they can be solved efficiently using commercial off-the-shelf solvers. In this study, all models were implemented with the RC approach, using C++ with IBM ILOG CPLEX (Cplex, 2017) version 12.7. The CPLEX-C++ interface was used to formulate the objective function and constraints of the problem and find the optimal solution. The implemented C++ code is available for download in the Supporting Material of Ripsman et al. (2023).

To evaluate the quality of the plans, the RO, PRO, and the LPRO ϵ model at loss levels $\epsilon = [0.1, 0.2, 0.3, 0.4, 0.5]$ were run for each of the five patients on the Digital Research

Alliance of Canada’s¹ Graham cluster. A set of metrics on both the CTV (Average, D0.5%, D99%, and D99%) and the heart (Average, D1cc, D10cc, D25cc) were calculated for each plan, accumulated for the nominal and simulated breathing scenarios. Note the dose volume criteria (Dx%) indicate the amount of dose delivered to the hottest x% of the tumor. The Dxcc is the amount of dose delivered to the hottest x cubic centimetres of the heart.

Section 3.5.1 illustrates the results for the nominal breathing pattern, while Section 3.5.2 shows the plans under 50 random simulated realizations of uncertainty. Finally, Section 3.5.3 demonstrates the CVaR results.

3.5.1 Basic Model Nominal Results

The RO, PRO, and LPRO models are first compared when plans are both optimized for the nominal breathing scenario and the dose is accumulated based on the nominal breathing cycle realization. Table 3.1 shows the resulting dose metrics for the CTV and the heart. Note that all first-stage models are basic RO models that enforce a prescribed dose to all CTV voxels across all breathing scenarios and aim to minimize the expected heart dose in the robust plan. The PRO and LPRO models then benefit from a second-stage optimization that aims to minimize overdose to the tumour, but the first model is the same across all three model types. Hence, the RO and PRO plans always have the same average heart dose metric but potentially perform differently in other metrics.

Given that the minimum prescribed CTV dose is always guaranteed to be met in all of the models, this metric is not reported in Table 3.1. For all the metrics reported, a lower dose indicates a better plan, but a trade-off does occur. An RO model run with no knowledge of any requirements beyond Average_H yields very conservative heart doses, but extreme target overdoses (on the scale of millions of gray in the extreme case of patient

¹www.alliancecan.ca

Patient	ϵ	CTV Dose Metrics				Heart Dose Metrics			
		Average _T	D0.5%	D95%	D99%	Average _H	D1cc	D10cc	D25cc
1	-	12372.9	330268.9	134.1	44.6	0.08	39.5	3.9	0.6
	0	10317.2	291064.3	113.5	44.5	0.08	39.5	3.8	0.6
	0.1	56.0	207.0	42.6	42.5	0.18	38.8	10.9	4.7
	0.2	43.2	59.2	42.4	42.4	0.28	37.7	16.9	8.7
	0.3	42.9	50.8	42.4	42.4	0.38	42.9	23.8	12.7
	0.4	42.8	47.9	42.4	42.4	0.48	42.6	28.5	17.5
	0.5	42.7	46.9	42.4	42.4	0.58	43.1	34.6	22.3
2	-	4116.2	53813.3	66.2	43.4	1.35	65.4	50.3	43.4
	0	1571.0	45548.2	52.2	43.1	1.35	65.2	50.4	43.3
	0.1	55.2	191.2	42.5	42.4	1.45	60.2	48.0	42.4
	0.2	43.9	59.6	42.4	42.4	1.55	53.4	46.1	41.8
	0.3	43.6	53.6	42.4	42.4	1.65	49.3	44.7	42.0
	0.4	43.6	52.2	42.4	42.4	1.75	46.3	44.2	42.1
	0.5	43.5	51.9	42.4	42.4	1.85	46.1	44.2	42.1
3	-	3852.8	52818.6	53.8	44.0	0.19	46.6	24.5	6.6
	0	3852.3	52814.5	54.0	44.3	0.19	46.6	24.5	6.6
	0.1	127.7	3957.9	42.8	42.6	0.29	51.3	28.9	13.3
	0.2	43.9	53.7	42.5	42.4	0.39	44.9	31.7	18.6
	0.3	43.6	48.7	42.5	42.4	0.49	44.2	36.1	24.2
	0.4	43.6	48.9	42.5	42.4	0.59	48.7	40.0	28.3
	0.5	43.6	49.0	42.5	42.4	0.69	50.6	41.8	32.4
4	-	6924245.6	322125604.1	44.7	42.8	3.17	80.7	51.3	46.9
	0*	1622597.3	43512067.1	44.6	42.8	3.17	80.8	51.3	46.9
	0.1	135.1	1575.9	42.8	42.5	3.27	60.1	47.5	45.4
	0.2	46.8	82.8	42.5	42.5	3.37	52.4	48.1	45.6
	0.3	43.9	57.3	42.5	42.4	3.47	53.8	48.5	46.1
	0.4	43.7	55.7	42.5	42.4	3.57	54.5	48.9	46.8
	0.5	43.6	52.4	42.5	42.4	3.67	52.1	47.7	46.4
5	-	7685.0	89694.7	68.2	43.1	0.62	60.4	43.2	33.2
	0	2578.5	96723.6	47.2	42.7	0.62	58.3	43.7	33.0
	0.1	43.6	57.9	42.4	42.4	0.72	63.0	43.0	34.7
	0.2	43.2	49.2	42.4	42.4	0.82	47.1	42.8	37.2
	0.3	43.2	49.0	42.4	42.4	0.92	47.9	43.2	39.8
	0.4	43.2	48.7	42.4	42.4	1.02	47.2	43.0	40.9
	0.5	43.2	48.6	42.4	42.4	1.12	47.0	43.6	41.8

Table 3.1: Nominal aggregate dose values for each of the models.

* $\epsilon = 0.0001$ was needed to obtain a feasible PRO solution, rather than exactly 0

4). Running the PRO follow-up model leads to slight improvements in Average_T for all five patients, with no cost to average heart dose, but it still leaves extreme overdose to the region.

For the most part, across the remaining metrics, patient plans appear to improve slightly (or change negligibly) when plans go from RO to PRO. There are, however, a couple of cases where uncontrolled metrics regress, such as Patient 2's slight increase at the D10cc level, and a slight uptick in the D95% and D99% for Patient 3. Similarly, Patient 4 sees mostly improvement from PRO, but the D1cc gets 0.1 Gy hotter, and Patient 5's D0.5% and D10cc values move a bit unfavourably. The remaining metrics, however, show a much stronger trend in a positive direction with the addition of PRO.

When the LPRO model is used with $\epsilon = 0.1$ Gy the average dose to the heart increases by exactly 0.1 Gy, as expected. All CTV metrics uniformly improve or stay the same with each increase in ϵ , which is also to be expected. In general, there is also a downward trend in the hottest region of the heart, captured by the D1cc metric, as epsilon is introduced, however, there are outliers to this trend, such as Patient 5, which increases significantly with the shift to $\epsilon = 0.1$. There is an increasing trend in the D10cc and D25cc metrics for Patients 1,3 and 5, whereas Patients 2 and 4 remain fairly consistent in this regard.

For the three patients with increasing D10cc metrics, this trade-off could assist clinicians in making planning decisions. Take Patient 1, for example, if the clinician wants a D10cc under 20 Gy, they may settle for an $\epsilon = 0.2$, even though this results in an average dose to the target of 43.2 Gy, instead of the lower 42.9 Gy afforded by $\epsilon = 0.3$. Clinicians may also look at the diminishing returns on the CTV improvements as indications of stopping criteria. For most patients, the Average_T does not improve much beyond $\epsilon = 0.2$, with Patient 4 being the only real exception to this rule. The D95% and D99% metrics are even less sensitive, whereas considering a more sensitive metric like D0.5% would lead to higher- ϵ stopping criteria, which is lowest for $\epsilon = 0.5$ for all patients except Patient 3, whose lowest D0.5% is at $\epsilon = 0.3$.

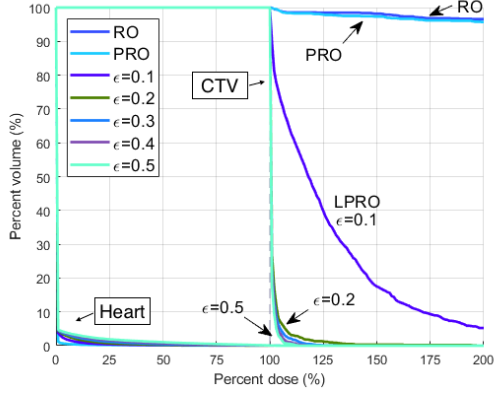
To gain greater insight into how the dose changes with each plan, the DVH of the nominal realizations of each the plan is depicted in Figure 3.3. It can be seen that the RO and PRO for all patients are largely overdosing the CTV due to their neglect of the secondary objective. The LPRO curves, on the contrary, are closer to reasonable doses without any major visible loss on the DVH plots for the heart. Moreover, because no hard constraints or tuneable parameters (aside from ϵ) were added to make this improvement, no *a-priori* knowledge about the model or manual tuning was needed to obtain these benefits.

It is evident from the DVHs that the impact of ϵ varies quite a lot from plan to plan, whereas adding some value of $\epsilon > 0$ is valuable in preventing significant overdose for all patients. We also note that the magnitude of the baseline heart dose does not appear to be particularly correlated with the impact of changing ϵ , meaning it is likely tied to more geometric planning features. The overall impactfulness of each change can, however, be inferred from the DVH plot. For Patient 1, the jump to a more conformal plan clearly occurs at $\epsilon = 0.2$, and the same could be argued for Patients 2 and 3. Patient 4, however, really appears to benefit from the jump to $\epsilon = 0.3$ and Patient 5 is already fairly conformal by $\epsilon = 0.1$.

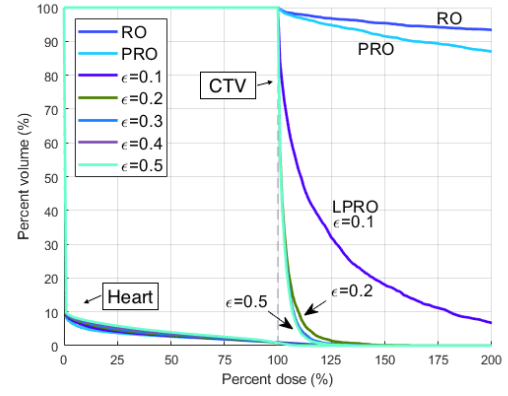
3.5.2 Basic Model Random Breathing Simulations

Fifty random breathing patterns, all contained within our uncertainty set $\mathcal{U}$, were generated using the algorithm detailed in Section 3.4.3. Table 3.2 uses the same metrics as Table 3.1 but each entry contains the realized upper and lower bounds for a given dose metric across the 50 simulated scenarios. Note that each of the RO, PRO, and LPRO plans were optimized for the expected (nominal) heart and target doses (PRO and LPRO), whereas the dose in the table is calculated for the 50 non-nominal breathing scenarios. The results demonstrate that optimizing for a given non-worst-case scenario (i.e., the nominal, in this case) will improve the plan quality across other simulated scenarios as well.

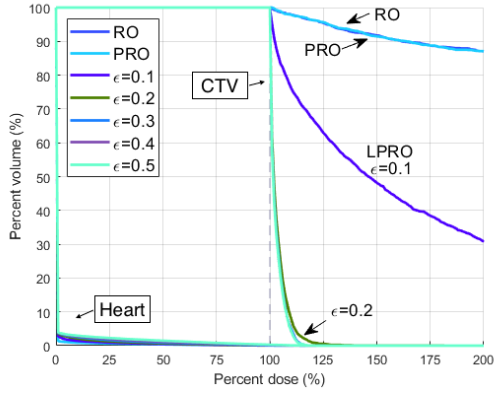
The ranges also provide information about plan performance under various realizations



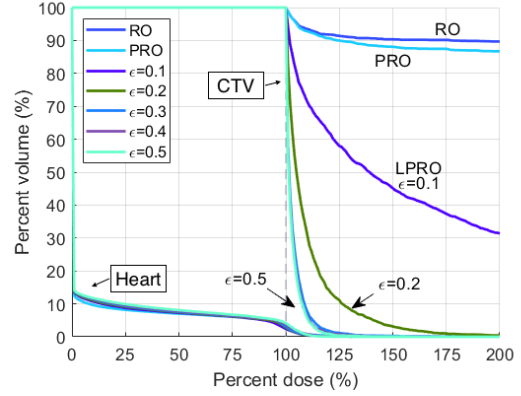
(a) Patient 1



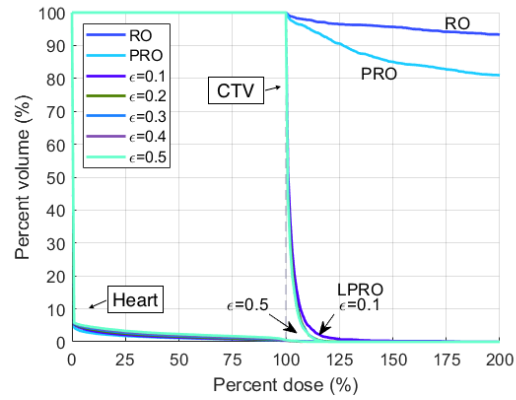
(b) Patient 2



(c) Patient 3



(d) Patient 4



(e) Patient 5

Figure 3.3: The DVHs of the CTV and heart for each of every patient's seven plans, assuming nominal breathing realizations.

of the uncertainty set. It can be observed that the plans typically vary around their nominal realizations, without straying too far from the observations seen in the nominal cases. In all plans, the dose metric differences vary by between 0% to just under 15% from the nominal realization. The largest ranges are found at the more extreme metrics (D0.5%, D1cc), which are not directly controlled by the optimization models. Additionally, ranges tended to grow considerably tighter as the Pareto trade-off is given more weight, and the plans become more conformal. In general, the relative stability of the plans under uncertainty is promising for clinical use.

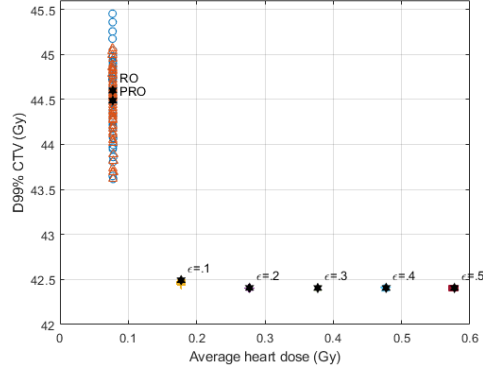
To further illustrate the relationship between the different metrics, Figure 3.4 depicts the relationship between the average heart dose and D99% for the various realization of each patient’s generated plans. The variation shown in the graph is a result of the uncertain breathing realizations, which can be seen to cluster around the nominal solution, depicted as a black star in the images. These images correspond to columns 6 and 7 from the summary in Table 3.2 and graphically depict some of the trade-off decisions that must be made; i.e., with increasing average heart dose, we see a decrease in overdose at the D99% level.

Figure 3.4 also graphically demonstrates the diminishing returns with respect to increasing heart dose, as noted above for singular nominal realizations. Note, for example that for most of the patients, beyond $\epsilon = 0.1$, there is no significant improvement at the D99% level, so if that is a major decision criteria, no more than 0.1 Gy should be sacrificed. The primary outlier is Patient 4, who to some degree seems to benefit until roughly $\epsilon = .3$. Other dose criteria can be compared and evaluated in a similar manner, in order to obtain a full picture of the plan trade-offs.

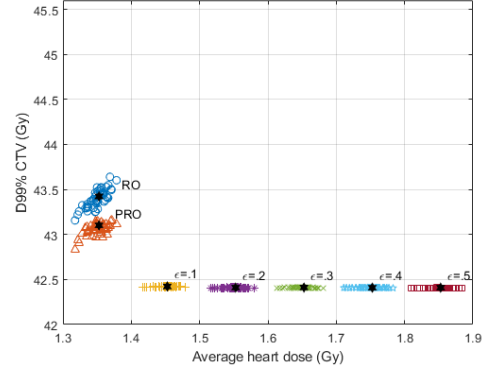
Patient	ϵ	CTV Dose Metrics				Heart Dose Metrics			
		Average r	D0.5%	D95%	D99%	Average H	D1cc	D10cc	D25cc
1	-	[11872.0, 12983.9]	[309539.9, 349315.7]	[129.5, 138.2]	[43.6, 45.5]	[.08, .08]	[38.8, 40.9]	[3.7, 4.0]	[6, 6]
	0	[9876.6, 10854.9]	[272775.6, 307882.4]	[108.9, 115.0]	[43.6, 45.1]	[.08, .08]	[38.8, 41.3]	[3.7, 3.9]	[6, 7]
	0.1	[55.8, 56.2]	[182.8, 218.2]	[42.5, 42.6]	[42.4, 42.5]	[.18, .18]	[38.3, 39.4]	[10.6, 11.2]	[4.6, 4.8]
	0.2	[43.2, 43.2]	[58.3, 59.9]	[42.4, 42.4]	[42.4, 42.4]	[.28, .28]	[37.6, 38.4]	[16.7, 17.3]	[8.6, 8.8]
	0.3	[42.9, 42.9]	[50.7, 50.9]	[42.4, 42.4]	[42.4, 42.4]	[.38, .38]	[42.8, 43.2]	[23.7, 24.1]	[12.6, 12.8]
	0.4	[42.8, 42.8]	[47.9, 48.0]	[42.4, 42.4]	[42.4, 42.4]	[.47, .48]	[42.6, 43.2]	[28.3, 28.7]	[17.3, 17.7]
2	0.5	[42.7, 42.7]	[46.8, 47.0]	[42.4, 42.4]	[42.4, 42.4]	[.57, .58]	[43.1, 43.2]	[34.4, 34.9]	[22.1, 22.7]
	-	[4106.3, 4128.2]	[52902.5, 55336.6]	[65.3, 66.8]	[43.2, 43.6]	[1.32, 1.38]	[62.9, 71.1]	[50.1, 50.6]	[43.0, 43.5]
	0	[1563.9, 1582.2]	[44099.7, 46481.5]	[51.4, 52.8]	[42.8, 43.2]	[1.32, 1.38]	[62.4, 66.3]	[50.0, 50.6]	[43.1, 43.5]
	0.1	[55.2, 55.3]	[189.6, 193.6]	[42.5, 42.5]	[42.4, 42.4]	[1.42, 1.48]	[57.0, 63.8]	[47.9, 48.2]	[42.2, 42.6]
	0.2	[43.9, 44.0]	[59.3, 59.8]	[42.4, 42.4]	[42.4, 42.4]	[1.52, 1.58]	[52.6, 55.0]	[46.0, 46.3]	[41.7, 42.0]
	0.3	[43.6, 43.6]	[53.5, 53.7]	[42.4, 42.4]	[42.4, 42.4]	[1.61, 1.68]	[48.3, 49.8]	[44.6, 44.9]	[42.0, 42.2]
3	0.4	[43.5, 43.6]	[52.0, 52.5]	[42.4, 42.4]	[42.4, 42.4]	[1.71, 1.78]	[46.2, 46.5]	[44.2, 44.4]	[41.9, 42.2]
	0.5	[43.5, 43.6]	[51.7, 51.9]	[42.4, 42.4]	[42.4, 42.4]	[1.81, 1.88]	[46.0, 46.4]	[44.1, 44.3]	[42.0, 42.1]
	-	[3811.6, 3873.9]	[51638.3, 53504.0]	[53.4, 55.0]	[43.4, 44.2]	[1.17, .22]	[42.7, 51.0]	[21.8, 28.0]	[6.0, 7.6]
	0	[3811.1, 3873.4]	[51639.1, 53501.0]	[53.8, 54.8]	[43.6, 44.4]	[1.17, .22]	[42.7, 51.0]	[21.8, 28.0]	[6.0, 7.6]
	0.1	[125.0, 130.3]	[3608.4, 4236.8]	[42.7, 42.9]	[42.5, 42.6]	[.26, .33]	[46.6, 60.2]	[25.8, 32.1]	[12.1, 14.7]
	0.2	[43.9, 43.9]	[53.2, 53.8]	[42.4, 42.5]	[42.4, 42.4]	[.36, .43]	[42.4, 48.5]	[28.9, 34.8]	[17.1, 20.7]
4	0.3	[43.6, 43.6]	[48.6, 48.9]	[42.4, 42.5]	[42.4, 42.4]	[.45, .54]	[41.7, 47.3]	[33.3, 39.0]	[22.3, 26.6]
	0.4	[43.6, 43.6]	[48.7, 49.2]	[42.4, 42.5]	[42.4, 42.4]	[.55, .65]	[46.0, 51.0]	[37.7, 42.3]	[26.3, 30.9]
	0.5	[43.6, 43.6]	[48.8, 49.3]	[42.4, 42.5]	[42.4, 42.4]	[.65, .75]	[49.1, 51.9]	[40.2, 43.7]	[30.3, 34.7]
	-	[6813154.3, 7020979.8]	[304199002.4, 329536800.4]	[44.4, 44.9]	[42.6, 42.8]	[3.11, 3.27]	[72.2, 94.3]	[51.0, 52.5]	[46.7, 47.5]
	0*	[1613398.6, 1632438.4]	[41937859.6, 46928982.8]	[44.2, 44.7]	[42.6, 42.8]	[3.11, 3.27]	[72.2, 94.2]	[51.0, 52.4]	[46.7, 47.5]
	0.1	[134.4, 136.4]	[1552.8, 1632.6]	[42.6, 42.8]	[42.4, 42.5]	[3.21, 3.36]	[55.6, 66.2]	[47.4, 48.0]	[45.3, 45.8]
5	0.2	[46.8, 46.9]	[81.7, 86.0]	[42.5, 42.5]	[42.4, 42.5]	[3.31, 3.47]	[52.0, 53.0]	[47.9, 48.4]	[45.5, 45.9]
	0.3	[43.8, 43.9]	[56.2, 57.9]	[42.4, 42.5]	[42.4, 42.4]	[3.41, 3.57]	[53.4, 54.7]	[48.4, 49.2]	[46.1, 46.4]
	0.4	[43.7, 43.7]	[55.2, 56.2]	[42.4, 42.5]	[42.4, 42.4]	[3.51, 3.68]	[54.0, 55.1]	[48.7, 49.6]	[46.7, 47.0]
	0.5	[43.6, 43.6]	[52.1, 54.1]	[42.4, 42.5]	[42.4, 42.4]	[3.61, 3.78]	[51.8, 52.9]	[47.6, 48.0]	[46.3, 46.6]
	-	[7678.5, 7695.0]	[86543.2, 93472.4]	[67.2, 69.0]	[43.0, 43.2]	[.58, .65]	[59.6, 61.6]	[43.1, 43.3]	[32.3, 33.5]
	0	[2572.7, 2586.6]	[96143.1, 97296.4]	[47.1, 47.4]	[42.6, 42.8]	[.58, .65]	[57.8, 59.0]	[43.5, 43.9]	[31.8, 33.5]
6	0.1	[43.6, 43.6]	[57.6, 58.7]	[42.4, 42.4]	[42.4, 42.4]	[.68, .76]	[62.5, 63.4]	[43.0, 43.2]	[33.9, 35.3]
	0.2	[43.2, 43.2]	[49.0, 49.5]	[42.4, 42.4]	[42.4, 42.4]	[.77, .86]	[46.7, 47.5]	[42.7, 42.8]	[36.8, 37.7]
	0.3	[43.2, 43.2]	[48.5, 49.2]	[42.4, 42.4]	[42.4, 42.4]	[.87, .97]	[47.7, 48.0]	[42.9, 43.3]	[39.5, 40.0]
	0.4	[43.2, 43.2]	[48.6, 48.8]	[42.4, 42.4]	[42.4, 42.4]	[.97, 1.07]	[47.2, 47.3]	[42.9, 43.2]	[40.6, 41.1]
	0.5	[43.2, 43.2]	[48.5, 48.9]	[42.4, 42.4]	[42.4, 42.4]	[1.07, 1.17]	[46.8, 47.5]	[43.5, 43.7]	[41.7, 41.9]

Table 3.2: Dose values at random breathing realizations for each of the models.

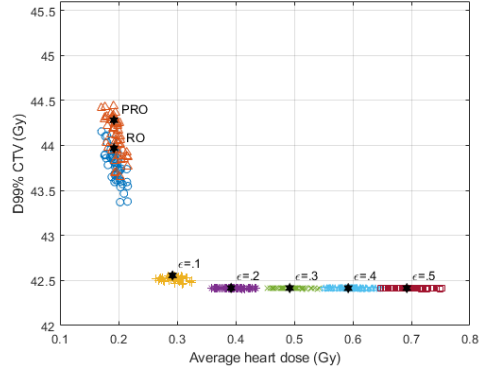
* $\epsilon = 0.0001$ was needed to obtain a feasible PRO solution, rather than exactly 0



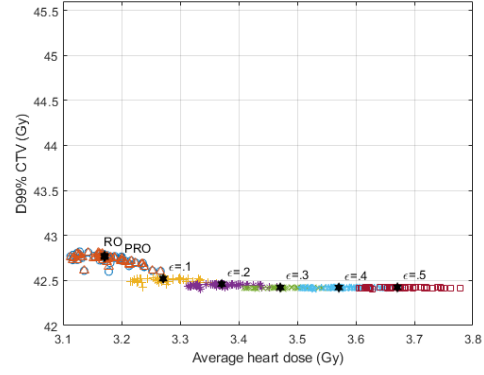
(a) Patient 1



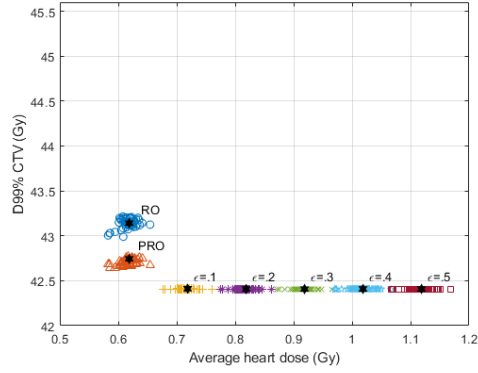
(b) Patient 2



(c) Patient 3



(d) Patient 4



(e) Patient 5

Figure 3.4: D99% of CTV vs. average heart dose for 50 simulated scenarios. The black stars are nominal realizations while each cluster is 50 simulated realizations of the 7 models.

3.5.3 Extended CVaR Model

In this section, only patient 5 is considered, but with a greater emphasis on breathing patterns and realizations. Starting with the standard nominal breathing pattern from the previous subsection, then slowly reducing conservatism, the uncertainty sets were made incrementally larger, without making them too large to be reasonable. The small, medium, and large breathing uncertainty sets are bounded as follows:

$$\mathcal{U}^S: \mathbf{p} - \underline{\mathbf{p}} = (0.025, 0.025, 0.025, 0.025, 0.4) \text{ and } \mathbf{p} + \bar{\mathbf{p}} = (0.225, 0.225, 0.225, 0.225, 0.6)$$

$$\mathcal{U}^M: \mathbf{p} - \underline{\mathbf{p}} = (0.005, 0.005, 0.005, 0.005, 0.3) \text{ and } \mathbf{p} + \bar{\mathbf{p}} = (0.3, 0.3, 0.3, 0.3, 0.7)$$

$$\mathcal{U}^L: \mathbf{p} - \underline{\mathbf{p}} = (0, 0, 0, 0, 0.2) \text{ and } \mathbf{p} + \bar{\mathbf{p}} = (0.4, 0.4, 0.4, 0.4, 0.8)$$

The robust CVaR formulations were then run for each of the three uncertainty sets. RO, PRO, and LPRO at $\epsilon = 0.1, 0.2, 0.3, 0.4$, and 0.5 Gy were each once again run using the CVaR framework. The run time for each model, along with the uncertainty-set size are given in minutes in Table 3.3. The times vary from under two hours to over four hours. The RO model is quickest as it is a single-stage implementation, while the PRO and LPRO take longer to solve due to their second-stage component. Note, all were, once again, run using the RC setup vs. CP.

$\mathcal{U}$	Model						
	RO	PRO	$\epsilon = 0.1$	$\epsilon = 0.2$	$\epsilon = 0.3$	$\epsilon = 0.4$	$\epsilon = 0.5$
Small	110.81	203.76	224.80	233.45	230.83	269.69	236.76
Medium	128.28	185.25	244.33	247.08	246.13	240.67	244.60
Large	112.86	166.61	238.92	258.88	247.22	273.16	247.74

Table 3.3: Time (mins) to run the CVaR case of each model on the full-sized patient data

The results from all seven RO models run with the medium-sized uncertainty set are plotted with a nominal realized distribution in Figures 3.5 and 3.6. Since all RO, PRO,

and LPRO models guarantee the robustness of the solutions under the worst-case scenario, the tumour dose-volume constraints are implicitly met in all solutions. Another immediate observation is that the RO and PRO models are visually indistinguishable, whereas light PRO goes a long way to reducing conservatism across all uncertainty set sizes. For the relatively small trade-offs in heart dose (Figure 3.6) fairly large reductions in tumour overdose are seen in Figure 3.5, characterized by the distribution of dose shifting to the left.

It should, however, be noted that unlike in the basic case in Section 3.5.1, a complete leftwards shift isn't necessarily desired because there are both upper and lower dose-volume constraints on the tumour. The implemented robust CVaR requirement ensures that no more than 10% of the tumour receives less than 90% of the prescribed dose, and no more than 10% receives more than 110% of the prescribed dose, for any realization of uncertainty $\tilde{\mathbf{p}}$. Clinicians might decide that a more central tumour dose, and in turn, a slightly more conservative heart dose, such as $\epsilon = 0.3$ might be ideal, even if 0.5 Gy of extra average heart dose would be acceptable in the final plan. In this way, using PRO and light PRO alongside CVaR constraints becomes a systematic exploration of the decision space, in terms of a metric that clinicians are incredibly concerned with, i.e., mean heart dose – of which there is no safe amount (Bergom et al., 2021) – versus overall tumour-dose conformity.

The models run with the smaller and larger breathing pattern, $\mathcal{U}^S$ and $\mathcal{U}^L$, even with different realized $\mathbf{p}$ values, yield nearly identical-looking plots to Figures 3.5 and 3.6, due to the scale of the problem, and the aggregate mechanism of the plot. For this reason, we will examine plots with higher resolution in Figure 3.7, as well as clinical indicators in Table 3.4 to discuss the differences in results and guarantees between these different models.

From Figure 3.7, it is evident that for the nominal realization of the breathing pattern, the difference between a smaller and larger uncertainty set is a right-shifted tumour dose, and an upwards-shifted heart dose. The value of the shift is small, due to the number of

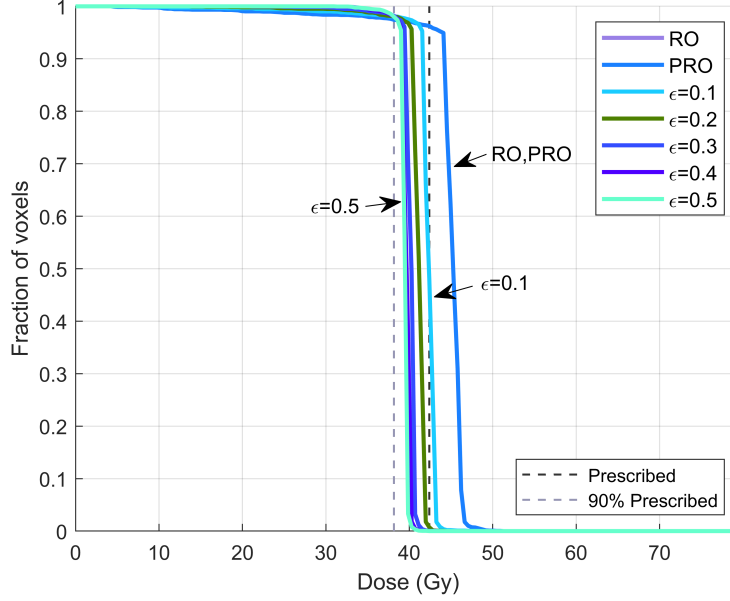


Figure 3.5: Comparison of tumour DVH plots for the RO, PRO and light PRO model results with the $\mathcal{U}^M$ uncertainty set and a nominal realized $\mathbf{p}$.

voxels present in the organs of interest.

Table 3.4 shows the outcome of running the models optimized for the large uncertainty set, when the realized $\tilde{\mathbf{p}}$ value is nominal $\mathbf{p}$, maximally exhale-weighted $\tilde{\mathbf{p}} = (0, 0, 0, 0.2, 0.8)$ and maximally inhale-weighted $\tilde{\mathbf{p}} = (0.4, 0.4, 0.2, 0, 0)$. The first reported value is the average dose to the tumor, $\text{Avg}_{\mathcal{T} \text{ Dose}}$, which is predictably similar for RO and PRO which consider heart dose to the exclusion of other factors. The RO and PRO values do differ slightly, indicating that they are not identical, however, the improvement is negligible, meaning without sacrificing any heart dose, it is difficult to overdose the tumor any less.

The RO and PRO doses are again similar across these metrics. In some cases, the PRO allows for marginally hotter doses at smaller concentrations (i.e., the hottest dose to the heart for the nominal model, and hotter D0.5% values across the board). However, as the majority of each organ is considered in the average doses and higher, the calculated dose

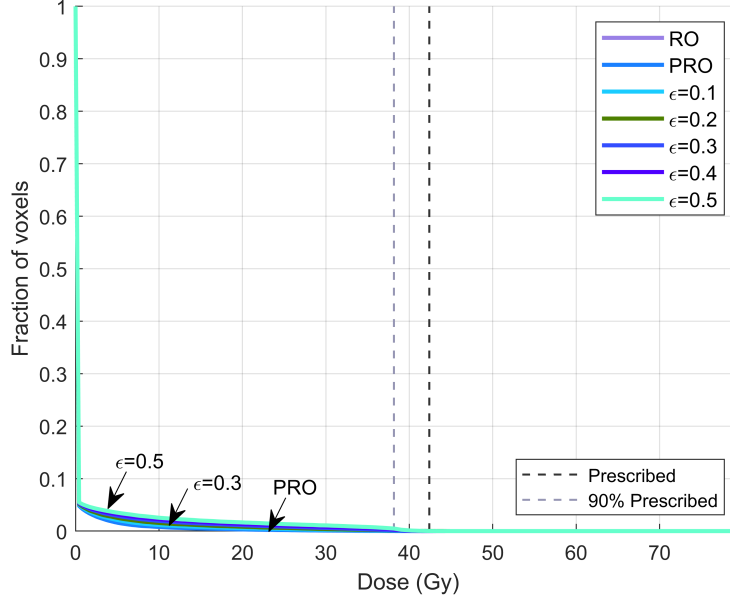
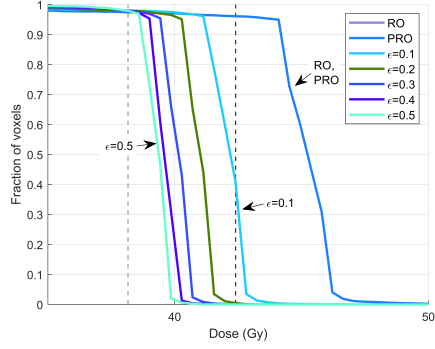


Figure 3.6: Comparison of heart DVH plots for the RO, PRO and light PRO model results with the $\mathcal{U}^M$ uncertainty set and a nominal realized $\mathbf{p}$.

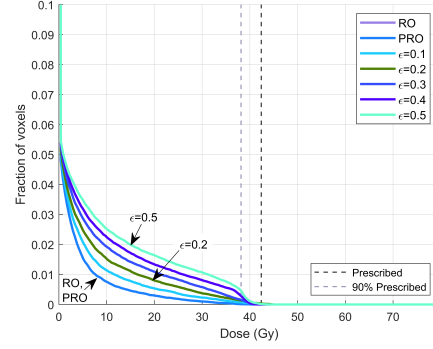
is slightly lower. This gives us some insight into what the PRO plan is trading off in order to gain lower tumor doses on average.

The dose-volume metrics also show that the light PRO models gradually sacrifice heart dose for excess tumor dose, and then perhaps move beyond that into necessary tumor dose. The average heart dose, $\text{Avg}_{\mathcal{H}} \text{Dose}$, trade-off is clear in the nominal model, where for every .1 Gy it gives up, the uniformity of the treatment increases and overdose decreases. Finally, the D10cc column shows the increase in hot spots as more dose is given to the heart, a trade-off that would need to be considered by a clinician.

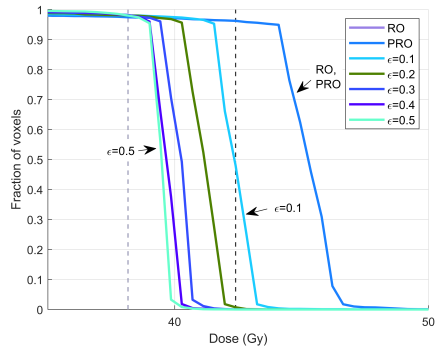
One last approach to quantify the trade-off between tumour control and heart dose offered by the LPRO model is by plotting either the gain function, or the second-term loss derived in Section 3.3. Figure 3.8 captures this function as the relative improvement of the tumour-control objective and ϵ for each of the three uncertainty sets with a nominal



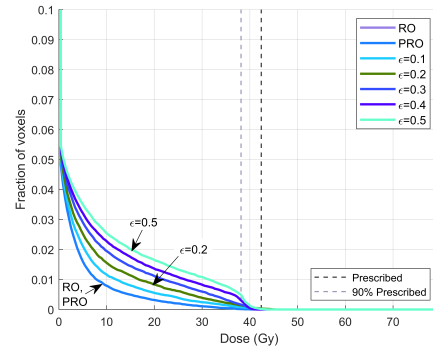
(a) Tumor dose with $\mathcal{U}^S$



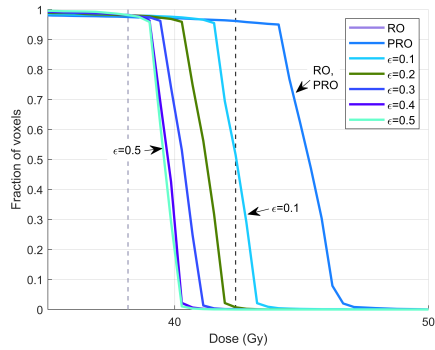
(b) Heart dose with $\mathcal{U}^S$



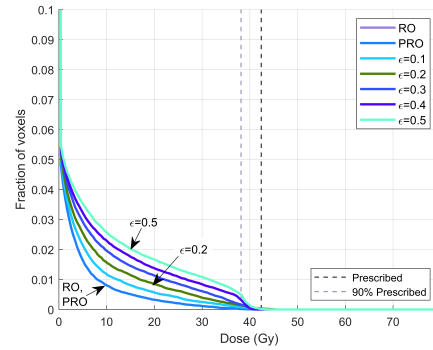
(c) Tumor dose with $\mathcal{U}^M$



(d) Heart dose with $\mathcal{U}^M$



(e) Tumor dose with $\mathcal{U}^L$



(f) Heart dose with $\mathcal{U}^L$

Figure 3.7: Plots (a), (c) and (e) show the tumour with a nominal realized $\mathbf{p}$ vector for each model and uncertainty set size, and the dose range condensed to the critical region. Plots (b), (d) and (f) depict the heart dose, with a smaller fraction of the voxels for clearer viewing.

Model	Realized $\mathbf{p}$	$\text{Avg}_{\mathcal{T}} \text{Dose}$	D0.5%	D10%	D90%	D95%	$\text{Avg}_{\mathcal{H}} \text{Dose}$	D10cc
RO	Nominal	44.7375	47.9429	46.1972	44.2688	44.1936	0.2875	20.9661
	w-Exahle	44.6980	48.0760	46.2267	44.2020	44.0646	0.2552	20.0903
	w-Inhale	44.8023	48.2398	46.2519	44.2068	44.0557	0.3538	23.5231
PRO	Nominal	44.7374	47.9434	46.1972	44.2687	44.1935	0.2875	20.9666
	w-Exahle	44.6979	48.0777	46.2267	44.2019	44.0646	0.2552	20.0903
	w-Inhale	44.8022	48.2399	46.2519	44.2067	44.0556	0.3538	23.5231
PRO$_{\epsilon=.1}$	Nominal	42.0956	44.0621	42.9446	41.7318	41.6519	0.3875	27.5815
	w-Exahle	42.0629	44.1268	42.9842	41.6303	41.4528	0.3394	26.7878
	w-Inhale	42.1503	45.1020	43.0657	41.6529	41.4403	0.4864	30.4950
PRO$_{\epsilon=.2}$	Nominal	41.0119	42.6681	41.6949	40.5865	40.4879	0.4875	32.7623
	w-Exahle	40.9785	42.7064	41.7314	40.4886	40.2869	0.4272	32.2855
	w-Inhale	41.0678	43.3321	41.7971	40.5207	40.3163	0.6114	34.2892
PRO$_{\epsilon=.3}$	Nominal	40.1670	41.4429	40.7238	39.7195	39.6720	0.5875	35.6350
	w-Exahle	40.1383	41.5416	40.7556	39.6351	39.4501	0.5187	35.2655
	w-Inhale	40.2154	42.4855	40.8158	39.6611	39.4669	0.7287	36.3634
PRO$_{\epsilon=.4}$	Nominal	39.6473	40.9193	40.1136	39.2272	39.2136	0.6875	38.2675
	w-Exahle	39.6229	40.8049	40.1464	39.1401	38.9823	0.6098	38.1895
	w-Inhale	39.6884	41.7501	40.1973	39.1721	39.0085	0.8473	38.6019
PRO$_{\epsilon=.5}$	Nominal	39.4968	40.5882	39.9046	39.1185	39.1184	0.7875	38.9138
	w-Exahle	39.4740	40.5787	39.9344	39.0282	38.8707	0.7021	38.8177
	w-Inhale	39.5348	41.3171	40.0161	39.0628	38.9051	0.9629	39.3141

Table 3.4: Dose metrics for nominal and extreme breathing patterns for each model run with $\mathcal{U}^L$.

realization $\mathbf{p}$. Similar to the conclusion reached for the DVH diagram, it can be seen that most of the slack from each model is removed by $\epsilon = 0.3$ and there are diminishing returns, i.e., a flatter slope, for the larger ϵ -values afterwards. The line isn't completely flat, so if the user is willing to sacrifice more average heart dose, the model overdose and underdose could likely be further reduced, by choosing beamlets that have a greater impact on the heart.

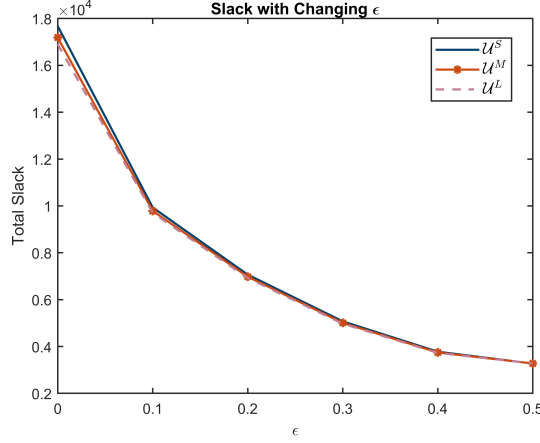


Figure 3.8: Tumor over and underdose (slack in the constraints) given nominal realized $\mathbf{p}$ values for each ϵ .

3.6 Discussion

It is evident that in terms of plan conformity, patients benefit from some degree of relaxation of the original RO model. What remains unclear, however, is which trade-offs are more important and whether all experts would evaluate all trade-offs in an equivalent manner. It is noteworthy, for example, that the visual DVH evaluation for the five basic patients might lead to one set of ϵ values (i.e., $[0.2, 0.2, 0.2, 0.3, 0.1]$) based on diminishing returns of the diagrams noted above, whereas a similar trade-off evaluation at the D99% level leads to a different set of ϵ decisions for some patients ($[0.1, 0.1, 0.1, 0.3, 0.1]$). Furthermore, by changing the modelling type, as in the CVaR case for patient 5, the trade-offs change once again. A future avenue of research might be sequencing and visualizing trade-offs in an informative manner for clinicians, perhaps by using visualization software, or through automatic multi-criteria evaluation techniques from the literature (Zarepisheh et al., 2014; Ripsman et al., 2015; Hasan et al., 2019).

A potential benefit of the proposed two-stage methodology is the opportunity for intervention and clinician guidance. After running the RO for Patient 1, for example, a clinician

might look at an average heart dose of 0.08 Gy and immediately set the value of epsilon to 0.2 Gy, seeing as there is a lot of room for improvement in the plan there. For a patient like Patient 4, they might be more careful and move iteratively, since the patient already has a high average dose to the heart. This potential for interaction provides additional incentive for integrating [LPRO](#) into a future planning software, over more black-box approaches.

Another benefit of [LPRO](#) is that each subsequent optimization remains linear, and is thus fairly fast to resolve. This provides an opportunity for decision-makers to interact with the solver, and give timely expert input, while still reducing the overwhelm felt by clinicians day to day. To generate the ultimate treatment, however, some form of leaf-sequencing or aperture generation would need to follow the optimization. Alternatively, this mode of modelling could be incorporated into a [DAO](#) framework. Currently, however, that would make each iteration far too time-consuming for practical iterative expert feedback.

3.7 Conclusions

The light PRO modelling approach generates high-quality [IMRT](#) plans, without sacrificing robustness guarantees or non-worst-case performance as was seen in past [RO](#) models. The relaxation of the primary objective function value allows for a controlled examination of the solution space, without losing sight of the total irradiation goal at hand. As the [LPRO](#) model was allowed more and more freedom, dose drop-offs increased, creating clinically viable plans for patients. If too much freedom was given, however, the model began to increasingly neglect its original objective and overdose the organ at risk. A number of approaches to trade-off analysis were possible using different visualization and dose-trade-off philosophies. This study shows that there are many benefits to the light PRO method, and that a loss of robustness is not strictly necessary for improving dose conformity in [IMRT](#) treatment planning.

Chapter 4

A Geometrical Approach to Beam Angle Optimization

Beam angle optimization is one of the theoretically hardest parts of [RTTP](#) because it can have virtually unlimited degrees of freedom, and technically requires output from the [FMO](#) and [LS](#) stages of the planning process in order to be performed optimally ([Cao et al., 2012](#)). While many sophisticated [BAO](#) methods have been proposed in the literature, in clinic, typical practice entails arbitrarily selecting 3–9 equidistant beams, unless recommended otherwise by an expert clinician ([Romeijn and Dempsey, 2008](#); [Bertsimas et al., 2013](#); [Amit et al., 2015](#)). When expert clinicians do play a part, the decision-making process can be quite time-consuming and iterative, involving repeat angle-selections and runs of an ([FMO](#)) model to assess quality ([Amit et al., 2015](#)).

One possible approach to eliminating iteration is by combining [FMO](#) and [BAO](#) into a single optimization. However, this combination similarly runs into problems, as the combined problem is highly non-linear, and non-convex, with multiple local minima ([Bertsimas et al., 2013](#)). Thus, modern methods tend towards non-optimal, multi-start solvers that integrate [FMO](#) calculations, as in [Aleman et al. \(2009\)](#) and [Bertsimas et al. \(2013\)](#). These

approaches have theoretical bounds at infinity, but are run a finite number of times. [Rocha et al. \(2019\)](#) employs a similar methodology, but argues that the [FMO](#) method does not guarantee optimal tumour quality control and opts for different treatment metrics (however, [FMO](#) remains standard across the rest of the literature). [Lee et al. \(2003\)](#) combines the [FMO](#) and [BAO](#) frameworks into a single model, but in doing so, incorporates limitations associated with both problems: the [BAO](#) selection must be quite limited (16 beam options), and rely on in-house beam approximations, while the [FMO](#) part of the problem is simplified, resulting in a method that proposes a full solution, but is unlikely practical for clinical use.

Since the [BAO](#) problem alone is the source of a lot of the complexity, there also exists a number of non-optimal solvers that do not claim to solve the two optimizations at once, e.g., genetic, conjugate gradient and simulated annealing ([Li et al., 2004](#); [Lu et al., 1997](#)) as well as machine learning approaches like random forest, as in [Amit et al. \(2015\)](#). Note that the beam’s eye view perspective, adopted in [Lu et al. \(1997\)](#) has similarities to the one introduced in this report, in that it relies on the geometrical presence of an organ in the field. It is, however, a more macroscopic simulated annealing-based method that chooses full couch and beam angles simultaneously, rather than individual beamlets.

The goal of this proposed work is to develop a [BAO](#), using more modern and sophisticated techniques. Ideally, the ultimate outcome of this exploration is a model that can be solved using accessible technology, is not very resource-intensive, and can provide better solutions than the equidistant clinical standard. In the short term, the goal is a proof of concept for the methodology. Note the approach does not seek to replace the [FMO](#) or [LS](#) steps of the problem, but instead to reduce or remove the iterations leading up to these steps, with fewer simplifying assumptions. Unlike many modern methods, the proposed approach does not require sophisticated dose calculators, which yield the $D_{v,b}$ matrix from previous chapters. Instead, it favours geometry, and works more analogously to some conceptual Fourier transform measures, discussed in [Soderstrom and Brahme \(1992\)](#).

The contributions of this chapter include a proposed:

1. method for reducing the [BAO](#) search space to a finite set of angles, geometrically
2. NP-hard set-covering formulation for choosing an optimal selection of angles
3. column-generation and heuristic approach for the set-covering formulation.

Section [4.1](#) describes the geometrical methods employed. Section [4.2](#) develops the methodology for producing the aperture options and proposes a model for the selection. Section [4.3](#) discusses solution and model-population approaches. Section [4.4](#) demonstrates the use of the most practical, complete heuristic generation approach, to generate a small 2D set of results and finally, Section [4.5](#) closes out the chapter.

4.1 Problem Definition

The [BAO](#) problem begins with two grids: the *3-dimensional* (3D) grid of patient voxels (i.e., discrete slicing of a [CT](#) scan), and the *2-dimensional* (2D) grid of beamlets, (i.e., the discretized mapping of the beam of radiation). A 2D representation of the relationship between beamlets and voxels, as governed by gantry motion, is depicted in [Figure 4.1](#).

As the gantry rotates, the geometrical relationship between each beamlet and voxel changes in a predictable manner. To make the problem more tractable, it is simplest to begin with a number of assumptions. First, assume each beamlet can “see” different voxels, as it rotates to a given angle, as depicted in [Figure 4.2a](#). In reality, different structures have different absorbency rates, and electrons interact with one-another, meaning voxels are not limited by, or even guaranteed to be seen by beamlets in such a restrictive manner. This is why sophisticated planning software will typically incorporate proper dose calculation engines such as Monte Carlo simulations to calculate the dose deposition from each beam ([Hurkmans et al., 1995](#); [Dorje, 2014](#); [Pandur et al., 2022](#)). The hypothesis of

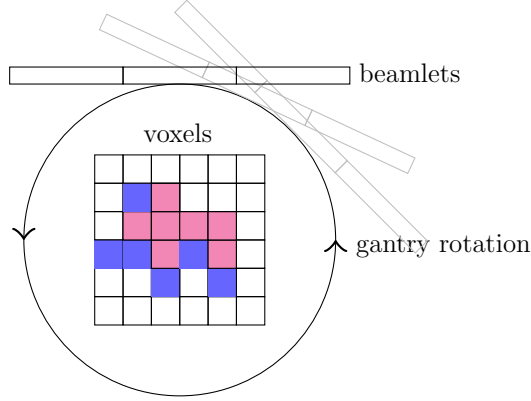


Figure 4.1: A 2D depiction of a beam’s path on a gantry around a patient. The darker blue voxels represent healthy OARs and the lighter pink voxels are the CTV. The fixed beam is sitting at 90° , while the lighter beamlets represent rotations to, 45° and 65° .

this chapter, however, is that the partitioning of seen and unseen voxels will serve as a good approximation for reality, which can ultimately be tested by assessing the [FMO](#) plan quality calculated with proper $D_{v,b}$ ’s, calculated at the recommended dose angles.

A second abstraction comes from the observation that in some cases, a given beamlet at a certain angle may see portions of a voxel. E.g., in [Figure 4.2a](#), notice that some voxels, such as the [OAR](#) voxel in the upper left corner, are half-covered and half-exposed by an active beamlet. This may be abstracted out of the problem by assuming each voxel is a single grid-point, i.e., assume the patient grid is granular enough that any voxel may be considered completely included or completely excluded from a beamlet’s perspective, as in [Figure 4.2b](#).

Finally, the third assumption is that the beam will have an equal effect on all voxels within its path, as depicted in [Figure 4.2c](#). Note that while overall these features help with abstracting the problem, the latter two may ultimately be loosened in practice. With all the abstractions depicted in [Figure 4.2](#), formulas may be derived to determine where it is beneficial to stop the beam, i.e., which angles to select.

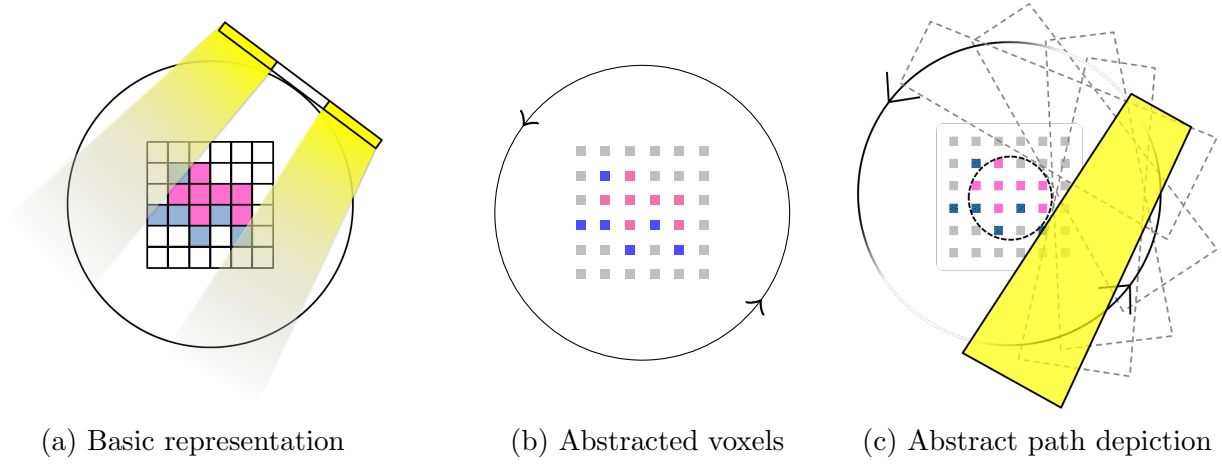


Figure 4.2: Three 2D abstractions for understanding the beam-rotation problem. Both (a) and (c) depict active beamlets, (a) with both outer beamlets on and (c) depicting the path of the right-most beamlet over a rotation.

4.2 Methods

In this section, two approaches to solving the problem are introduced in two-dimensions (Sections 4.2.1 and 4.2.2), then extensions to the third dimension are discussed for each in Section 4.2.3. Finally, the models used to solve the problem are described in Section 4.2.4.

4.2.1 2-Dimensional Beam Rotation Approach

In order to model the problem, the system must be described on a set of coordinate axes, beginning with the voxel locations. Given some central or focal point of the gantry rotation, referred to in radiation therapy as an *isocenter*, every voxel can be described as a point on a coordinate system. In that system, $(0,0)$ is the isocenter, as depicted in Figure 4.3a. Additionally, the partial beamlets from the previous section, can be illustrated in full as originating from a point source within the gantry, as shown in Figure 4.3b. Figure 4.3c then shows the meeting point of each beamlet boundary at the depth of the isocenter, given the initial setup at 0° .

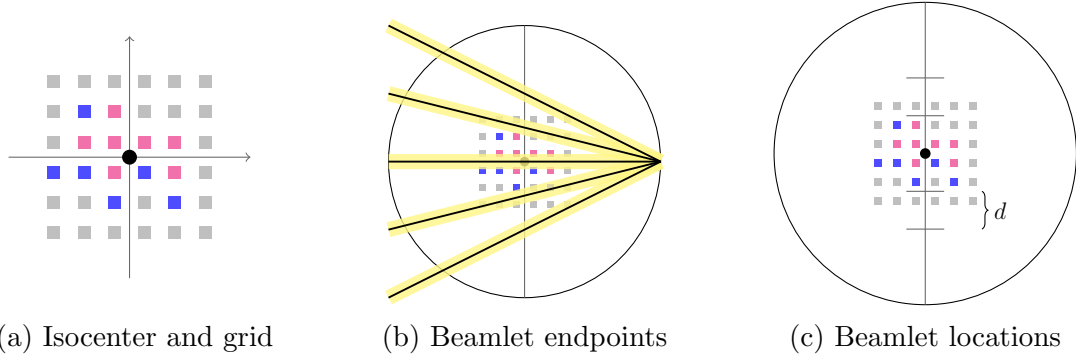


Figure 4.3: The rotating system related to its 2D coordinate grids. The isocenter is indicated by the black circle, and the beamlet distances are shown along the axis containing the isocenter, perpendicular to the radiation point-source.

Note, the size of the beamlet is defined on this indicated plane of the isocenter, within the patient. In other words, a $d = 5\text{mm}$ beamlet is defined as a 5mm distance between beamlet boundaries within the patients body at a specified depth, not at the mouth of the gantry, where beamlets are typically depicted. The mouth of the gantry, on the other hand, contains the controls that determine which beamlets are turned on or off, which will then be adjusted for the full extent of beamlet deflection at the depth of the isocenter. An abstract image with the beamlets is provided in Figure 4.4.

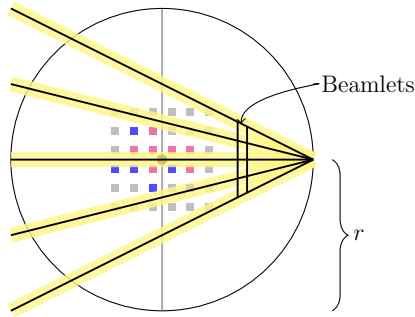


Figure 4.4: Beamlets depicted in their correct position in space, between the patient and the point source at starting angle 0° .

Assume that the isocenter is r units away from the point source, and there is a distance

of d_b between each given beamlet-boundary b in the set of all beamlets $\mathcal{B}$, and the origin, i.e., the isocenter (this is different than the size of the beamlet, d). Note that there is one additional starting edge that will not be in set $\mathcal{B}$, since each adjacent beamlet shares two boundaries.

Then, the line formulation for each beamlet boundary in the setup at 0° can then be formulated as,

$$y = \frac{d_b - 0}{0 - r}x + d_b \quad \forall b \in \{0 \cup \mathcal{B}\}.$$

As the gantry rotates the entire beam around the isocenter, each plane, or beamlet boundary, individually rotates around the same focal point as well, as shown in Figure 4.5.

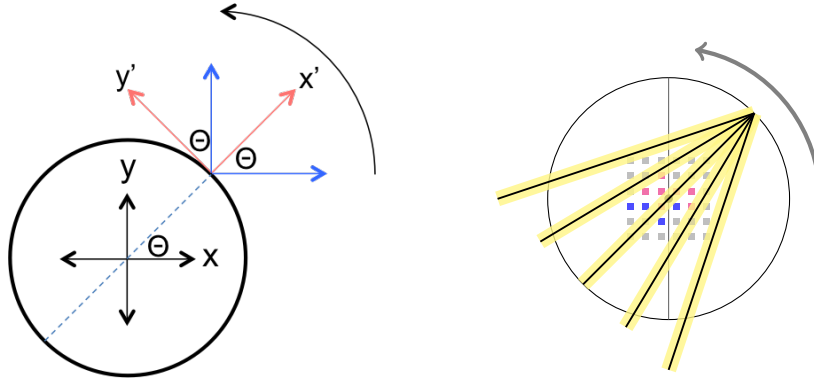


Figure 4.5: Rotating the full beam by $\theta = 45^\circ$.

This can be viewed as tilting the axis (around the origin) yielding the exact formula for the lines, but on an alternate coordinate system with coordinates (x', y') ,

$$y' = -\frac{d_b}{r}x' + d_b \quad \forall b \in \{0 \cup \mathcal{B}\}.$$

Or equivalently,

$$y \cos \theta - x \sin \theta = -\frac{d_b}{r}(x \cos \theta + y \sin \theta) + d_b \quad \forall b \in \{0 \cup \mathcal{B}\},$$

which simplifies to,

$$y = \frac{x(\sin \theta - \frac{d_b}{r} \cos \theta) + d_b}{\cos \theta + \frac{d_b}{r} \sin \theta} \quad \forall b \in \{0 \cup \mathcal{B}\}. \quad (4.1)$$

Note, when $\cos \theta + \frac{d_b}{r} \sin \theta = 0$, i.e., when the boundary line becomes vertical, y is undefined. At all other angles, a given voxel with coordinates (x, y) falls exactly on the line, when it satisfies the equation. Moreover, when it falls between two adjacent lines, it is “seen” by the corresponding beamlet.

4.2.2 2-Dimensional Patient Rotation Approach

Since the focal point of the beam’s rotation is the middle of the patient, a second approach to conceptualizing the 2D problem is to imagine that rather than the beamlets rotating, the patient is rotating and the beamlets are staying still. See Figure 4.6 for an example of patient rotation. Notice, the patient “rotates” in the opposite direction of the beam, to recreate the previous rotating effect.

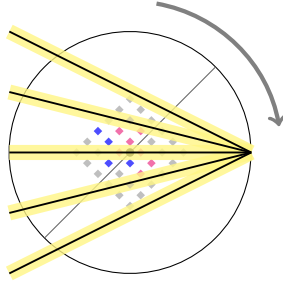


Figure 4.6: Rotating the patient-voxel grid by $\theta = -45^\circ$.

A voxel v at location (x_v, y_v) intersects with a beamlet boundary when it is entering or exiting the field. So in this case, r_v is relevant, denoting the radius of a given voxel from the isocenter. Using this radius, a circular path can be created that voxel v will follow throughout the rotation, as in Figure 4.7.

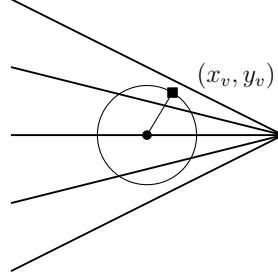


Figure 4.7: Voxel v rotation summary. It is immediately evident that the selected voxel will spend some time within all four beamlets.

This setup means that there are at most 2 critical points per voxel per beamlet angle: representing when a voxel enters and exits the beam.

$$y = -\frac{d_b}{r}x + d_b \quad \text{intersects with} \quad x^2 + y^2 = r_v^2.$$

This happens when,

$$\left(x_{v,b}^{int} = \frac{\frac{d_b^2}{r} \pm \sqrt{\frac{d_b^2 r_v^2}{r^2} - d_b^2 + r_v^2}}{1 + \frac{d_b^2}{r^2}}, \quad y_{v,b,\pm}^{int} = -\frac{d_b}{r}x_{v,b,\pm}^{int} + d_b. \right)$$

Note, since it is expected that $r \gg d_b$, the value of x will always be positive. When the square root term, $\frac{d_b^2 r_v^2}{r^2} - d_b^2 + r_v^2 < 0$, the two geometries do not intersect, and if the equality is strict, the line is a tangent to the circle, so they intersect exactly once.

Finding the angular intersection distance from the original voxel location within the

$0^\circ \leq \theta \leq 360^\circ$, then involves taking the dot products of the intersection points, so,

$$\theta_{v,b,\pm}^{int} = \cos^{-1} \left(\frac{x_v x_{v,b,\pm}^{int} + y_v y_{v,b,\pm}^{int}}{r_v^2} \right).$$

Since the gap between the starting voxel location and intersection coordinate already accounts for the voxel's starting coordinates, finding θ relative to the full beam rotation is simply,

$$\theta_{v,b,\pm} = 360^\circ - \theta_{v,b,\pm}^{int}.$$

4.2.3 Extensions to the 3rd Dimension

The 3D rotation works similarly to the 2-dimensional cases, except rather than passing through lines, voxels now pass through 3-dimensional beamlets, made up of four planes, like the trapezoidal prism, depicted in Figure 4.8a. The outward fluctuation of the beam in four different directions, makes the system a bit more difficult to conceptualize, so more clarifying assumptions are made.

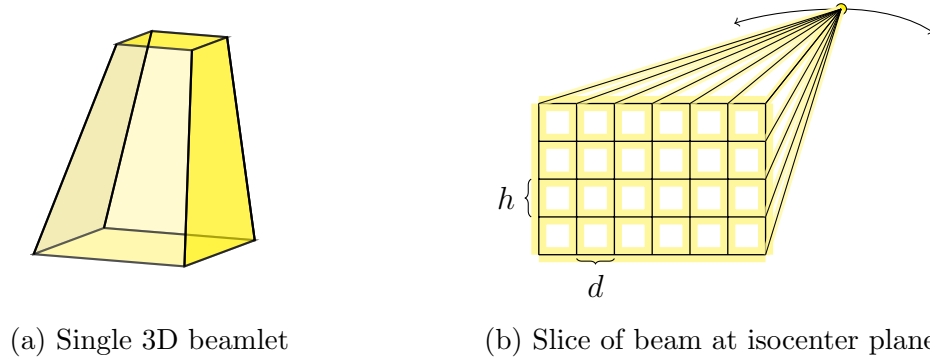


Figure 4.8: Different renderings of the 3D beam components. As the beamlet fluctuates outwards from the source, at the perpendicular plane, the beamlets are rectangular. This system, when viewed from above, matches the 2D images from the previous section.

First, assume rotation around the isocenter is still a direct 360° , with no twisting or

rotating motion. Second, as is the convention in planning systems, assume that each beamlet is properly rectangular at the cross-sections perpendicular to the point source, as shown in Figure 4.8b. This means that only two dimensions of rotation must be considered, i.e., the system only rotates in the $x - y$ plane. In other words, the vertical inclusion decisions are made in two dimensions, whereas the inclusion of voxels in a horizontal plane must also require the z -dimension. Note, a fixed rotation of the bench or patient throughout treatment is still allowable even with this simplification, as voxels can trivially be projected onto the required rotating $x - y$ plane in a pre-processing step.

The beamlet size in the z -dimension governs the depth of a beamlet at a given point in the beam (i.e., picture the beamlets going into the plane in the 2D images from the previous sections). Once again, while the beamlet size does not need to be constant, it is assumed to be, for ease of notation and is the height of each beamlet is denoted with h . The indexed value h_b indicates the total from the isocenter to the beamlet boundary in the z -domain at the isocenter plane. Finally, we assume that rows and columns within set $\mathcal{B}$ are represented by sets $\mathcal{Q}$ and $\mathcal{N}$, respectively, where $\mathcal{Q} := \{1, \dots, |\mathcal{Q}|\}$ and $\mathcal{N} := \{1, \dots, |\mathcal{N}|\}$.

To handle direct beam rotation, a formula for the planes rather than lines must be derived. From the perspective of the beam's cross section, this is a horizontal set of planes, indicating depth, and a vertical set of planes, indicating distance as derived in the 2D case.

For the horizontal planes, there are 3 sets of (x, y, z) points. The point that describes the point source is a given, $p_1 = (r, 0, 0)$, as it is r units away from the isocenter, but exactly aligned with the centre in both other planes. Next, simply choose a point on the plane perpendicular to the isocenter, $p_2 = (0, 0, h_b)$ and an adjacent beamlet, d units away $p_3 = (0, d, h_b)$, which is three points on the plane. This leads to two vectors, $p_1 - p_2 = (r, 0, -h_b)$ and $p_1 - p_3 = (r, -d, -h_b)$, which can be used to calculate the normal vector,

$$n = \begin{vmatrix} i & j & k \\ r & 0 & -h_b \\ r & -d & -h_b \end{vmatrix} = \langle -h_b d, 0, -rd \rangle,$$

and equation,

$$-h_b dx + 0y - rdz + D = 0.$$

If a point, e.g., p_1 is subbed in, that's $D = h_b dr$. So the full equation is,

$$h_b x + rz = h_b r.$$

In rotating the beam, x is once again replaced with x' to get the formula,

$$x h_b \cos \theta + y h_b \sin \theta + rz = h_b r \quad \forall b \in \{0 \cup \mathcal{Q}\}. \quad (4.2)$$

Similarly, for the vertical plane,

$$d_b x + ry = d_b r.$$

This works out to the following with rotation

$$d_b(x \cos \theta + y \sin \theta) + r(y \cos \theta - x \sin \theta) = d_b r,$$

and simplifies to

$$(d_b \cos \theta - r \sin \theta)x + (d_b \sin \theta + r \cos \theta)y = d_b r \quad \forall b \in \{0 \cup \mathcal{K}\}. \quad (4.3)$$

Determining which planes surround the voxels will now indicate if voxel v , located at (x_v, y_v, z_v) , is currently covered by beamlet b at rotation θ .

Once again, however, the beam can also be seen as static, and the patient rotating

backwards. In this case, given the z_v location of every voxel v (assuming the patient voxels have been projected to prevent rotation in the z dimension, as discussed previously), r_v can be kept in the $x - y$ plane. This leaves $x_{v,b,\pm}^{int}$ and $y_{v,b,\pm}^{int}$ and just as in the 2D version, because the beam is rotating in a circle, every beamlet may be entered and exited once every 180° . This means it can still only intersect each *plane* zero, one or two times; as captured in the following two formulas,

$$\text{horizontal: } h_b x + r z = h_b r \quad \text{intersects with} \quad x^2 + y^2 = r_v^2, \quad z = z_v$$

and

$$\text{vertical: } d_b x + r y = d_b r \quad \text{intersects with} \quad x^2 + y^2 = r_v^2, \quad z = z_v.$$

For the horizontal intersection, we have,

$$\left(x_{v,b}^{int-h} = r - \frac{r z_v}{h_b}, \quad y_{v,b}^{int-h} = \pm \sqrt{r_v^2 - \left(r - \frac{r z_v}{h_b} \right)^2}, \quad z_{v,b}^{int-h} = z_v \right) \quad (4.4)$$

meaning if $r_v^2 - \left(r - \frac{r z_v}{h_b} \right)^2 \geq 0$, the paths will intersect.

Similarly, for the vertical, the equation simply rearranges to the same equation as in the 2D case, which makes sense, since the plane is expected to be the same throughout the z -axis, so the following are the set of intersections for each plane,

$$\left(x_{v,b}^{int-v} = \frac{\frac{d_b^2}{r} \pm \sqrt{\frac{d_b^2 r_v^2}{r^2} - d_b^2 + r_v^2}}{1 + \frac{d_b^2}{r^2}}, \quad y_{v,b,\pm}^{int-v} = -\frac{d_b}{r} x_{v,b,\pm}^{int-v} + d_b, \quad z_{v,b}^{int-v} = z_v \right). \quad (4.5)$$

Once again, the beamlet boundaries only intersect a given voxel if $\frac{d_b^2 r_v^2}{r^2} - d_b^2 + r_v^2 \geq 0$.

This means the following function can still be used to retrieve $\theta_{v,b,\pm}^{int}$:

$$360^\circ - \cos^{-1} \left(\frac{x_v x_{v,b,\pm}^{int-*} + y_v y_{v,b,\pm}^{int-*}}{r_v^2} \right), \quad (4.6)$$

where $int - *$ denotes any intersection of interest.

4.2.4 Model

Given the availability of the fundamentals from the previous section, a model can be built and populated with beamlet-voxel relationship information in mind. The following section proposes a set-covering model, that embeds a simplified version of the [FMO](#) model. In essence, due to the absence of dose-influence information to capture the entire [FMO](#) model, a preview of this model is optimized over instead, using the dose path indicators as proxies. Because of incomplete dose-influence information, this model does not *guarantee* the best selection of apertures in the final plan, however, it will ensure that all [CTV](#) voxels, $\mathcal{V}_{\mathcal{T}}$, can be “seen” by beamlets that have a relatively low cost to the [OAR](#) voxels in $\mathcal{V}_{\mathcal{H}}$. It uses the ranges from the previous section – a process that is detailed in the next section but we take as given for now – to form constant tensors $\mathbf{T}$ and $\mathbf{H}$, where,

$$T_{v,b,\theta} = \begin{cases} 1, & \text{if voxel } v \in \mathcal{V}_{\mathcal{T}} \text{ receives radiation from beamlet } b \text{ at angle } \theta \\ 0, & \text{otherwise,} \end{cases}$$

$$H_{v,b,\theta} = \begin{cases} 1, & \text{if voxel } v \in \mathcal{V}_{\mathcal{H}} \text{ receives radiation from beamlet } b \text{ at angle } \theta \\ 0, & \text{otherwise.} \end{cases}$$

Note, $H_{v,b,\theta}$ can be efficiently aggregated over $v \in \mathcal{V}_{\mathcal{H}}$ to be total OAR voxels to receive radiation, however, leaving it disaggregated makes it easier to pivot to a max-dose-to-OAR penalty. In terms of variables, y_θ indicates if angle $\theta \in \Theta$ is active and $x_{b,\theta}$ is the amount of the required dose supplied by beamlet $b \in \mathcal{B}$ at angle $\theta \in \Theta$.

The model is formulated as follows,

$$\min \sum_{v \in \mathcal{V}_H} \sum_{b \in \mathcal{B}} \sum_{\theta \in \Theta} H_{v,b,\theta} x_{b,\theta} \quad (4.7a)$$

$$\text{s.t.} \sum_{b \in \mathcal{B}} \sum_{\theta \in \Theta} T_{v,b,\theta} x_{b,\theta} \geq 1 \quad \forall v \in \mathcal{V}_T, \quad (4.7b)$$

$$\sum_{\theta \in \Theta} y_{\theta} \leq A, \quad (4.7c)$$

$$x_{b,\theta} \leq y_{\theta} \quad \forall b \in \mathcal{B}, \theta \in \Theta, \quad (4.7d)$$

$$y_{\theta} \in \{0, 1\} \quad \forall \theta \in \Theta,$$

$$x_{b,\theta} \in \mathbb{R}_+ \quad \forall b \in \mathcal{B}, \theta \in \Theta,$$

where A is the total number of allowable angles. The objective (4.7a) is to minimize the total proportion of the dose that is likely going directly to the healthy organs. Constraints (4.7b) ensure each voxel in the treatment region can receive 100% of treatment dose with the selected angles. Constraints (4.7c) enforce an upper limit on the selected number of angles. Constraints (4.7d) are the disaggregated activity constraints that prevent turned-off beams from having active beamlets. The remainder of the constraints are domain constraints.

4.3 Solution Approaches

With the model proposed, there comes the issue of populating $\mathbf{T}$ and $\mathbf{H}$ appropriately. There are infinite angles between 0° and 360° , and therefore infinite possibilities of values. There are, however, several potentially finite options in terms of populating the tensors. In this section, a pricing sub-problem is proposed in Section 4.3.1. Followed by two potential sub-problem models to solve the pricing problem, in Sections 4.3.2 and 4.3.3, followed by an exact heuristic approach in Section 4.3.4.

4.3.1 Pricing Problem

Temporarily leaving off the number of aperture requirements, the primary goal of treating each target voxel may be considered, yielding the simpler model,

$$\begin{aligned}
& \min \sum_{v \in \mathcal{V}_{\mathcal{H}}} \sum_{b \in \mathcal{B}} \sum_{\theta \in \Theta} H_{v,b,\theta} x_{b,\theta} \\
& \text{s.t.} \sum_{b \in \mathcal{B}} \sum_{\theta \in \Theta} T_{v,b,\theta} x_{b,\theta} \geq 1 \quad \forall v \in \mathcal{V}_{\mathcal{T}}, \\
& \quad x_{b,\theta} \in \mathbb{R}_+ \quad \forall b \in \mathcal{B}, \theta \in \Theta.
\end{aligned}$$

If only a subset of angles $\Theta' \subset \Theta$ and $\mathcal{B}' \in \mathcal{B}$ are considered, this becomes the restricted CG master problem,

$$\begin{aligned}
& \min \sum_{v \in \mathcal{V}_{\mathcal{H}}} \sum_{b \in \mathcal{B}'} \sum_{\theta \in \Theta'} H_{v,b,\theta} x_{b,\theta} \\
& \text{s.t.} \sum_{b \in \mathcal{B}'} \sum_{\theta \in \Theta'} T_{v,b,\theta} x_{b,\theta} \geq 1 \quad \forall v \in \mathcal{V}_{\mathcal{T}}, \\
& \quad x_{b,\theta} \in \mathbb{R}_+ \quad \forall b \in \mathcal{B}', \theta \in \Theta'.
\end{aligned} \tag{4.8}$$

A pricing problem would subsequently use the dual values π_v from this problem to select a new beamlet and angle combination, b, θ , or declare the current problem optimal. The reduced costs of this problem would (theoretically, for all beamlets and angles) be,

$$\bar{c}_{b,\theta} = \sum_{v \in \mathcal{V}_{\mathcal{H}}} H_{v,b,\theta} - \pi_v \cdot T_{v,b,\theta}^T.$$

Based on these reduced costs, either no angle contains a beamlet with a negative reduced cost, or the beamlet-angle combination with the minimum reduced cost should be added to the master, i.e.,

$$\min_{b \in \mathcal{B}, \theta \in \Theta} \sum_{v \in \mathcal{V}_{\mathcal{H}}} H_{v,b,\theta} - \pi_v \cdot T_{v,b,\theta}^T. \tag{4.9}$$

4.3.2 Beam Rotation-Based Pricing

Given the pricing problem (4.9), a set of constraints is defined such that $\mathbf{T}$ and $\mathbf{H}$ correspond to an optimal θ selection. Each voxel v , defined by its (x_v, y_v, z_v) coordinates, is contained within a beamlet if it is in between its boundaries. Therefore, the model must enforce $\mathbf{T}$ and $\mathbf{H}$ to 1 at angles where this occurs and 0 otherwise, for each beamlet $b = 1, \dots, |\mathcal{B}|$, with its associated rows and columns $\mathcal{Q}$ and $\mathcal{N}$, where $b \leftarrow (q, n)$, as follows,

$$\min \sum_{v \in \mathcal{V}_\mathcal{H}} H_{v,b} - \sum_{v \in \mathcal{V}_\mathcal{T}} \pi_v T_{v,b}$$

$$\text{s.t. } x_v \left(\frac{d_{n-1}}{r} \cos \theta - \sin \theta \right) + y_v \left(\cos \theta + \frac{d_{n-1}}{r} \sin \theta \right) \geq d_{n-1} - (1 - T_{v,b})M \quad \forall v \in \mathcal{V}_\mathcal{T} \quad (4.10a)$$

$$x_v \cos \theta + y_v \sin \theta \geq r - \frac{rz_v}{h_{q-1}} - (1 - T_{v,b})M \quad \forall v \in \mathcal{V}_\mathcal{T} \quad (4.10b)$$

$$x_v \left(\frac{d_n}{r} \cos \theta - \sin \theta \right) + y_v \left(\cos \theta + \frac{d_n}{r} \sin \theta \right) \leq d_n + (1 - T_{v,b})M \quad \forall v \in \mathcal{V}_\mathcal{T} \quad (4.10c)$$

$$x_v \cos \theta + y_v \sin \theta \leq r - \frac{rz_v}{h_q} - (1 - T_{v,b})M \quad \forall v \in \mathcal{V}_\mathcal{T} \quad (4.10d)$$

$$M\bar{T}_{v,b} \geq x_v \left(\frac{d_{n-1}}{r} \cos \theta - \sin \theta \right) + y_v \left(\cos \theta + \frac{d_{n-1}}{r} \sin \theta \right) - d_{n-1} \quad \forall v \in \mathcal{V}_\mathcal{T} \quad (4.10e)$$

$$M\hat{T}_{v,b} \geq x_v \cos \theta + y_v \sin \theta - r + \frac{rz_v}{h_{q-1}} \quad \forall v \in \mathcal{V}_\mathcal{T} \quad (4.10f)$$

$$M\underline{T}_{v,b} \geq d_n - x_v \left(\frac{d_n}{r} \cos \theta - \sin \theta \right) - y_v \left(\cos \theta + \frac{d_n}{r} \sin \theta \right) \quad \forall v \in \mathcal{V}_\mathcal{T} \quad (4.10g)$$

$$M\hat{\underline{T}}_{v,b} \geq r - \frac{rz_v}{h_{q-1}} - x_v \cos \theta - y_v \sin \theta \quad \forall v \in \mathcal{V}_\mathcal{T} \quad (4.10h)$$

$$M\bar{H}_{v,b} \geq x_v \left(\frac{d_{n-1}}{r} \cos \theta - \sin \theta \right) + y_v \left(\cos \theta + \frac{d_{n-1}}{r} \sin \theta \right) - d_{n-1} \quad \forall v \in \mathcal{V}_\mathcal{H} \quad (4.10i)$$

$$M\hat{H}_{v,b} \geq x_v \cos \theta + y_v \sin \theta - r + \frac{rz_v}{h_{q-1}} \quad \forall v \in \mathcal{V}_\mathcal{H} \quad (4.10j)$$

$$M\underline{H}_{v,b} \geq d_n - x_v \left(\frac{d_n}{r} \cos \theta - \sin \theta \right) - y_v \left(\cos \theta + \frac{d_n}{r} \sin \theta \right) \quad \forall v \in \mathcal{V}_\mathcal{H} \quad (4.10k)$$

$$M\hat{\underline{H}}_{v,b} \geq r - \frac{rz_v}{h_{q-1}} - x_v \cos \theta - y_v \sin \theta \quad \forall v \in \mathcal{V}_\mathcal{H} \quad (4.10l)$$

$$T_{v,b} \geq \bar{T}_{v,b} + \underline{T}_{v,b} + \hat{T}_{v,b} + \hat{\underline{T}}_{v,b} - 3 \quad \forall v \in \mathcal{V}_\mathcal{T} \quad (4.10m)$$

$$H_{v,b} \geq \bar{H}_{v,b} + \underline{H}_{v,b} + \hat{H}_{v,b} + \hat{\underline{H}}_{v,b} - 3 \quad \forall v \in \mathcal{V}_\mathcal{H} \quad (4.10n)$$

$$T_{v,b}, \bar{T}_{v,b}, \underline{T}_{v,b}, \hat{T}_{v,b}, \hat{\underline{T}}_{v,b} \in \{0, 1\} \quad \forall v \in \mathcal{V}_\mathcal{T}$$

$$H_{v,b}, \bar{H}_{v,b}, \underline{H}_{v,b}, \hat{H}_{v,b}, \hat{\underline{H}}_{v,b} \in \{0, 1\} \quad \forall v \in \mathcal{V}_\mathcal{H}$$

$$0^\circ \leq \theta < 360^\circ.$$

Note, $\bar{\mathbf{H}}, \underline{\mathbf{H}}, \hat{\mathbf{H}}, \mathbf{H}, \bar{\mathbf{T}}, \underline{\mathbf{T}}, \hat{\mathbf{T}}$ and $\mathbf{T}$ are all auxiliary variables and M is a large number that may be replaced by the radius of the furthest voxel from the isocenter r_v , or r , as an even simpler shorthand. Constraints (4.10a) and (4.10c) are both exactly the vertical plane definitions. If a voxel is not $\geq$ the first plane and $\leq$ the second, $T_{v,b}$ must be 0 at the given angle. Similarly, (4.10b) and (4.10d) do the same for the horizontal planes. Constraints (4.10e) and (4.10g) set the auxiliary target voxel variables to 1 if the voxel is located between the vertical beamlet boundaries, and the constraints (4.10f) and (4.10h) do the same for the horizontal boundaries, respectively. Constraints (4.10i), (4.10k), (4.10j), (4.10l) enforce these vertical and horizontal boundaries for healthy voxels. Constraints (4.10m) and (4.10n) enforce target and healthy voxel patterns, respectively, to match that of the active voxels identified in by the upper and lower variables. Note that the healthy tissue must only be constrained in one direction seeing as the objective handles the opposing requirement, whereas the target tissue is constrained in both directions.

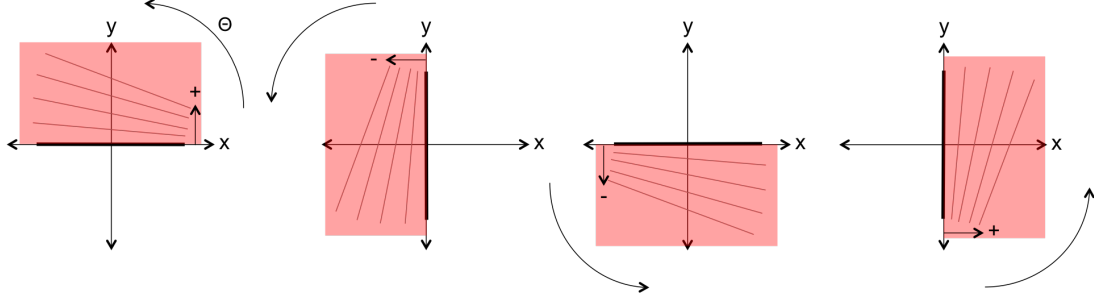
This is a difficult formulation to solve for two reasons:

1. Tractability - this is a non-linear problem that has many variables and constraints and will have to be solved for every beamlet.
2. The rotation of the beam changes the direction of the inequalities.

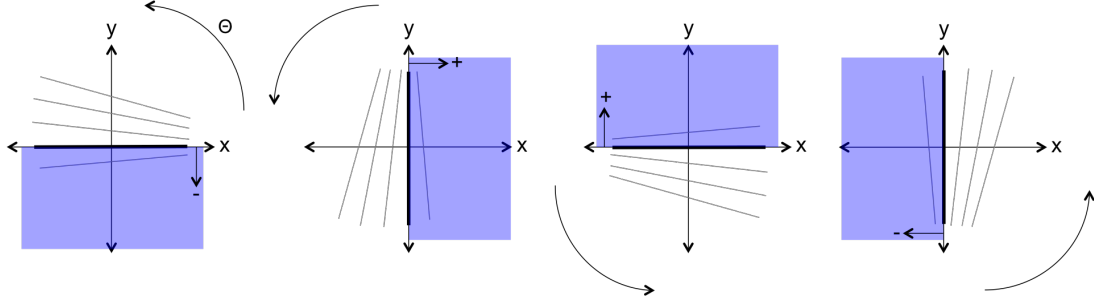
Difficulty 1 arises due to the trigonometric nature of the functions, which are naturally nonlinear. In order to incorporate this into solvers, typically extra binaries are used, and the problem is linearized in ranges. While this would be detrimental for the solver, it shouldn't be prohibitively large, particularly since the values of $\sin \theta$ and $\cos \theta$ are consistent throughout the model. However, given the abundance of existing binary variables and the fact that this has to be solved for every beamlet, and may not return an exact solution, it is not ideal.

Difficulty 2 arises as a result of beamlet rotation and its effect on the constraint direction in model 4.10. As depicted in Figure 4.9, as the direction that determines whether a voxel is

“in” a beamlet switches as the beam rotates through quadrants, past $\theta = 90^\circ$ and $\theta = 270^\circ$.



(a) Direction of inclusion for lower beam originating from Q1, Q2, Q3 and Q4.



(b) Direction of inclusion for upper beam originating from Q1, Q2, Q3 and Q4.

Figure 4.9: As the beam rotates through Q1 ($0-90^\circ$), the inclusive direction is **above** the bottom beam and **below** the upper beam, but from Q2 to Q3 ($90-270^\circ$), the inclusive direction is **below** the lower beam and **above** the upper beam. It then reverts back to being included **above** the bottom beam and **below** the upper in Q4 ($270-360^\circ$).

Also note that this switch is not consistent for each beamlet, as each one is offset from the total beam rotation by an angle relative to 0, denoted ϕ , as depicted in Figure 4.10. For a given pair of beamlet boundaries, there are two of these value that must be considered, which form critical points.

The inequality for $b = 0$ in the example above will reverse direction when $\theta = 90^\circ - \phi_0$, and will flip back again at $\theta = 270^\circ - \phi_0$. This must be accounted for in the model. The difference in angle between the two beams must also be accounted for, i.e., the lower line will straighten a few degrees sooner than the upper line. For those intermediate degrees,

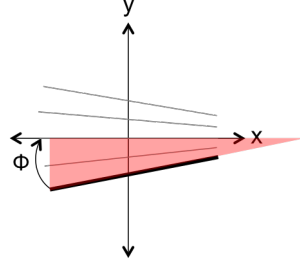


Figure 4.10: The lowest beam starts with an offset of ϕ at $\theta = 0^\circ$.

the inequalities should face the same direction (eg. from $90^\circ + \phi_0$ to $90^\circ + \phi_1$). This is depicted in Figure 4.11.

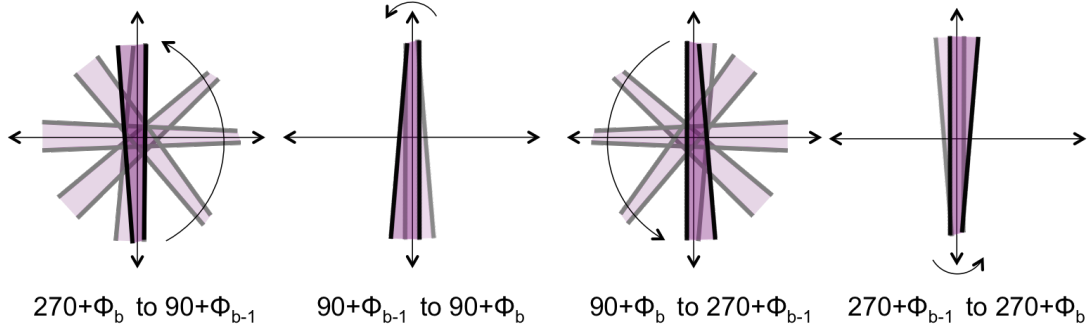


Figure 4.11: The sign pattern regions for a single beamlet, in terms of the overall beams θ value. Note the critical regions occur when each boundary hits 90° and 270° . Note, there is no critical region at 180° or at 360° , as, based on Figure 4.9, the direction of inclusion is the same both above and below the line.

Knowing that the beamlets will rotate according to these patterns, critical regions may be established for five possible scenarios.

1. If $\phi_b < 0^\circ$ when $\theta = 0^\circ$, i.e., the upper and lower bound on the beamlet both begin below $\theta = 0^\circ$, the critical points are:

$$\theta = \{0^\circ, 90^\circ + \phi_{b-1}, 90^\circ + \phi_b, 90^\circ, 180^\circ, 270^\circ + \phi_{b-1}, 270^\circ + \phi_b, 270^\circ\}.$$

2. If $\phi_{b-1} > 0^\circ$ when $\theta = 0^\circ$, i.e., both bounds are above $\theta = 0^\circ$, the sequence becomes:

$$\theta = \{0^\circ, 90^\circ, 90^\circ + \phi_{b-1}, 90^\circ + \phi_b, 180^\circ, 270^\circ, 270^\circ + \phi_{b-1}, 270^\circ + \phi_b\}.$$

3. If $\phi_{b-1} < 0^\circ$ and $\phi_b > 0^\circ$ when $\theta = 0^\circ$, i.e., $\theta = 0^\circ$ is between bounds, the sequence becomes:

$$\theta = \{0^\circ, 90^\circ + \phi_{b-1}, 90^\circ, 90^\circ + \phi_b, 180^\circ, 270^\circ + \phi_{b-1}, 270^\circ, 270^\circ + \phi_b\}.$$

4. If $\phi_{b-1} = 0^\circ$ when $\theta = 0^\circ$, the sequence becomes:

$$\theta = \{0^\circ, 90^\circ, 90^\circ + \phi_b, 180^\circ, 270^\circ, 270^\circ + \phi_b\}.$$

5. If $\phi_b = 0^\circ$ when $\theta = 0^\circ$, the sequence becomes:

$$\theta = \{0^\circ, 90^\circ + \phi_{b-1}, 90^\circ, 180^\circ, 270^\circ + \phi_{b-1}, 270^\circ\}.$$

Note that cases 4 and 5 can be collapsed into 1 and 2, respectively, with a $\leq$ and $\geq$ equalities. Partitioning and re-partitioning the sub-problems will lead to an even more impractical set of models for the solver, especially since linearization must be performed in each sub-problem as well. For that reason, alternative approaches were investigated.

4.3.3 Patient-Rotation-Based Pricing

Just like rotating the patient vs. the beam led to new equations in Section 4.2, it also allows for a re-framing of the pricing problem model. The same equations must be satisfied, but in this case, the planes are static, so the constraint directions will hold the entire time. Given that $x_v = r_v \cos(\tan^{-1}(\frac{y_v}{x_v}))$ and $y_v = r_v \sin(\tan^{-1}(\frac{y_v}{x_v}))$, if θ is added, these values should rotate accordingly. The patient-rotation-based pricing model therefore looks as follows for

every beamlet $b \in \mathcal{B}$,

$$\begin{aligned}
& \min \sum_{v \in \mathcal{V}_H} H_{v,b} - \sum_{v \in \mathcal{V}_T} \pi_v T_{v,b} \\
& \text{s.t. } \frac{d_{n-1}}{r} r_v \cos \left(\tan^{-1} \frac{y_v}{x_v} + \theta \right) + r_v \sin \left(\tan^{-1} \frac{y_v}{x_v} + \theta \right) \geq d_{n-1} - (1 - T_{v,b})M \quad \forall v \in \mathcal{V}_T \quad (4.11a) \\
& r_v \cos \left(\tan^{-1} \frac{y_v}{x_v} + \theta \right) \geq r - \frac{rz_v}{h_{q-1}} - (1 - T_{v,b})M \quad \forall v \in \mathcal{V}_T \quad (4.11b) \\
& \frac{d_n}{r} r_v \cos \left(\tan^{-1} \frac{y_v}{x_v} + \theta \right) + r_v \sin \left(\tan^{-1} \frac{y_v}{x_v} + \theta \right) \leq d_n + (1 - T_{v,b})M \quad \forall v \in \mathcal{V}_T \quad (4.11c) \\
& r_v \cos \left(\tan^{-1} \frac{y_v}{x_v} + \theta \right) \leq r - \frac{rz_v}{h_q} - (1 - T_{v,b})M \quad \forall v \in \mathcal{V}_T \quad (4.11d) \\
& M\bar{T}_{v,b} \geq \frac{d_{n-1}}{r} r_v \cos \left(\tan^{-1} \frac{y_v}{x_v} + \theta \right) + r_v \sin \left(\tan^{-1} \frac{y_v}{x_v} + \theta \right) - d_{n-1} \quad \forall v \in \mathcal{V}_T \quad (4.11e) \\
& M\hat{T}_{v,b} \geq r_v \cos \left(\tan^{-1} \frac{y_v}{x_v} + \theta \right) - r + \frac{rz_v}{h_{q-1}} \quad \forall v \in \mathcal{V}_T \quad (4.11f) \\
& M\underline{T}_{v,b} \geq d_n - \frac{d_{n-1}}{r} r_v \cos \left(\tan^{-1} \frac{y_v}{x_v} + \theta \right) - r_v \sin \left(\tan^{-1} \frac{y_v}{x_v} + \theta \right) \quad \forall v \in \mathcal{V}_T \quad (4.11g) \\
& M\hat{T}_{v,b} \geq r - \frac{rz_v}{h_{q-1}} - r_v \cos \left(\tan^{-1} \frac{y_v}{x_v} + \theta \right) \quad \forall v \in \mathcal{V}_T \quad (4.11h) \\
& M\bar{H}_{v,b} \geq \frac{d_{n-1}}{r} r_v \cos \left(\tan^{-1} \frac{y_v}{x_v} + \theta \right) + r_v \sin \left(\tan^{-1} \frac{y_v}{x_v} + \theta \right) - d_{n-1} \quad \forall v \in \mathcal{V}_H \quad (4.11i) \\
& M\hat{H}_{v,b} \geq r_v \cos \left(\tan^{-1} \frac{y_v}{x_v} + \theta \right) - r + \frac{rz_v}{h_{q-1}} \quad \forall v \in \mathcal{V}_H \quad (4.11j) \\
& M\underline{H}_{v,b} \geq d_n - \frac{d_{n-1}}{r} r_v \cos \left(\tan^{-1} \frac{y_v}{x_v} + \theta \right) - r_v \sin \left(\tan^{-1} \frac{y_v}{x_v} + \theta \right) \quad \forall v \in \mathcal{V}_H \quad (4.11k) \\
& M\hat{H}_{v,b} \geq r - \frac{rz_v}{h_{q-1}} - r_v \cos \left(\tan^{-1} \frac{y_v}{x_v} + \theta \right) \quad \forall v \in \mathcal{V}_H \quad (4.11l) \\
& T_{v,b} \geq \bar{T}_{v,b} + \underline{T}_{v,b} + \hat{T}_{v,b} + \hat{T}_{v,b} - 3 \quad \forall v \in \mathcal{V}_T \quad (4.11m) \\
& H_{v,b} \geq \bar{H}_{v,b} + \underline{H}_{v,b} + \hat{H}_{v,b} + \hat{H}_{v,b} - 3 \quad \forall v \in \mathcal{V}_H \quad (4.11n) \\
& T_{v,b}, \bar{T}_{v,b}, \underline{T}_{v,b}, \hat{T}_{v,b}, \hat{T}_{v,b} \in \{0, 1\} \quad \forall v \in \mathcal{V}_T \\
& H_{v,b}, \bar{H}_{v,b}, \underline{H}_{v,b}, \hat{H}_{v,b}, \hat{H}_{v,b} \in \{0, 1\} \quad \forall v \in \mathcal{V}_H \\
& 0^\circ \leq \theta < 360^\circ.
\end{aligned}$$

The constraints all serve the same purpose as in previous models, so the the only difference is that there is no relationship flipping in terms of the constraints. The value

for $\tan^{-1} \frac{y_v}{x_v}$ can also be pre-computed for each voxel v , so the model must only solve for $\theta, \mathbf{H}, \mathbf{T}$ and the auxiliary variables. It is still a difficult model, due to non-linearity, but much more manageable than model 4.10 from the previous section.

Note that in order to use these pricing models with master problem model 4.7, a general branch and price algorithm is run, with the additional detail that the list of angles Θ' must be tracked at each iteration, and if a new angle is returned from the sub-problem, a corresponding new variable y_θ must be added to the master problem as well. If an added beamlet joins other beamlets, but that same angle already exists in the list, then no new y_θ is added, since multiple beamlets may be active in a single angle. In that regard, this is technically a constraint-and-column generation algorithm.

4.3.4 Heuristic Angle-Generation Approach

An even better option to solving the the simpler pricing problem, is not directly solving the pricing model at all. Because the primary search is for active beamlets, if tractable, computing all intersections to generate the tensor patterns at relevant points in the dataset should be a more effective approach.

The methodology works as follows, given a set of all intersection with horizontal beamlet boundaries taken from equation (4.4) and all vertical beamlet boundaries taken from equation (4.5), all angles at which a change happens in the decision space are known. Calculating all those values will yield a set of critical points $\mathcal{C}$, which represent the boundaries of inclusion of a certain voxel in a certain beamlet, see the left side of Figure 4.12, for an example of what continuous ranges might look like in a toy 3-beamlet example. Each horizontal upper blue line represents an OAR voxel, with the lower pink lines representing CTV. The lines represent at which points a voxel is “seen” at a given angle. On the left, the superimposed vertical lines represent the set of critical points $\mathcal{C}$.

The range of concern is actually not the critical point, but the angle between each

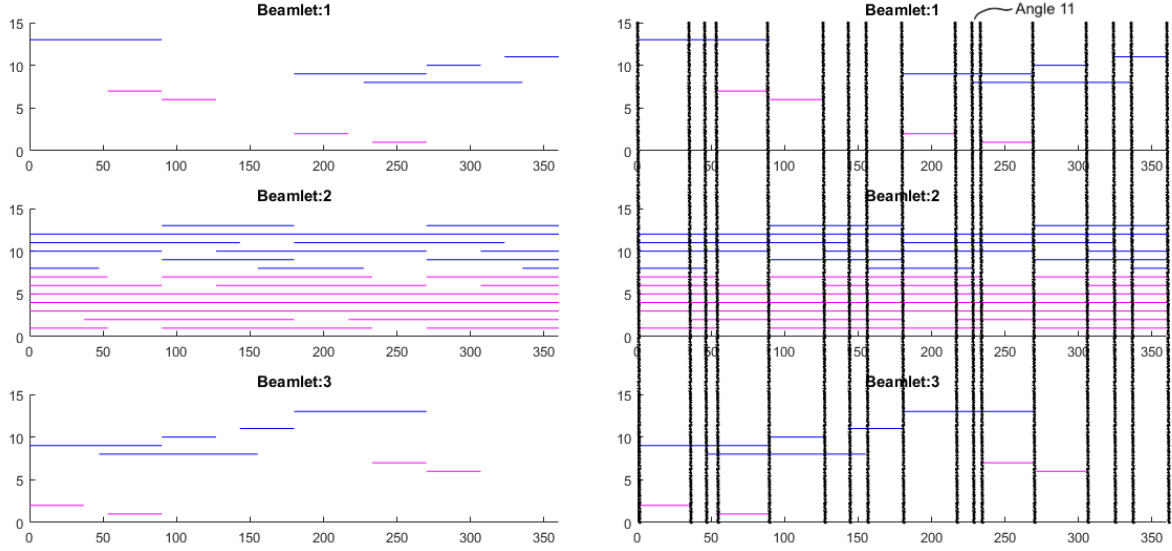


Figure 4.12: Plots depicting the voxels reached by each beamlet at each angle. The blue is OAR voxels and the pink is CTV. The left diagram is the same plot, but with all critical points noted. Here $|\mathcal{C}| = 17$ so $|\Theta| = 16$.

neighbouring critical point. If the exact centre of each two critical points is chosen, exactly $|\mathcal{C}| - 1$ angles will remain. This set Θ containing those mid-points may be large but finite, seeing as each voxel intersects with each beamlet boundary at most twice. Note that these boundaries are defined by the row and column planes, not the beamlets themselves, i.e., sets $\mathcal{Q}$ and $\mathcal{N}$, and there is an additional boundary for each dimension. This means $|\Theta| \leq [2 \cdot (|\mathcal{Q}| + |\mathcal{N}| + 2) \cdot |\mathcal{V}| - 1]$. However, it is not expected that every voxel will interact with every boundary plane. Voxels that are closest to the isocenter should interact with very few planes, therefore the set $|\Theta|$ should remain considerably lower than its upper bound.

Algorithm 7 can be used to calculate all intersections, and then populate the $\mathbf{H}$ and $\mathbf{T}$ values. Note these are even larger, as they are $|\Theta| \cdot |\mathcal{B}| \cdot |\mathcal{V}|$ in size. However, they do not need to be added to an optimization at once, they are simply stored in memory.

Given that Algorithm 7 outputs a lot of data, the full tensors should not be entered

Algorithm 7 Generate all relevant angles and tensors

```

1:  $\mathcal{C} \leftarrow \{\}, \Theta \leftarrow \{\}$ 
2:  $\mathbf{H} \leftarrow \mathbf{0}, \mathbf{T} \leftarrow \mathbf{0}$ 
3: for  $v \in \mathcal{V}$  do ▷ both target and healthy voxels must be included
4:   for  $q \in \mathcal{Q}$  do
5:     if  $r_v^2 - \left(r - \frac{rz_v}{h_q}\right)^2 \geq 0$  then ▷ horizontal intersections
6:       Use Eq. 4.4  $\rightarrow (x_{\pm}^{int}, y_{\pm}^{int}, z_{\pm}^{int})$ 
7:       Eq. 4.6  $\rightarrow \theta_{\pm}$ 
8:        $\mathcal{C} \leftarrow \{\theta_+, \theta_-\}$  ▷ Just  $\theta$  if the equality is strict
9:     for  $n \in \mathcal{N}$  do
10:      if  $\frac{d_n^2 r_v^2}{r^2} - d_n^2 + r_v^2 \geq 0$  then ▷ Vertical intersections
11:        Use Eq. 4.5  $\rightarrow (x_{\pm}^{int}, y_{\pm}^{int}, z_{\pm}^{int})$ 
12:        Eq. 4.6  $\rightarrow \theta_{\pm}$ 
13:         $\mathcal{C} \leftarrow \{\theta_+, \theta_-\}$  ▷ Just  $\theta$  if the equality is strict
14:  $\mathcal{C} \leftarrow \text{sort}(\mathcal{C})$ 
15: for  $i \in 1 \dots |\mathcal{C}| - 1$  do
16:    $\Theta \leftarrow (\mathcal{C}[i] + \mathcal{C}[i + 1])/2$ 
17: for  $\theta \in \Theta$  do
18:   for  $b \in \mathcal{B}$  do ▷ Recall map  $b \rightarrow (n, b)$ 
19:     for  $v \in \mathcal{V}$  do
20:        $\frac{d_{n-1}}{r} r_v \cos\left(\tan^{-1} \frac{y_v}{x_v} + \theta\right) + r_v \sin\left(\tan^{-1} \frac{y_v}{x_v} + \theta\right) \geq d_{n-1}?$   $c_1 = T : c_1 = F$ 
21:        $\frac{d_n}{r} r_v \cos\left(\tan^{-1} \frac{y_v}{x_v} + \theta\right) + r_v \sin\left(\tan^{-1} \frac{y_v}{x_v} + \theta\right) \leq d_n?$   $c_2 = T : c_2 = F$ 
22:        $r_v \cos\left(\tan^{-1} \frac{y_v}{x_v} + \theta\right) \geq r - \frac{rz_v}{h_{q-1}}?$   $c_3 = T : c_3 = F$ 
23:        $r_v \cos\left(\tan^{-1} \frac{y_v}{x_v} + \theta\right) \leq r - \frac{rz_v}{h_q}?$   $c_4 = T : c_4 = F$ 
24:       if  $v \in \mathcal{V}_{\mathcal{T}}$  and  $c_1 \cup c_2 \cup c_3 \cup c_4$  then  $t_{v,b,\theta} = 1$ 
25:       if  $v \in \mathcal{V}_{\mathcal{H}}$  and  $c_1 \cup c_2 \cup c_3 \cup c_4$  then  $h_{v,b,\theta} = 1$ 
26: return  $\Theta, \mathbf{T}, \mathbf{H}$ 

```

into the master problem (4.7), rather it should be populated with a subset, then more columns should be priced in iteratively, as in any CG implementation. This method is

outlined in Algorithm 8, where vectors with promising beamlet-angle combinations are first populated, then the highest performing subset of angles is added using the reduced costs at each iteration.

Algorithm 8 Run the column generation with fixed angles

```

1:  $\Theta' \leftarrow \{\}, \mathcal{B}' \leftarrow \{\}, \mathbf{H}' \leftarrow \{\}, \mathbf{T}' \leftarrow \{\}$ 
2: for  $v \in \mathcal{V}_{\mathcal{T}}$  do ▷ Populate the original master problem
3:   choose the  $b, \theta$  with  $\min(\sum_{v \in \mathcal{V}_{\mathcal{H}}} H_{v,b,\theta})$ 
4:   if not in  $\Theta', \mathcal{B}'$  then
5:      $\Theta' \leftarrow \theta, \mathcal{B}' \leftarrow b$ 
6:      $\mathbf{H}' \leftarrow \mathbf{H}_{b,\theta}, \mathbf{T}' \leftarrow \mathbf{T}_{b,\theta}$ 
7: while True do
8:   Populate master problem (4.7) with  $\Theta', \mathcal{B}', \mathbf{H}'$  and  $\mathbf{T}'$ 
9:   Solve master problem (4.7)
10:  for each remaining  $b, \theta$  combination do
11:     $\bar{c}_{b,\theta} = \sum_{v \in \mathcal{V}_{\mathcal{H}}} H_{v,b,\theta} - \pi_v \cdot T_{v,b,\theta}^T$ 
12:    if  $\bar{c} \geq 0$  then
13:      return  $y_{\theta}, \Theta'$ 
14:      break ▷ At optimality
15:    else
16:       $b, \theta \leftarrow \underset{b,\theta}{\operatorname{argmin}} \bar{c}_{b,\theta}$ 
17:       $\mathbf{H}' \leftarrow \mathbf{H}_{b,\theta}, \mathbf{T}' \leftarrow \mathbf{T}_{b,\theta}$ 

```

A major benefit of calculating the results this way is that the depth of each intersection is provided, meaning that in future work, beam attenuation corrections can be applied to the calculation, rather than only embedding 0-1 tensors.

4.4 Results

The following section shows simple 2D models being solved directly using `intlinprog` in MATLAB. The random instances were generated as follows, in order to resemble real-life problems: random grids of voxels were generated, of different sizes, but as one moves

towards the inside of the grid, the odds of choosing a CTV voxel increases, hitting a maximum in the centre of the frame. The odds of finding an OAR voxel are uniform throughout, with unspecified tissue filling the remainder. This should resemble how clinical CT scans are oriented, as the focal point is the CTV, and all surrounding tissue is incidental.

4.4.1 Solving the Sample Problem

Solving the example model was naturally, very fast; the solver has an optimal value of 3, which was found when only angle 11 was in use (i.e., choose 230.24°). The correctness of this solution can be verified in Figure 4.12, since angle 11 covers all required CTV voxels with only 3 OAR voxels exposed when beamlet 2 is in use.

4.4.2 Solving Larger Sample Problems

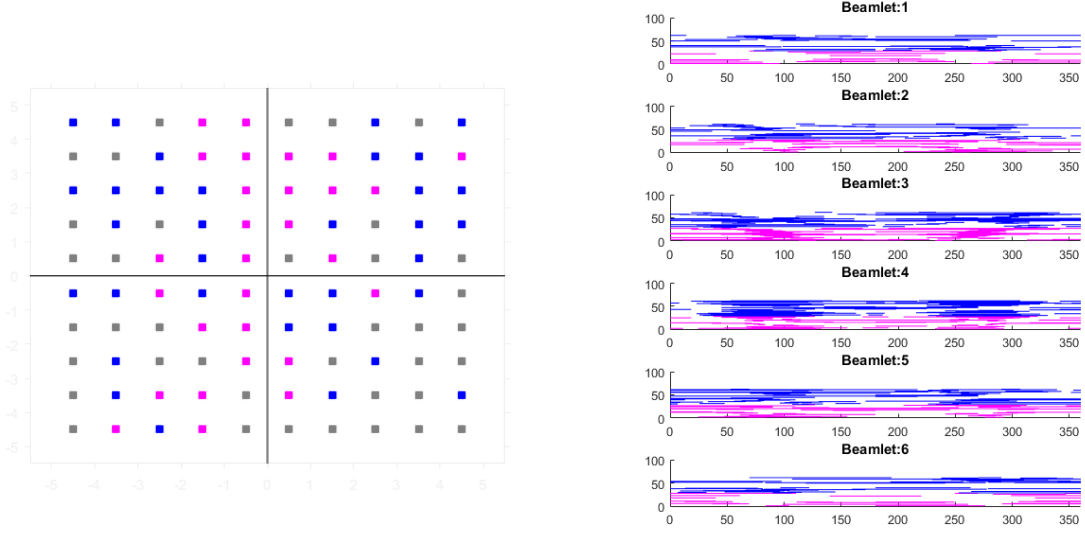
The sample problem was fairly trivial, however, when we get to larger grids, like 10×10 grids, with 1.5mm beamlets, the problems get noticeably harder to solve using just the regular model. In that case, the solver time can vary from a couple seconds to on the order of hours.

Sample Case

For example, the 28 CTV, 34 OAR voxel case in Figure 4.13 ran in 6.5 seconds, with $A = 6$, choosing angles: [3.30, 93.06, 103.03, 166.35, 210.34, 304.60].

Sensitivity to A

Varying A further results in Table 4.1. Some observations from this table are: it is equally valuable to use 3 angles as 8 angles, so 3 should be selected, to minimize setup time during the actual treatment. The model tends to solve much faster for lower A values. The biggest value gain was between 1 and 2 angles.



(a) The randomly generated voxels (b) Plot of voxel activity during beam rotation

Figure 4.13: Output of a random trial with 6-beamlet and 100 (10×10) voxels.

A	Runtime (s)	Angles Selected	Obj. Function
1	0.6	86.94	32
2	0.45	3.30, 97.14	24
3	1.0	3.30, 93.06, 103.03	22
4	8.9	3.30, 93.06, 103.03, 166.35	22
5	17.8	3.30, 93.06, 103.03, 166.35, 210.34	22
6	6.5	3.30, 93.06, 103.03, 166.35, 210.34, 304.60	22
7	0.8	3.30, 47.59, 166.35, 210.34, 255.03, 266.94, 304.60	22
8	8.8	3.30, 47.59, 166.35, 190.00, 210.34, 256.35, 266.94, 304.60	22

Table 4.1: Result changes at various A -values, random trial 1

4.5 Conclusions

This report examined the feasibility of tackling a difficult clinical radiation therapy treatment problem, that is beam angle selection. As a proof of concept, it accomplished this goal by discretizing radiation therapy grids into a fixed set of angles, based on shared

beamlet-voxel interactions. It then introduced a simple set-covering model for limited angle selection, and demonstrated this model on simulated 2D CT slices.

Based on these results, the model seemed to generate insightful information. Consequently, this feasibility study appears to be a promising avenue, worth further exploration. Future work will need to be done to determine the large-scale feasibility of calculating all intersection points. If this is feasible, a heuristic approach for determining beam angles might be the most appropriate way forward in terms of clinical integration, even if a column generation approach would be optimal.

Chapter 5

An Exploration of Network-Based Direct Aperture Optimization

[DAO](#) is a difficult but very desirable problem to solve rapidly, due to its globally optimal properties in terms of [FMO](#) and [LS](#) (it still assumes [BAO](#) input). In this chapter, a completely novel outlook on the [DAO](#) problem is outlined. This methodology differs from past approaches, in that it uses a network setup, not as the typical pricing or decomposition problem, but rather as the basis for an iterative master problem. The proposed method integrates a shortest path setup with a number of side-constraints, constraint-and-column generation sub-problems and heuristics that directly rule out all non-aperture behaviour. This leads to a number of complex edge-cases, but with the hope of faster run times in aggregate, due to the efficient nature of solving network problems using modern solvers like Gurobi ([Gurobi Optimization, LLC, 2024](#)).

In Section [5.1](#), the relevant background on [DAO](#) in the literature is given. In Section [5.2](#), the problem inputs are formalized. In Section [5.3](#) the modelling and heuristics are proposed. In Section [5.4](#) a warm start heuristic that matches the purposed network formulation is described. Finally, in Section [5.5](#), the project status is summarized.

5.1 Introduction

DAO has been addressed a number of times in past literature. The most common recommendation is CG, which is used to avoid directly solving a very large MILP model. The MILP substructure arises to circumvent a non-linear multiplication of continuous intensity variables by binary aperture decision variables. In CG, however, these may be directly split into master and sub-problem decisions, respectively, as depicted in Figure 5.1. The major drawback of CG approaches, however, is that they make it difficult to limit the total number of apertures used in a given treatment plan.

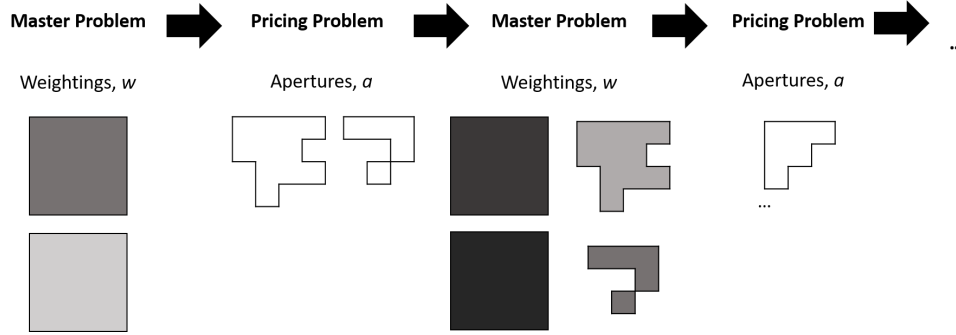


Figure 5.1: Depiction of a column generation problem, like that of Romeijn et al. (2005), where the aperture shapes are chosen in the pricing problem and the master problem determines the dose to each aperture. Note that the starting apertures are two fully open (rectangular) fields.

Controlling the number of apertures is a pressing requirement for two reasons.

1. The software for the linear accelerators typically has an aperture limit for its plans (potentially rendering plans with too many apertures undeliverable).
2. Dose from smaller apertures tends to be unpredictable, leading to poor quality plans.

The second issue stems from the observation that under some common objective conditions, a mathematically optimal, but practically terrible plan would entail delivering the

intensity of each FMO beamlet as its own aperture. Each one of these apertures would lead to a high error rate in dose computation and major quantities of dose-leakage. Additionally, the plan would take hours to deliver, but mathematically if one does not control for the number of apertures, it is a sound plan.

Bearing these drawbacks in mind, the work of Boland et al. (2004) may be considered. In a sense, Boland et al. (2004)'s major contribution is an LS algorithm, requiring the output of an FMO as input. It does, however, contain the foundational method for subsequent CG approaches for DAO, by showing that for a particular set of conditions, one can optimally choose apertures from a fluence map using a clever network reformulation. In terms of aperture control, Boland et al. (2004) use the objective of minimizing beam-on-time as a proxy for the number of apertures in their LS implementation, (assuming plan quality has already been enforced in the FMO). They further show that with careful modelling, this LS problem can be solved in polynomial time. In the scope of their work, they derive shortest path network problems, for beams like the 3×3 example in Figure 5.2.

$i=1$		0	1	2	3	4
$i=2$		0	1	2	3	4
$i=3$		0	1	2	3	4

Figure 5.2: A simple 3×3 beam.

The beam's corresponding graph $G = (E, N)$ is depicted as a set of nodes and edges, where nodes N represent every valid grouping of active beamlets, in terms of left l and right r leaves, for a given row i , and the edges E are valid connections between rows. The network, along with each node's (i, l, r) pairing, is subsequently laid out as depicted in Figure 5.3. Note, to allow for any aperture shape, each node is connected to every node in the row above and below it. However, Boland et al. (2004) show that some deliverability concerns may be accounted for by simply removing certain connecting arcs.

Subsequent papers have used a similar approach to that proposed by Boland et al.

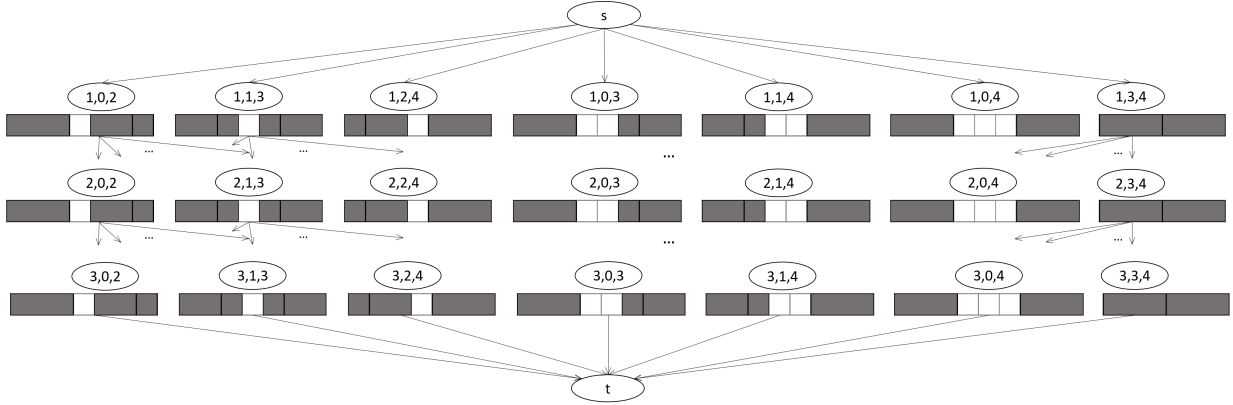


Figure 5.3: A network representation with all the nodes from a 3×3 beam. Note if each row of the graph is fully connected, this setup allows for any possible combination of aperture, based on the path selected between rows.

(2004) to solve the DAO problem directly using CG, yielding impressive results. Romeijn et al. (2005) propose a flexible CG approach with a pricing sub-problem, and further develop the network's arc construction to account for realistic deliverability constraints. The model works well on real world-data, cementing the CG approach as a gold-standard for subsequent DAO problems, especially those in the even-more complex field of VMAT, as in the works of Peng et al. (2012) and Mahnam et al. (2017). The number of apertures in all these methods, however, is largely left as an open problem. Used as both a stopping condition and an optimality trade-off, the authors conclude that there is an exploration factor to choosing the number of apertures at which to halt the CG algorithm. There is also the issue of optimality, as an optimal solution is still not guaranteed if the CG algorithm is halted prematurely.

The authors of Ripsman et al. (2022) defy the CG trend, instead formulating a very flexible model that optimizes a DAO plan for a specific number of apertures (taken as problem input). They use a set of binary variables corresponding to each beamlet and a group of uniformity constraints that enforce the same intensity for all active beamlets in a given aperture. They then go on to impose more aperture-feasibility constraints using

these binary variables. Using this approach, they are able to generate a complete DAO model, however, without capitalizing on the underlying network structure, even with their proposed warm-start heuristic, the plans failed to converge over a 3-day (72 hour) run period.

In this chapter, inspiration was taken from the network minimum path formulations of Boland et al. (2004), Romeijn et al. (2005) and Mahnam et al. (2017), but rather than using a CG-pricing implementation, a novel approach to modelling the DAO problem is proposed. The method specifies the number of apertures using a blend of the network formulations previously integrated into CG methods and some modern tools used in the decision diagram literature Rudich et al. (2023). A novel heuristic warm start is also proposed for the problem.

5.2 Problem Preliminaries

Similar to the previous work of Boland et al. (2004) and Romeijn et al. (2005), we start with a network formulation of the problem, which may also be viewed as a complete decision diagram. Like these authors, we begin with a dummy start, or root and end node, or sink node, as depicted in Figure 5.4. For modelling purposes, it is important to highlight that all angles originate at the source and end at the sink node, which will relate the angles, $\theta_1, \dots, \theta_n$ for aperture limitation purposes.

To stay in line with the works of Romeijn et al. (2005) and Ripsman et al. (2022), we further disallow vertically disconnected apertures, meaning that an aperture begins at the first row where a beam is “on” (i.e., the row has a non-zero number of beamlets exposed) and ends at the first row where the beam is “off” (i.e., has 0 beamlets exposed). This helps create better defined apertures, and is in line with many machine specifications. From an implementation perspective, it means that at a given angle θ , rather than including a shut node in each row, i.e., a node where the right and left nodes meet, we can say that the

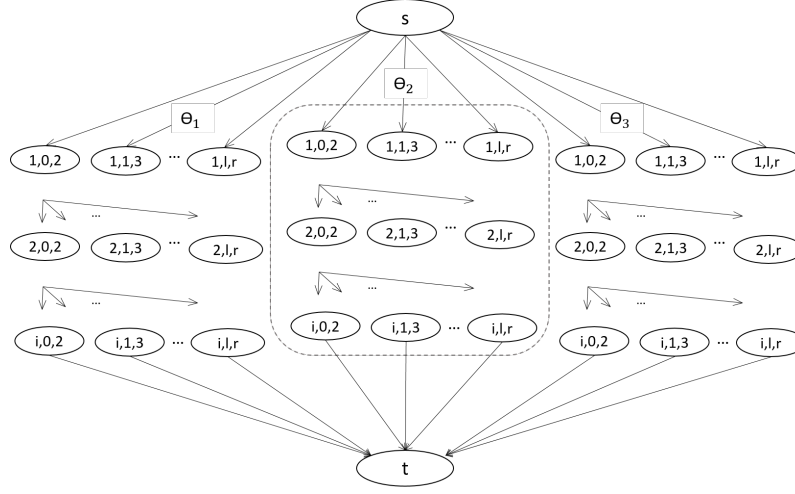


Figure 5.4: A network representation with 3 beam angles, where each angle contains all beam configurations as in Figure 5.3.

first open row is connected to the source and the last open row is connected to the sink. Each new graph angle θ is now represented as depicted in Figure 5.5.

The removed discontinuities simplify the problem, making certain apertures that were previously feasible infeasible, and removing a node from each row. For a beam with m rows and n columns, the added $(m-1)(n)(n+1)$ arcs are offset by the m removed nodes and their respective $2+(m-1)(n)(n+1)$ arcs. Figure 5.6 shows an example of a continuous and a disconnected aperture. Both may be represented using the original Figure 5.3 network, while the diagram in Figure 5.5 only contains a representation of the continuous aperture, as desired.

There is also the phenomenon of horizontal discontinuities, i.e., leaf collisions – when a leaf must pass the opposing leaf vertically adjacent to it (e.g., Figure 5.7b) – or discontinuities – when the leaves do not risk a collision, but the two adjacent parts of open beam are out of contact (e.g., Figure 5.7b). These behaviours are typically undesirable, as they risk overdose to the patient in the form of leakage (the edge of a leaf is often curved and radiation tends to spill over, making tiny apertures risky for the patient). These behaviours

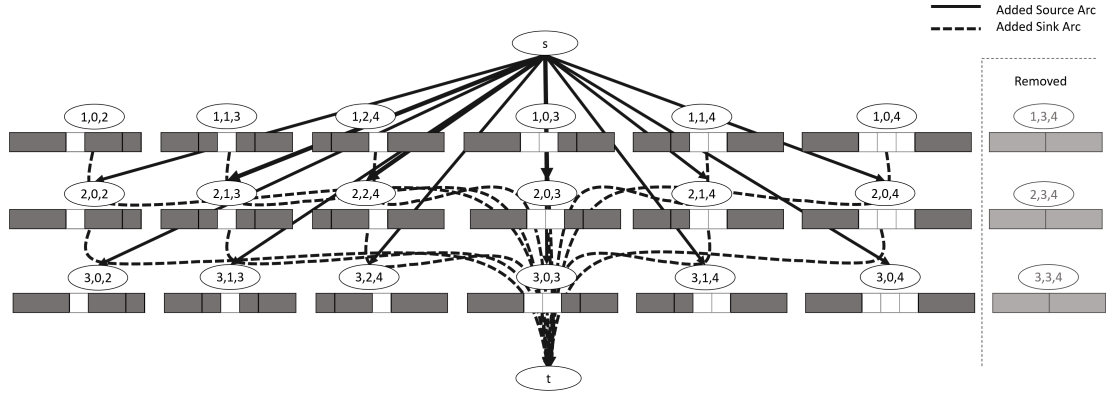


Figure 5.5: Single-angle network without vertical discontinuities. Note, the solid lines show the *added* arcs from the source to every node in every row, except the first (where the connections always exist), and the dotted lines show the added lines from each node, other than the last row (which again, already exist) to the sink node, replacing the nodes and connections on the right.

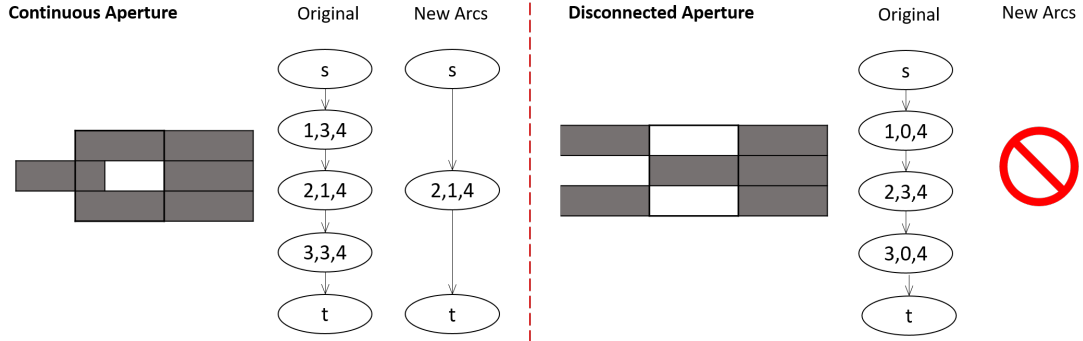


Figure 5.6: Graphical representation of a transformed legal aperture, and rejected disconnected aperture.

are much simpler to rule out in a network than in a typical DAO model seen in Ripsman et al. (2022). Instead of formulating extra constraints with new binary variables, we once again just simplify the model by only assigning edges to nodes in adjacent rows that share at least one beamlet.



Figure 5.7: Undesirable aperture properties due to leaf motion shown in the first two rows of the aperture. In a collision the leaf on the left must pass over a leaf on the right. In a discontinuity, the two parts of the aperture are simply disconnected.

5.3 A minimum-cost flow formulation

In terms of modelling, we note that the state space for each set of row beamlet orientations at a specified angle, $\mathcal{B}_i^\theta$, is associated directly with the node, independent of the incident arc. For ease of exposition, we map the index of a given node, from the angle, as well as the location triplet, to a single index, N , i.e., $[\theta, (i, l, r)] \rightarrow N$, where the corresponding set of beamlets $\mathcal{B}_N$ contains the 2D beamlet location of the beamlets l to r at the correct angle and row in the original matrix. The set of all nodes N (including s and t) is denoted $\mathcal{N}$. In that regard, each node continues to correspond to the $[\theta, (i, l, r)]$ notation, but also encodes the following state information:

- Total dose to all voxels in OAR, $\mathcal{H}$,

$$d_N^{\mathcal{H}} := \sum_{v \in \mathcal{V}_{\mathcal{H}}} \sum_{b \in \mathcal{B}_N} D_{v,b}.$$

- Total dose to each v in target $\mathcal{T}$,

$$d_{v,N}^{\mathcal{T}} := \sum_{b \in \mathcal{B}_N} D_{v,b}.$$

By convention, our intensity variable is denoted w_E , where E is edge (N_1, N_2) and N_1 and N_2 are both in $\mathcal{N}$. Note that although we only need to store state information for the nodes, we need a variable for every edge $E \in \mathcal{E}$. Additionally, we designate $\mathcal{E}_N^{out}$ as the set

of all out-edges of N , and $\mathcal{E}_N^{in}$ as the set of all in-edges of N . We can write our network model formulation as the following minimum-cost flow problem with side constraints,

$$\min_{\mathbf{w} \in \mathbb{R}_+} \sum_{N \in \mathcal{N}} d_N^{\mathcal{H}} \sum_{E \in \mathcal{E}_N^{in}} w_E \quad (5.1a)$$

$$\text{s.t.} \quad \sum_{N \in \mathcal{N}} d_{v,N}^{\mathcal{T}} \sum_{E \in \mathcal{E}_N^{in}} w_E \geq L_v \quad \forall v \in \mathcal{V}_{\mathcal{T}}, \quad (5.1b)$$

$$\sum_{E \in \mathcal{E}_N^{in}} w_E - \sum_{E \in \mathcal{E}_N^{out}} w_E = 0 \quad \forall N \in \mathcal{N} \setminus \{s, t\}, \quad (5.1c)$$

where L_v is once again the prescribed dose to each target voxel $\mathcal{V}_{\mathcal{T}}$. Objective function (5.1a) minimizes the total dose to the OAR, Constraints (5.1b) enforce the minimum dose to each target voxel v , and finally Constraints (5.1c) are flow-conservation constraints, which we need for every node except the source and sink. Note that function (5.1a) could be fairly trivially replaced by any linear (or non-linear, but convex) $f(\mathbf{d}^{\mathcal{H}})$ and constraints (5.1b) by any number of linear functions $g_v(\mathbf{d}^{\mathcal{T}})$, for dose control, etc. The method at hand has been derived for this simpler dose satisfaction case for the clarity of discussing specific features, not due to any inherent limitation of the decision diagram methodology.

Formulation (5.1) is a linearized problem that gives an FMO output. Due to the network structure it should be quite fast to solve, but it does not account for aperture concerns, as there is no requirement to generate finite, feasible apertures. To produce a finite set of apertures, $\mathcal{A}$, with predetermined aperture limit $|\mathcal{A}|$, we use the same approach as Ripsman et al. (2022), by adding sets of binary (or sometimes integer) constraints to the model in the upcoming sections. In Ripsman et al. (2022), however, for every continuous variable, the model requires $3 \times |\Theta| \times |\mathcal{A}| \times m \times n$ additional binary variables to represent the problem, plus auxiliary variables for other delivery concerns, which is a fairly large growth from the relaxed $|\Theta| \times m \times n$ FMO model.

Since the proposed relaxed model has an even larger number of linear decision variables i.e., a fully connected decision diagram leads to $|\Theta| \times (m - 1) \times [(n^2 + n)(\frac{n}{4}(n + 1)) + 1]$

connections, it is natural to think the [DAO](#) version of our problem will explode at an even greater rate than a typical [DAO](#) formulation. However, when implemented as a minimum flow problem, the [DAO](#) version of the model experiences less explosive variable growth for two reasons. First, only one set of integer variables is needed in the decision diagram setup, regardless of the number of apertures requested by the user. Second, the left and right hand variable considerations handled by an extra two sets of binary variables in the [DAO](#) model in [Ripsman et al. \(2022\)](#) are not needed, since the leaf-setups have already been enumerated. For this reason, the full set of binary variables is the exact same size as the number of continuous variables.

That said, the model is quite large, in fact, exponential in the number of columns, so we may further use heuristic and constraint and column generation methods to relax aspects of the model, without losing optimality guarantees. We provide the details of the modelling, as well as the recommended efficiency approaches in this section.

5.3.1 Aperture enforcement and splitting

The issue with the network model above is that the fluence may split many times between s and t . Each split would indicate at least one new aperture. Enforcing a cap on the total number of apertures initially seems as simple as introducing a boolean variable x_E and capping the sum of the variable at the desired total number of apertures $|\mathcal{A}|$ at the first and last rows of the decision diagram.

If only a maximum of $|\mathcal{A}|$ apertures can leave source node, s , and only a maximum of $|\mathcal{A}|$ apertures can enter sink node t , we should have the correct number of apertures running throughout the tree. Given an upper-bound on deliverable dose, M , and a set of rows $i \in \mathcal{I} = \{1, \dots, m\}$, that we assume is the same at each angle (this is done for simplicity, and by common practice; the model could trivially be updated to include the rows of each angles separately, denoted by $\mathcal{I}_\theta$). We can write the additional constraints as

follows. First, relate $\mathbf{w}$ to $\mathbf{x}$,

$$w_E \leq Mx_E \quad \forall E \in \mathcal{E}. \quad (5.2)$$

Next limit the number of apertures by the capacity. Let $\mathcal{E}_i$ be the set of arcs entering a node in row i . Note, this constraint is implemented for all rows, as a stronger cut, but technically is only mathematically necessary at the source and sink,

$$\sum_{E \in \mathcal{E}_i} x_E \leq |\mathcal{A}| \quad \forall i \in \mathcal{I} \cup t. \quad (5.3)$$

Finally, we implement flow-balance constraints, to prevent unlimited splitting,

$$\sum_{E \in \mathcal{E}_N^{in}} x_E - \sum_{E \in \mathcal{E}_N^{out}} x_E = 0 \quad \forall N \in \mathcal{N} \setminus \{s, t\}. \quad (5.4)$$

It seems like together, it should work, as we have one variable that turns on for every allowable aperture. However, the logic of the model breaks down for the first row as follows. When an edge is on, i.e., $x_E = 1$, we allow flow, and we assume that flow belongs to a given aperture. However, $x_E = 1$ does not necessarily mean that only one aperture is using that arc. We can see this splitting issue play out in the following two 3×3 apertures on the single-angle graph in Figure 5.8. Edge $(s, (1, 1, 3))$ is responsible for two apertures, meaning if the above constraints held, $x_{(s, (1, 1, 3))} = 2$. Note that the compliment of this issue is true for the last row of the graph going into sink node, t .

In deeper layers of the graph, e.g., row 2 in Figure 5.9, the flow constraints can be maintained by turning on auxiliary paths that are unused by apertures, so long as the problem has not exceeded the aperture limit. This is the desired behavior for the problem.

There are two potential solutions to this issue, 1) remove flow constraints from s to any node in row $i = 1$ and all flow variables from row $i = m$ to t , 2) maintain flow constraints, but switch these same variables to integer variables. Since the integer change would still allow for the indirect paths to form and would make the problem complexity larger, we

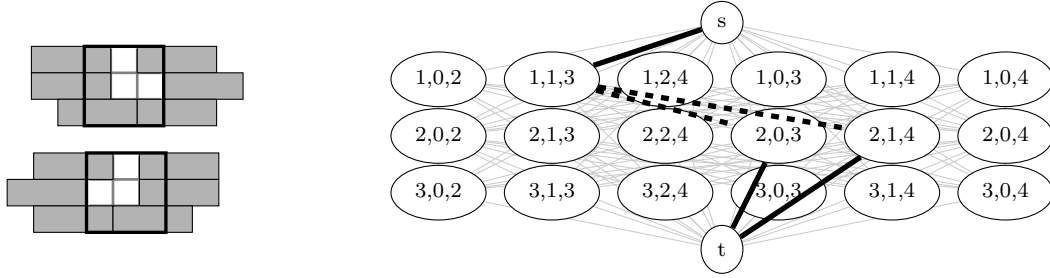


Figure 5.8: An infeasible graph recommendation for the two apertures on the left. Since both apertures share a first node (1,1,3) there is only one input arc. If the graph can split and all binary flow constraints are enforced, it would imply $x_{(s,(1,1,3))} = 2$.

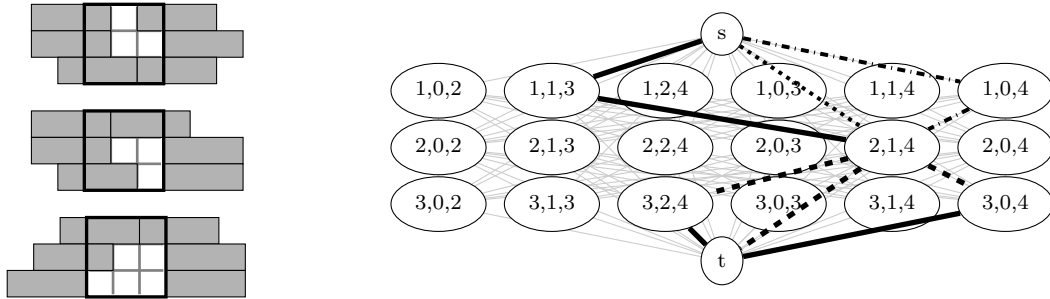


Figure 5.9: All apertures share a node (2,1,4). There is a 3-way+ split at node (2,1,4), which may be accounted for by switching any non-used binaries on, such as the path directly from the start node $s - (2,1,4)$, and the path from $s - (1,0,4) - (2,1,4)$.

start by removing the binary flow constraints on the first and last row, for a smaller feasible space.

This means that instead of constraints (5.4), we want constraints that do not include row $i = 1$ or row $i = m$ of any angle (i.e., the last row of nodes to t). To denote this, we will define $\mathcal{N}_i$ as set of nodes at row i . Note by definition, this does not include s ($i = 0$) or t ($i = m + 1$), because $\mathcal{I}$ represents the rows of the beamlets, not the rows of the graph. The flow constraints are redefined as follows.

$$\sum_{E \in \mathcal{E}_N^{in}} x_E - \sum_{E \in \mathcal{E}_N^{out}} x_E = 0 \quad \forall N \in \mathcal{N}_i, i \in \mathcal{I} \setminus \{1, m\}. \quad (5.5)$$

In order to use constraints (5.5), however, we also need an additional constraint that binds the total flow from the root, i.e., makes sure the sum of all originating arcs is less than or equal to the number of apertures. In that sense, “originating arcs” are defined as any arc that can originate flow (this becomes relevant again in future sections). Since we are giving that ability to all arcs in row 1, any arcs exiting row 1, and any arcs exiting the root s to go anywhere but row 1 must not exceed the total number of apertures. Thus we avoid this removed layer from creating duplicate apertures.

Formalizing these constraints,

$$\sum_{E \in \mathcal{E}_N^{out}, N \in \mathcal{N}_1} x_E + \sum_{E \in \mathcal{E}_s^{out}, N_2 \neq 1} x_E \leq |\mathcal{A}|. \quad (5.6)$$

As well as generating the compliment for the sink node for symmetry (recall our simpler shorthand for arc-row entry $\mathcal{E}_i$),

$$\sum_{E \in \mathcal{E}_m} x_E + \sum_{E \in \mathcal{E}_t, N_1 \neq m} x_E \leq |\mathcal{A}|. \quad (5.7)$$

We can also add a heuristic constraint as row 1 can originate binary flow, but not delete it, so the sum of arcs leaving a node on the first level has to be greater than or equal to

the incoming value,

$$\sum_{E \in \mathcal{E}_N^{out}, N \in \mathcal{N}_1} x_E \geq \sum_{E \in \mathcal{E}_1} x_E. \quad (5.8)$$

And once again, this is mirrored by the row m absorbing arcs, where binary flow can be reduced but not created as follows,

$$\sum_{E \in \mathcal{E}_N^{out}, N \in \mathcal{N}_m} x_E \leq \sum_{E \in \mathcal{E}_m} x_E. \quad (5.9)$$

Together, constraints (5.2), (5.3) and (5.5–5.9) which we will refer to as the all-binary (AB) method, enforces a binary bound on the number of apertures. A remaining issue with this method, however, is that in extreme cases it may rule out certain feasible paths. These paths occur in cases where there are insufficient relief or auxiliary paths to reach a given split node.

To ensure that even the edge cases with too few relief nodes are feasible, we need a set of integer variables z_E to replace the binary variables at edges leaving the source node and edges entering the sink node. We will refer to this method as the integer replacement (IR) method. Unlike the AB approach, this method does not just apply to the first row and last row's connection to the source and sink, respectively, but instead all rows connected to the source and sink, in order to ensure the flow constraints can be completely re-added into the model. In order to achieve this, we will remove all edges that are integer from the original sets of edges $\mathcal{E}$, such that $\hat{\mathcal{E}} = \mathcal{E} \setminus \{\mathcal{E}_s^{out} \cup \mathcal{E}_t^{in}\}$ and assume they are in their own

set $\hat{\mathcal{E}}' = \{\mathcal{E}_s^{out} \cup \mathcal{E}_t^{in}\}$. This gives us $\hat{\mathcal{E}} \cap \hat{\mathcal{E}}' = \emptyset$ and the following constraints,

$$\begin{aligned}
w_E &\leq Mx_E & \forall E \in \hat{\mathcal{E}}, \\
w_E &\leq Mz_E & \forall E \in \hat{\mathcal{E}}', \\
\sum_{E \in \hat{\mathcal{E}}_i} x_E + \sum_{E \in \hat{\mathcal{E}}'_i} z_E &\leq |\mathcal{A}| & \forall i \in \mathcal{I} \cup t, \\
\sum_{E \in \hat{\mathcal{E}}_N^{in}} x_E + \sum_{E \in \hat{\mathcal{E}}_N'^{in'}} z_E - \sum_{E \in \hat{\mathcal{E}}_N^{out}} x_E - \sum_{E \in \hat{\mathcal{E}}_N'^{out'}} z_E &= 0 & \forall N \in \mathcal{N} \setminus \{s, t\}, \\
z_E &\leq |\mathcal{A}| & \forall \hat{\mathcal{E}}', \\
z_E &\in \mathbb{Z}_+ & \forall \hat{\mathcal{E}}'.
\end{aligned} \tag{5.10}$$

Note, we may use the same M value, regardless of the number of apertures added, as this is our total intensity limit for the entire plan, not a per-aperture limitation. Also note that we may use the complete set of flow constraints again, since we would like to encourage the number of paths to be non-binary in the case of splits from the root node, or multi-splits later down in the graph.

There is an argument to be made that for these integer constraints to be necessary, however, the graph would require far fewer connections than apertures, which is not actually realistic for an [RTTP](#) case study. For that reason, we move forward with the (AB) method for the remainder of this chapter. Before moving on, however, we also note that even with all the changes in model (5.10), the formulation does not fully match reality. The concept of decoherence, discussed in Section 5.3.3, is something of the inverse of the splitting issue discussed in this subsection, however, before we get into that concern, it is worth briefly pivoting to implementation of the constraints discussed so far, as the heuristics introduced to this part of the model is a key element in maintaining the overall efficiency of our proposed approach.

5.3.2 Cutting plane binary relaxations

With all the constraints above, we can formulate a partially complete model. Note, regardless of the methodology, it will be helpful to keep the set partition between $\hat{\mathcal{E}}' = \{\mathcal{E}_s^{out} \cup \mathcal{E}_t^{in}\}$ and $\hat{\mathcal{E}}$ containing all other edges. The partial (AB) model appears as follows,

$$\begin{aligned}
& \underset{\mathbf{w} \in \mathbb{R}_+, \mathbf{x} \in \mathbb{Z}_2}{\text{minimize}} && \sum_{N \in \mathcal{N}} d_N^{\mathcal{H}} \sum_{E \in \{\hat{\mathcal{E}}_N^{in} \cup \hat{\mathcal{E}}_N^{in'}\}} w_E \\
& \text{subject to} && \sum_{N \in \mathcal{N}} d_{v,N}^{\mathcal{T}} \sum_{E \in \{\hat{\mathcal{E}}_N^{in} \cup \hat{\mathcal{E}}_N^{in'}\}} w_E \geq L_v \quad \forall v \in \mathcal{V}_{\mathcal{T}}, \\
& && \sum_{E \in \{\hat{\mathcal{E}}_N^{in} \cup \hat{\mathcal{E}}_N^{in'}\}} w_E - \sum_{E \in \{\hat{\mathcal{E}}_N^{out} \cup \hat{\mathcal{E}}_N^{out'}\}} w_E = 0 \quad \forall N \in \mathcal{N} \setminus \{s, t\}, \\
& && w_E \leq M x_E \quad \forall E \in \hat{\mathcal{E}}, \tag{5.11a}
\end{aligned}$$

$$w_E \leq M x_E \quad \forall E \in \hat{\mathcal{E}}', \tag{5.11b}$$

$$\sum_{E \in \hat{\mathcal{E}}_i} x_E + \sum_{E \in \hat{\mathcal{E}}'_i} x_E \leq |\mathcal{A}| \quad \forall i \in \mathcal{I} \cup t, \tag{5.11c}$$

$$\sum_{E \in \hat{\mathcal{E}}_N^{in}} x_E + \sum_{E \in \hat{\mathcal{E}}_N^{in'}} x_E - \sum_{E \in \hat{\mathcal{E}}_N^{out}} x_E - \sum_{E \in \hat{\mathcal{E}}_N^{out'}} x_E = 0 \quad \forall N \in \mathcal{N}_i, i \in \mathcal{I} \setminus \{1, m\}, \tag{5.11d}$$

$$\sum_{E \in \hat{\mathcal{E}}_N^{out}, N \in \mathcal{N}_1} x_E + \sum_{E \in \hat{\mathcal{E}}_N^{out'}, N \in \mathcal{N}_1} x_E + \sum_{E \in \hat{\mathcal{E}}_s^{out'}, N_2 \neq 1} x_E \leq |\mathcal{A}|, \tag{5.11e}$$

$$\sum_{E \in \hat{\mathcal{E}}_m} x_E + \sum_{E \in \hat{\mathcal{E}}'_m} x_E + \sum_{E \in \hat{\mathcal{E}}'_t, N_1 \neq m} x_E \leq |\mathcal{A}|, \tag{5.11f}$$

$$\sum_{E \in \hat{\mathcal{E}}_N^{out}, N \in \mathcal{N}_1} x_E + \sum_{E \in \hat{\mathcal{E}}_N^{out'}, N \in \mathcal{N}_1} x_E \geq \sum_{E \in \hat{\mathcal{E}}_1} x_E + \sum_{E \in \hat{\mathcal{E}}'_1} x_E, \tag{5.11g}$$

$$\sum_{E \in \hat{\mathcal{E}}_N^{out}, N \in \mathcal{N}_m} x_E + \sum_{E \in \hat{\mathcal{E}}_N^{out'}, N \in \mathcal{N}_m} x_E \leq \sum_{E \in \hat{\mathcal{E}}_m} x_E + \sum_{E \in \hat{\mathcal{E}}'_m} x_E. \tag{5.11h}$$

Constraints (5.11a–5.11h) represent the partitioned versions of the binary constraints previously introduced – into groups with and without source and sink connection. Constraints (5.11e) and (5.11f) are almost entirely partitioned by definition, the only outliers being the edges going straight from row 1 to the sink and straight to row m from the source.

The main take away from this model is there is a fairly large amount of binary variables. Since network models do tend to choose full paths, and have obvious splitting points, however, a lot of the binary variables are likely unnecessary at early stages of the modelling. To take advantage of this property, we can relax a number of $\mathbf{x}$ variables (in $\hat{\mathcal{E}}$) throughout the model. Arguably, if all edges in $\hat{\mathcal{E}}'$ are binary, the only time the model will create non-integral apertures, is when the model benefits from fractional solutions, and we can rule those out iteratively. By introducing linearized flow variable $\tilde{x}_E \in \mathbb{R}_+$, we get the following relaxed versions of constraints (5.11a),

$$\begin{aligned} w_E &\leq M\tilde{x}_E \quad \forall E \in \hat{\mathcal{E}}, \\ 0 &\leq \tilde{x}_E \leq 1 \quad \forall E \in \hat{\mathcal{E}}, \end{aligned} \tag{5.12}$$

The linearized variable is also included in constraints (5.11c–5.11h),

$$\begin{aligned} \sum_{E \in \hat{\mathcal{E}}_i} \tilde{x}_E + \sum_{E \in \hat{\mathcal{E}}'_i} x_E &\leq |\mathcal{A}| \quad \forall i \in \mathcal{I} \cup t, \\ \sum_{E \in \hat{\mathcal{E}}_N^{in}} \tilde{x}_E + \sum_{E \in \hat{\mathcal{E}}_N^{in'}} x_E - \sum_{E \in \hat{\mathcal{E}}_N^{out}} \tilde{x}_E - \sum_{E \in \hat{\mathcal{E}}_N^{out'}} x_E &= 0 \quad \forall N \in \mathcal{N}_i, i \in \mathcal{I} \setminus \{1, m\}, \\ \sum_{E \in \hat{\mathcal{E}}_N^{out}, N \in \mathcal{N}_1} \tilde{x}_E + \sum_{E \in \hat{\mathcal{E}}_N^{out'}, N \in \mathcal{N}_1} x_E + \sum_{E \in \hat{\mathcal{E}}_s^{out'}, N_2 \neq 1} x_E &\leq |\mathcal{A}|, \\ \sum_{E \in \hat{\mathcal{E}}_m} \tilde{x}_E + \sum_{E \in \hat{\mathcal{E}}'_m} x_E + \sum_{E \in \hat{\mathcal{E}}'_t, N_1 \neq m} x_E &\leq |\mathcal{A}|, \\ \sum_{E \in \hat{\mathcal{E}}_N^{out}, N \in \mathcal{N}_1} \tilde{x}_E + \sum_{E \in \hat{\mathcal{E}}_N^{out'}, N \in \mathcal{N}_1} x_E &\geq \sum_{E \in \hat{\mathcal{E}}_1} \tilde{x}_E + \sum_{E \in \hat{\mathcal{E}}'_1} x_E, \\ \sum_{E \in \hat{\mathcal{E}}_N^{out}, N \in \mathcal{N}_m} \tilde{x}_E + \sum_{E \in \hat{\mathcal{E}}_N^{out'}, N \in \mathcal{N}_m} x_E &\leq \sum_{E \in \hat{\mathcal{E}}_m} \tilde{x}_E + \sum_{E \in \hat{\mathcal{E}}'_m} x_E. \end{aligned}$$

The next step is a constraint-and-column generation approach. For any violating edge E that has flow ($\tilde{x}_E > 0$ and $w_E > 0$) but $\tilde{x}_E < 1$, i.e., the relaxed variable is non-integer,

we must generate a new binary variable x_E , and add the constraint,

$$\tilde{x}_E = x_E,$$

forcing the original value to a 0 or 1. This must be repeated until all positive flow variables are associated with a binary flow value. The set of all variables with binary counterparts will be referred to as $\mathcal{X}$. Algorithm 9 summarizes this process. Through these steps, the model should be prevented from creating additional apertures by forming new paths.

5.3.3 Problems with decoherence and merging

The model may no longer explicitly create new apertures when enforcing the methods from Sections 5.3.1 and 5.3.2, but it may do so implicitly by merging existing apertures in an undesirable manner. For example, Figure 5.10 shows how shared nodes can lead to fluence swapping, which would implicitly create new apertures that would go undetected in the model.

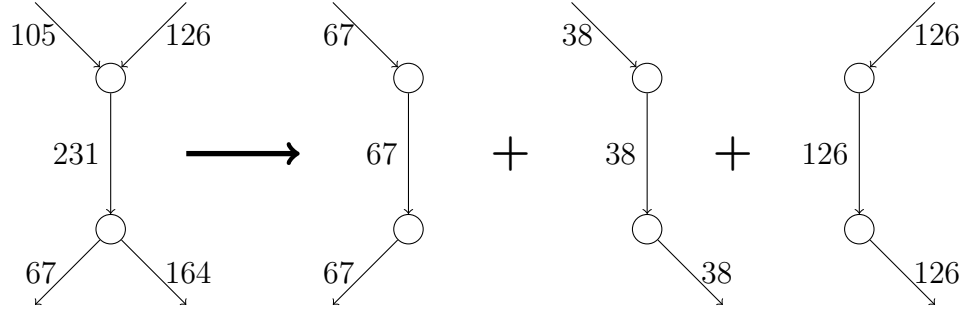


Figure 5.10: A decoherence example. To the model, this looks like two apertures that briefly overlap for two nodes. However, due to the fluence change, a minimum of three apertures is needed.

In this case, what initially appeared to be 2 apertures can be decomposed into 3 apertures, potentially exceeding the aperture capacity limit $|\mathcal{A}|$. Moreover, this allocation may only represent a locally optimal solution, as later down in the graph, a different aperture

Algorithm 9 Fractional flow removal

```

1: input:  $\mathbf{d}^{\mathcal{H}}, \mathbf{d}^{\mathcal{T}}, \mathbf{L}, \mathcal{N}, \mathcal{E}, |\mathcal{A}|$ 
2:  $\mathcal{X} \leftarrow \{x_E \mid \forall E \in \hat{\mathcal{E}}\}$ 
3: model  $\leftarrow$  relaxed model 5.11 with inputs  $(\mathbf{d}^{\mathcal{H}}, \mathbf{d}^{\mathcal{T}}, \mathbf{L}, \mathcal{N}, \mathcal{E})$ 
4: IsValid  $\leftarrow$  FALSE
5: while !IsValid do
6:    $\mathbf{w}, \mathbf{x}, \tilde{\mathbf{x}} \leftarrow \text{optimize}(\text{model})$ 
7:   for row  $i$  in  $\{\mathcal{I} \cup m + 1\}$  do
8:     for every edge  $E \in \mathcal{E}_i$  do ▷ Check the aperture limits
9:       totalEdges  $\leftarrow$  0
10:       $\mathcal{X}_i \leftarrow \emptyset$ 
11:      if  $w_E > 0$  then
12:        totalEdges  $\leftarrow$  totalEdges+1
13:         $\mathcal{X}_i \leftarrow \mathcal{X}_i \cup x_E$ 
14:      if totalEdges  $> |\mathcal{A}|$  then
15:        IsValid = FALSE
16:        for  $x_E \in \mathcal{X}_i \setminus \mathcal{X}$  do
17:          if  $x_E < 1$  then ▷ Add violating binaries to model
18:            Add variable  $x_E$  to model
19:            Add constraints  $\tilde{x}_E = x_E$  to model
20:           $\mathcal{X} \leftarrow \mathcal{X} \cup \mathcal{X}_i$ 
21:        if  $(i > 1)$  AND  $(i < m - 1)$  then ▷ Check flow constraints
22:          for  $N \in \mathcal{N}_i$  do flowIn  $\leftarrow$  0 flowOut  $\leftarrow$  0
23:            for  $E \in \mathcal{E}_{N_i}^{in}$  do
24:              if  $w_E > 0$  then
25:                flowIn  $\leftarrow$  flowIn+1
26:            for  $E \in \mathcal{E}_{N_i}^{out}$  do
27:              if  $w_E > 0$  then
28:                flowOut  $\leftarrow$  flowOut+1
29:            if flowIn  $\neq$  flowOut then
30:              IsValid = FALSE
31:              for  $x_E \in \mathcal{E}_{N_i}^{in} \cup \mathcal{E}_{N_i}^{out}$  do
32:                if  $x_E < 1$  then ▷ Add violating binaries to model
33:                  Add variable  $x_E$  to model
34:                  Add constraints  $\tilde{x}_E = x_E$  to model

```

allocation could potentially be beneficial (e.g, maybe splitting 126 into 67 and 59 matches a later merge-and-split, so splitting the 105 vs. the 126 necessitates more apertures later on). An example of this pattern is also illustrated on a 4×3 beamlet grid graph in Figure 5.11, for reference.

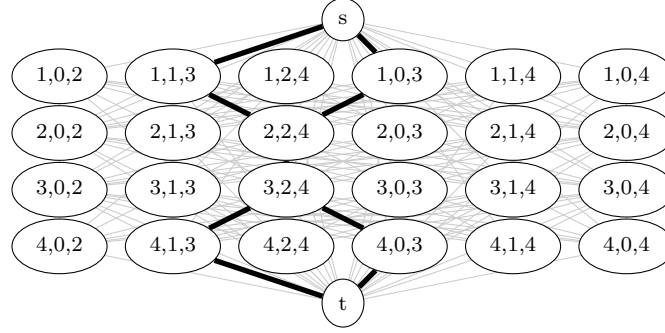


Figure 5.11: A representation of the decoherence shown in Figure 5.10 on a 4×3 graph.

When attempted directly, decoupling decoherence becomes a difficult take on the (already NP-hard) partition problem, wherein all connected apertures would need to be enumerated, and the minimum-cardinality split would be chosen. The outcome of such an algorithm will likely contain more decoupled paths than allowable apertures, which would mean it is not a feasible solution to the original problem. Rather than searching for a potentially sub-optimal solution, we instead employ heuristic methods to detect and then rule out decoherence iteratively. The output of these heuristics can be inserted back into the model to enforce the desired behaviour upon further iterations of the solver.

5.3.4 Decoherence detection heuristic

An early observation when it comes to decoherence is that it necessarily occurs after a flow-merge is followed by a flow-split. This is because if there is no additional fluence entering a given path (i.e., if there is no merge), there will be nothing to decouple further downstream in the graph, and splits are handled by the binary constraints. If there is a

merge but no split, the path simply follows to the sink.

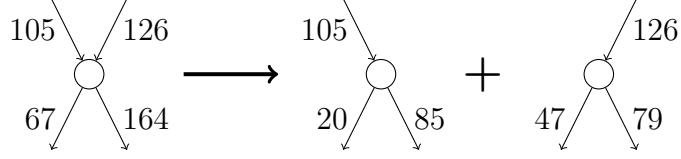


Figure 5.12: Decoherence example but over a single node. Note that with splitting, this could represent up to four apertures as depicted, but the model would only see two.

In light of this property, one way to detect a merge-then-split is by detecting 1) multiple non-matching inputs and outputs over a single node, as depicted in Figure 5.12 or 2) an increase in an otherwise monotonically decreasing set of flows. The first case is directly clear and detectable, whereas the second is formalized in Theorem 1.

Theorem 1. *Let $\mathcal{P}$ be a path that contains a set of nodes $\{s, N_1, \dots, N_{|\mathcal{P}|-1}, t\}$ and edge set $\mathcal{E}_p := E_q \in \mathcal{P} | E = (\mathcal{P}_q, \mathcal{P}_{q+1}) \forall q \in \{1, \dots, |\mathcal{P}| - 1\}$. There is decoherence on path $\mathcal{P}$, given the values of its ordered set $\{w_E | E \in \mathcal{E}_P\}$ if $\exists w_{E_x} > w_{E_{x-1}}$ and $\exists w_{E_y} < w_{E_x}$, such that index $y > x$.*

Proof. Suppose there exists an edge in $E_x \in P$ such that its flow value w_{E_x} is greater than the preceding value $w_{E_{x-1}}$, but there is no merge, i.e., no further input, then flow balance constraint (5.1c) has been violated. Therefore, $E_x(2)$ must be a merge node. Again, suppose by contradiction a value $\exists w_{E_y} < w_{E_x}$ where $y > x$, but there was no split, again constraint (5.1c) would be violated because flow has disappeared. A split node $E_x(1)$ has been identified. $\square$

Based on this theorem and the properties above, we propose Algorithm 10, for detecting merge and split nodes in the network. The goal of this algorithm is to return a set of $\leq |\mathcal{A}|$ apertures that each contain a specified flow, meaning it would be run only on potentially optimal plans, i.e., plans where Algorithm 9 was satisfied that returned no cuts.

Algorithm 10 Decoherence detection

```

1: input: feasible solution to relaxed model,  $\mathbf{w}$  and  $\mathcal{E}$ .
2: Merge  $\leftarrow$  False,  $E_{merge} \leftarrow \{\}$ ,  $a = 1$ 
3: while  $\exists w_E > 0, E \in \mathcal{E}_s^{out}$  do ▷ While there are still active apertures
4:   Choose active edge  $E \in \mathcal{E}_s^{out}$  ▷ Can use any method, e.g., always left-most
5:    $f_a \leftarrow w_E$ ,  $f_a^{last} \leftarrow w_E$  with  $E = (s, N_2)$ ,  $\mathcal{P}_a \leftarrow \{s\}$ 
6:    $\mathcal{P}_a \leftarrow \mathcal{P}_a \cup \{N_2\}$ 
7:   while  $N_2 \neq t$  do
8:      $N_1 \leftarrow N_2$ 
9:     Choose active edge  $E \in \mathcal{E}_{N_1}^{out}$  ▷ e.g., left-most
10:    if  $w_E \leq f_a^{last}$  then
11:      if  $|\{w_E > 0 | E \in \mathcal{E}_{N_1}^{in}\}| > 1$  then ▷ Node has multiple incoming edges
12:        Merge  $\leftarrow$  True ▷ Follow merge protocol
13:      if !Merge then ▷ Only splits
14:         $f_a \leftarrow w_E$ , with  $E = (N_1, N_2)$  ▷ Nothing has come in - must be a split
15:         $\mathcal{P}_a \leftarrow \mathcal{P}_a \cup \{N_2\}$ ,  $f_a^{last} \leftarrow w_E$ ,
16:      else
17:        if  $\exists w_E = f_a, E \in \mathcal{E}_{N_2}^{out}$  then ▷ Only continue if there's matching outflow
18:           $\mathcal{P}_a \leftarrow \mathcal{P}_a \cup \{N_2\}$ ,
19:          set  $E \in \mathcal{E}_{N_2}^{out}$  to contain the equal edge only.
20:           $f_a^{last} \leftarrow f_a$ 
21:          Merge  $\leftarrow$  False ▷ Reset. Any merges did not cause decoherence.
22:        else
23:          return  $E_{merge}, \mathcal{P}_a$  ▷ Stop the algorithm, decoherence found
24:      else
25:        Merge  $\leftarrow$  True,  $E_{merge} \leftarrow (N_1, N_2)$ 
26:         $\mathcal{P}_a \leftarrow \mathcal{P}_a \cup \{N_2\}$ ,  $f_a^{last} \leftarrow w_E$ 
27:        Subtract  $f_a$  from all  $w_E \in P_i$ 
28:         $a = a + 1$ 
29: return  $\mathcal{P}_a, f_a \forall a \in |\mathcal{A}|$ 

```

We further propose in Proposition 1 that if no merge is detected in Algorithm 10, the solution $\mathbf{w}$ is optimal, and sets $\mathcal{P}_a$ contain all of the optimal apertures and fluences f_a , $\forall a \in \mathcal{A}$.

Proposition 1. *If Algorithm 10 $\rightarrow \mathcal{P}_a, f_a$, then $\mathbf{w}^* = \mathbf{w}$ and $|\mathcal{P}| = |\mathcal{A}|$.*

Proof. Given that Algorithm 10 returns the appearance of the first merge edge E_{merge} , the return of $\mathcal{P}_a, f_a \forall a \in |\mathcal{A}| \implies$ no merge edge exists. If $\mathbf{w}$, which is optimal to model (5.11) and satisfies Algorithm 9, then in the absence of merges, the binaries must indicate the the model contains the correct number of apertures. Therefore, it must also be optimal to the entire problem. $\square$

Note that on top of separating apertures, Algorithm 10 does try to fix decohering apertures in lines 17–21. It does so by not breaking out of the algorithm after a merge if the fluence from before the merge exactly matches one of the fluence values exiting the downstream split. This would mean that the aperture is already implicitly decoupled at this point, so it is assigned to the path and the algorithm continues.

5.3.5 Super paths for decoherence prevention

Given that a path containing decoherence has been found, it must be ruled out in the main model. One way to do this is by iteratively choosing auxiliary variables that represent super paths, found during the merge-detection process in Algorithm 10. Starting from the merge-edge E_{merge} and continuing along the returned violating path $\mathcal{P}_a$, these super paths essentially provide a direct connection between the start and ending of decoherence. By isolating this path from the solution space, we theorized that we would be able to determine exactly what fluence belongs to that path, or disallow the use of the path entirely if it is sub-optimal without decoherence. The method takes inspiration from both column and row generation, in that it introduces new variables and constraints simultaneously.

The method was conceptualized as follows; starting with the incoming edge to a detected merge, create a set of edges that form a path from the incident node to the first merge node, N_1 , to the outgoing node from the split with an unmatched value N_2 . Once

a decohering path has been isolated, add a new path $p = (N_1, N_2)$ to an edge set, along with a new couple of variables x_p and w_p to represent the new edge.

The main difference between these path variables and the main variables is the cost that gets assigned to N_2 due to this path is different for our new edge. The route from N_1 to N_2 is a path $p \in \mathcal{P}$, with the associated nodes $\mathcal{N}_p$ and edges $\mathcal{E}_p$ on the way. It has the associated constants,

$$d_p^{\mathcal{H}} = \sum_{N \in \mathcal{N}_p} d_N^{\mathcal{H}} \quad (5.13)$$

and for all v in target $\mathcal{T}$,

$$d_{v,p}^{\mathcal{T}} = \sum_{N \in \mathcal{N}_p} d_{v,N}^{\mathcal{T}}. \quad (5.14)$$

Which must be added to the objective function,

$$\sum_{N \in \mathcal{N}} d_N^{\mathcal{H}} \sum_{E \in \mathcal{E}_N^{\text{in}}} w_E + \sum_{p \in \mathcal{P}} d_p^{\mathcal{H}} w_p, \quad (5.15)$$

and minimum dose constraints, $\forall v \in \mathcal{V}_{\mathcal{T}}$,

$$\sum_{N \in \mathcal{N}} d_{v,N}^{\mathcal{T}} \sum_{E \in \mathcal{E}_N^{\text{in}}} w_E + \sum_{p \in \mathcal{P}} d_{v,p}^{\mathcal{T}} w_p \geq L_v. \quad (5.16)$$

Additionally, we need a constraint that prevents this exact path from being used without activating the new variables, so, we need an additional constraint,

$$\sum_{E \in \mathcal{E}_p} x_E \leq |\mathcal{E}_p| - 1, \quad (5.17)$$

and all binaries should be enforced on this path, i.e., $\forall E \in \mathcal{E}_p$,

$$\tilde{x}_E = x_E.$$

Finally, the new edge should be added to all $\mathcal{E}$ sets, as a direct link between N_1 and N_2 . Computationally, this requires an update in the remaining constraints in the model, as it is added to the sets, but formulation-wise, there is no change.

There is an issue with this approach however, as how does one prevent multiple legal paths from being ruled out by super path exclusions?

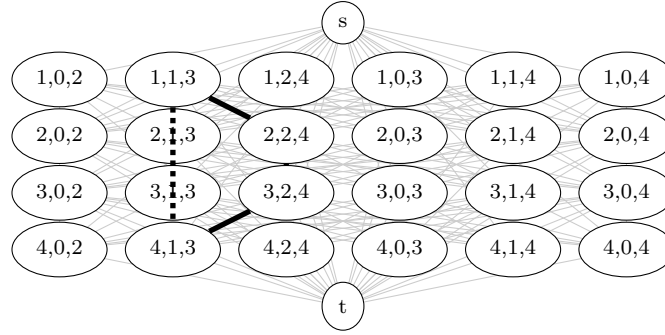


Figure 5.13: The super path is the leftmost path, which now directly connects the previously 3-edge path from (1,1,3) to (4,1,3). The original path is shown with the dark solid lines, while the super path is the dotted line.

For example, given the decoherence pattern in Figure 5.13, with a constraint that rules out the path, it would also erroneously rule out the legal set of paths shown in Figure 5.14. This is because we are forcing the sum of the 3 arcs on the path to be ≤ 2 , as required by equation (5.17).

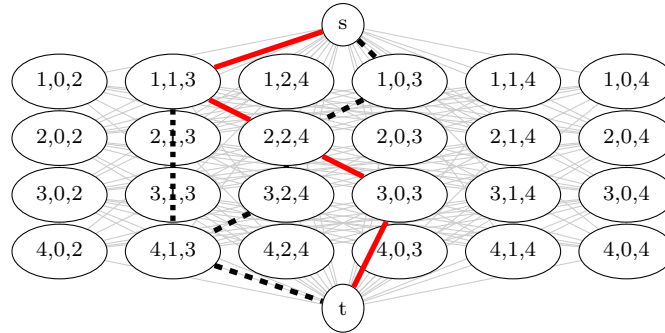


Figure 5.14: A set of legal apertures that violate the generated super path sum constraint.

To rectify this situation, it becomes clear that constraint (5.17) is not sufficient for this application. Instead, one edge found on the violating path must be removed altogether. However, in removing that edge, even more paths become disallowed, and will have to be separately generated. So by removing an entry edge into the merge, $(1,1,3)-(2,2,4)$, every legal path using that that edge, must be replaced. E.g., $(1,1,3)-(2,2,4)-(3,0,3)$ would have to become a direct path, as would $(1,1,3)-(2,2,4)-(3,2,4)-(4,2,4)$. Every permutation of path that started along the removed path, followed the super path, but has a different terminal node, would need to be enumerated. In this case this is $(1,1,3)$ to all outgoing nodes from $(2,2,4)$ and $(1,1,3)-(2,2,4)$ to all outgoing nodes from $(3,2,4)$ except $(4,1,3)$, as depicted in Figure 5.15.

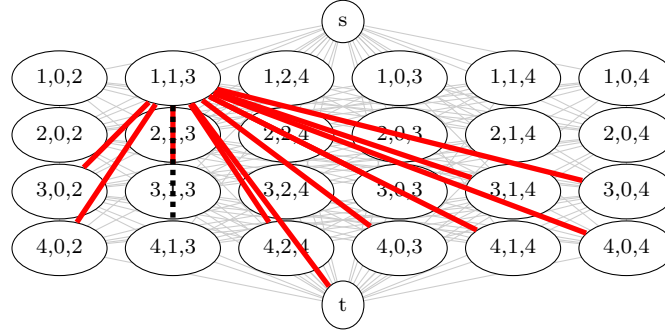


Figure 5.15: The set of 12 paths required to replace $(1,1,3)-(2,2,4)$ and $(1,1,3)-(2,2,4)-(3,2,4)$. Note, two paths are represented by the same path $(1,1,3)-(t)$; 1) $(1,1,3)-(2,2,4)-(t)$ and 2) $(1,1,3)-(2,2,4)-(3,2,4)-(t)$.

This must be done for all $E \in \mathcal{E}_p \setminus \mathcal{E}_{|P|}$, i.e., all edges in the super path except the final edge, since the complete path should use the new edge. The new objective and constrain coefficients would need to be calculated for these arcs as well (equations (5.13) and (5.16)), and integrated into the objective function and constraints, as in equations (5.15) and (5.14). However, equation (5.17) is no longer necessary, and the variables do not need to be enforced as binaries, unless they are found to violate Algorithm 9.

Note, this method is quite cumbersome to implement and track, seeing as the original paths must be stored in memory for each new node, and multiple super paths could be

formed between the same two nodes, with differing intermediaries (e.g., if path $(1,1,3)-(2,0,2)-(3,0,2)-(4,1,3)$ had decoherence in a later iteration), generating identical paths with different d_p^H and $d_{v,p}^T$ values, which may both be needed in the optimal solution. For that reason a second approach was then considered, called node-splitting.

5.3.6 Node-splitting for decoherence prevention

A second method to prevent decoherence is a node-splitting approach, which prevents flow from merging in a similar, but perhaps more intuitive manner than super paths. Once again, we use the familiar decoherence example, redrawn in Figure 5.16a. In this method, the node with the detected troublesome merge point in Algorithm 10 is separated into the original $N = E_{merge}(2)$ and a new dummy node, N' . The arc E_{merge} now only points to N' , and is removed from $\mathcal{E}_N^{in}$, and is the only in edge in $\mathcal{E}_{N'}^{in}$. The outgoing edges of N , $\mathcal{E}_N^{out}$ (and remaining in edges) remain identical.

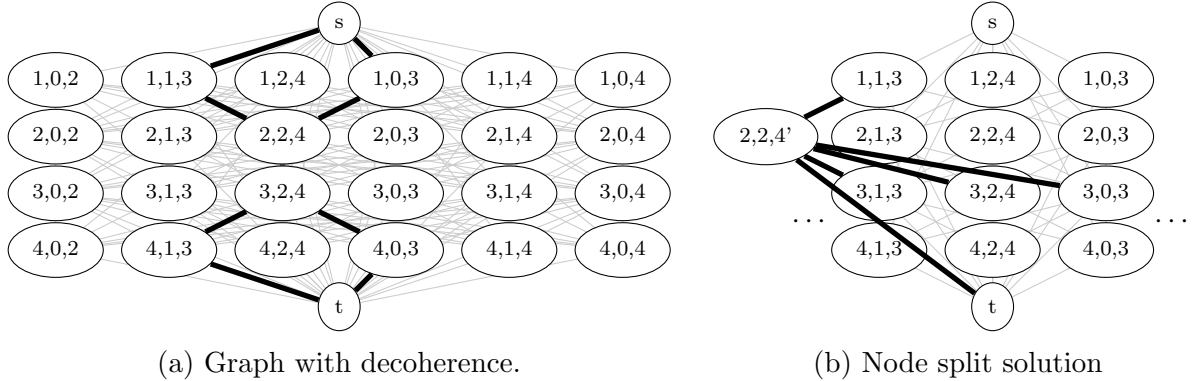


Figure 5.16: Given the decoherence in Figure (a), Figure (b) shows the resolution by splitting the merge node $(2,2,4)$ into two nodes $(2,2,4')$ and $(2,2,4)$. Edge $(1,1,3)-(2,2,4)$ is removed from the problem, whereas $(1,1,3)-(2,2,4')$ and all outgoing arcs from $(2,2,4')$ are added to the problem.

All outgoing arcs of N' are duplicates of $\mathcal{E}_N^{out}$, to ensure that the only difference between the updated model and the old model, the new node and its associated new variables. All

non-decohering paths remain identical.

Conceptually, this approach is much more straight-forward than the previous approach, however it is clear that a second iteration will be needed to actually change the values that get assigned to not decohere (see Figure 5.17a). So we will actually want to make a dummy node for every node on the violating path, as discussed in the previous section, or the path will necessarily form an identical split at another node in the path.

The node split method ends up working almost identically to the path-split approach, wherein the entire set of nodes along a violating path must be removed from the graph, and then re-attached everywhere but in the space with the violation. However, notation-wise and implementation wise, this is a far superior way to implement the problem, seeing as the node costs and formulation basics all remain the same, but with extra nodes and changed edges.

The variables change as follows in this approach; a new w_E and $\tilde{x}_E$ is required for each new edge in $\mathcal{E}_{N'}^{out}$, and the flow constraints and objectives must be updated to reflect the new sets, but no new constants must be constructed as in equations (5.13) and (5.14).

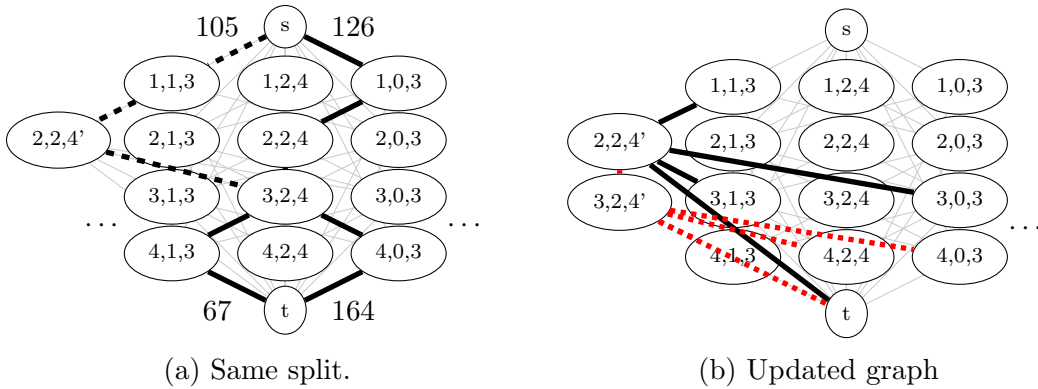


Figure 5.17: If the entire violating path is not ruled out, the algorithm will simply decohere further up a path. Figure (b) allows path (s)–(1,0,3)–(2,2,4)–(3,2,4)–(4,0,3)–(t) to exist, without allowing the 105 fluence from the left side to meet it via (1,1,3) at (2,2,4) or (3,2,4), due to the novel (2,2,4') and (3,2,4') nodes and edges.

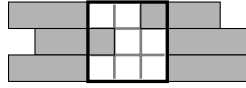
5.4 Heuristic Warm Start Approach

The past sections introduced a sketch of how a path-based approach could be used to generate feasible plans for [RTTP](#). Again, taking inspiration from [Ripsman et al. \(2022\)](#), it is natural to assume that starting the model with a strong feasible solution should improve solver time. In [Ripsman et al. \(2022\)](#), the authors implement their warm start using a combination of bounding the model close to the relaxed [FMO](#) objective value, while minimizing the largest intensity in a given aperture. The idea is that this method will force active beamlets to have uniform fluence values in a given aperture. The final stage of their algorithm must fill in the gaps of the aperture between leaves (filling vertical and horizontal gaps, if need be, as well) and force every active beamlet to its maximum value in a given aperture.

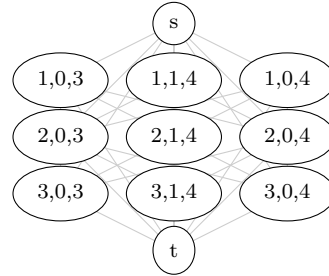
Once again, given the nature of our decision diagram method, a more accurate warm-start is not only possible, but not hard to implement. Rather than working with a completely different set of models, we can simply take relaxed model [\(5.11\)](#) and feed it a smaller version of the graph, that still represents a valid solution.

In our case, we took one less than the minimum number of columns in any angle ($n - 1$, for the purpose of this paper) and made that the smallest allowable gap of the nodes we generate. In the 3×3 example, that would mean at least a 2-beamlet gap must be visible for the row to be turned on, as depicted in [Figure 5.18](#). Only the nodes that represent rows with at least 2 open beamlets are generated, reducing variables and model complexity significantly, but given a strong warm start, that respects the initial binary constraints.

Note that if the warm start model is potentially being considered for a fast solution alternative, [Algorithms 9 and 10](#) should be run until the solution is optimal. If it is being used directly as a warm start to speed up a larger model, there is no reason to use them, as the next iteration will likely choose a different solution, making any added dummy nodes and binaries unnecessary. Moreover, if nodes and binaries are not added, the model will



(a) Allowable apertures.



(b) Warm start graph.

Figure 5.18: Warm start example with a 3×3 beam. Note the leaves depict the allowable apertures at any row (or the row may be turned off). Smallest allowable active aperture is now 2-beamlets long, instead of 1.

simply undo this progress.

5.5 Conclusions

The proposed approach is a promising direction for solving large-scale [DAO](#) problems. Due to the graphical structure, a strong, relevant warm-start and lack of immediate need for binary variables, we hypothesize that the model should run very fast, and have seen that to be the case in small-scale trials so far. The complexity of implementing the algorithms for the first time however, means the project is still underway, with the intention of completing it and open-sourcing the code in the near future, so nobody else has to go through the arduous implementation process. The code is actively being developed in a combination of Julia, JuMP and Gurobi ([Lubin et al., 2023](#); [Gurobi Optimization, LLC, 2024](#)).

The outline given in this thesis, however, should provide a complete road-map for anybody interested in implementing the code faster, or using a different set of software and solvers, or finding inspiration for similar finite [CG](#) problems.

Chapter 6

A Framework for Comparing Voxel Down-Sampling Approaches

When a patient receives CT imaging before a radiation therapy treatment, the clinicians receive very high-granularity images in order to contour the organs. Since the steps of the RTTP process, e.g., formulating the FMO model, are then typically run on a per-voxel-basis, the number of voxels under consideration directly impacts problem size and complexity. Typically, clinical planning software uses lower granularity than the CT images, as shown in Figure 6.1.

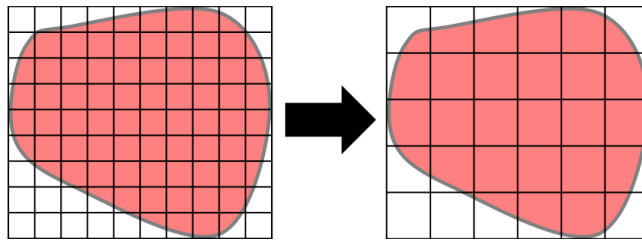


Figure 6.1: Original CT voxels in an abstract, 2D organ of interest $9 \times 10 = 90$ voxels, might translate to a less granular $5 \times 6 = 30$ voxel grid in a planning system.

For research purposes, this effect is often replicated by only considering every n^{th} voxel,

as in Figure 6.2. The issue with this process, which will be referred to as integer down-sampling, is that it is unclear if it is the best or most representative sampling approach for a given patient structure. As such, the goal of this chapter is to introduce common voxel space-pruning methodologies, and to provide a framework for comparing them with the integer approach as well as with each other, in order to establish best sampling practices.

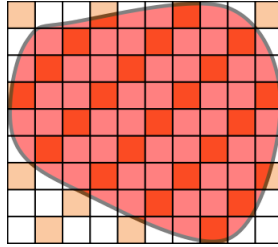


Figure 6.2: Integer- n sampling (in this case $n = 3$). The chosen voxels are the darker/orange voxels, for a total of 30 voxels.

The contributions of this chapter include:

1. A review of and some novel updates to popular sampling techniques in the literature
2. A novel geometric method for sampling, not previously applied to radiation therapy
3. A framework for comparing both new and existing methodologies.

Within the chapter, background literature is surveyed in Section 6.1, followed by a summary of the existing techniques under consideration in Section 6.2. In Section 6.3, the geometric approach, referred to as the layered approach is proposed. In Section 6.4, a framework for comparing these methodologies is proposed. Section 6.5 contains the preliminary results from a single-patient pilot test and the chapter is concluded in Section 6.6.

6.1 Background Literature

The largest bottleneck in quickly and efficiently solving [RTTP](#) problems, across all stages of the process tends to be problem size ([Rocha et al., 2011](#)). This scaling issue is largely what separates [RTTP](#) from simpler problems of the same structure. The problems are so large in fact, that some form of problem sampling is used effectively across the board in the field. Papers such as [Dempsey et al. \(2005\)](#), provide a cautionary message that the [FMO](#) model in particular will take advantage of sampled blind spots and meet the worst-case information loss, due to the opportunistic nature of optimization. For that reason, naive methods, such as the integer sampling approach discussed above, and stochastic methods, such as the method shown in [Sjölund et al. \(2019\)](#), should be used with caution.

The review paper by [Rocha et al. \(2011\)](#) compares two types of sampling common to [RTTP](#): uniform sampling and non-uniform sampling. Here, benefits of non-uniform sampling across critical structures were found over traditional uniform sampling. In the non-uniform approach, more critical organ tissue was un-sampled while less relevant surrounding tissue was aggressively down-sampled, leading to a notable reduction in inter-organ effects. It should be noted that because any sampling approach can further be applied non-uniformly, this will largely be left out of the scope of this particular exploration.

Another form of sampling relies on clustering approaches to simplify the problem space. [Scherrer et al. \(2005\)](#) use a method called adaptive hierarchical clustering, which breaks the decision space geometrically, based on dose-influence similarity. At each iteration, a variant of the [FMO](#) model is solved, and then the differences in the resulting full and hierarchical doses are used to motivate more granular sub-clusters in high-error regions. While this method works more rapidly than running a traditional [FMO](#), setting up the initial hierarchy is non-trivial, which may add undue complexity when merging complex steps, such as [FMO](#) and [LS](#). This problem is worsened when the model is an [MILP](#) like a [DAO](#) problem, where multiple iterations make an already prohibitively-large problem even more difficult to solve. For these reasons, and for the sake of fair comparisons, only the

single-pass portion of algorithms will be considered in this study.

At its core, the [Scherrer et al. \(2005\)](#) single-pass clustering component appears to work similarly to a deterministic k -means method. Their approach clusters voxels based on dose similarity, however, it is unclear exactly what metric is used to describe similarity at each phase of the clustering process, making this implementation difficult to exactly replicate.

[Mahnam et al. \(2017\)](#) propose a single-pass voxel-clustering algorithm based on the k -means method, with a bias towards nearest-neighbour clustering. This method requires a fixed number of clusters chosen *a priori*, which might not respect the true geometry of the target region. The authors in [Martin et al. \(2007\)](#) also propose a single-pass sampling approach, but with randomized sampling built into their steepest descent algorithm, choosing different voxels at every iteration. The performance is found to be fast, however, it must use constraint-free optimization techniques, which rules out a lot of modern applications and provides a noisy convergence, with unclear stopping criteria.

[Fountain et al. \(2022\)](#) propose an iterative sampling method based on the beam of radiation, which can be described as follows in a single-pass setting. They cluster voxels based on their most dominant beamlet. Then, they add the “most affected” [CTV](#) voxel from that beamlet to the set of considered voxels. This method intends to democratize beamlet sampling, in that no beamlet will be disregarded in the treatment. It may be detrimental in terms of coverage, however, as the most affected voxel does not necessarily need the most attention from the solver. Focusing on the most affected voxels may in fact lead to under-dosing less affected voxels, especially in a single-pass setup.

[Tiwari \(2015\)](#) propose a sampling scheme most similar to the layered approach proposed in this chapter, called Boundary and Interior sampling (BIS). In it, the author takes all boundary voxels, as well as a k -means sampling of interior voxels into consideration (which are sub-sampled quasi-randomly) when running [FMO](#) models. This method has similar issues to other k -means clustering and randomized methods, without providing as much of a speed-up as comparable methods.

In the proposed layered approach, the boundary-based logic used to motivate the BIS algorithm in the first place, is continued throughout the remainder of the structure, which should make it appropriate for [BAO](#) as well as [FMO](#) and [DAO](#), and take away the reliance on random sampling. The layered method allows for greater control of the sampling process, which should make it easier to design reliable procedures going forward.

This chapter provides the framework for comparing these methods over large classes of patients. The goal is to ultimately recommend a single best approach, that may lead to more accurate planning in future studies. The present study only demonstrates the approaches on a single patient study, setting the groundwork for future, larger-scale efforts.

6.2 Sampling Techniques

This chapter considers four sampling approaches from the literature, with additional implementation recommendations for several of the approaches. Any associated constraint generation, or non-uniform sampling algorithms proposed alongside the technique by the literature are filtered from the procedures, as any sampling method could be integrated with these subsequent heuristics. The four algorithms described in this section are:

1. Integer- n sampling (Section [6.2.1](#))
2. k -means sampling (Section [6.2.2](#))
3. k -neighbour sampling (Section [6.2.3](#))
4. Beamlet-based sampling (Section [6.2.4](#)).

While all four algorithms are general, only the [CTV](#) is down-sampled for this study, since each [OAR](#) behaves in aggregate in the considered ([FMO](#)) model. Some sampling algorithms result in a pruning of voxels, while others result in a complete transformation

of the $D_{v,b}$ matrix. In both cases, the adjusted $D_{v,b}^{alg}$ matrices and $\mathcal{V}_{\mathcal{T}}^{alg}$ sets are then used to populate the constraints in the (FMO) model, which is adapted as follows,

$$\begin{aligned} \min_{\mathbf{w} \in \mathbb{R}_+^{|\mathcal{B}|}} \frac{1}{|\mathcal{V}_{\mathcal{H}}|} \sum_{v \in \mathcal{V}_{\mathcal{H}}} \sum_{b \in \mathcal{B}} D_{v,b} w_b + \frac{1}{|\mathcal{V}_{\mathcal{T}}^{alg}|} \sum_{v \in \mathcal{V}_{\mathcal{T}}^{alg}} \sum_{b \in \mathcal{B}} D_{v,b}^{alg} w_b \\ \text{s.t. } \sum_{b \in \mathcal{B}} D_{v,b}^{alg} w_b \geq L_v \quad \forall v \in \mathcal{V}_{\mathcal{T}}^{alg}. \end{aligned} \quad (6.1)$$

Note that none of the transformations studied will impact the set of beamlets $\mathcal{B}$, so the number of columns in $D_{v,b}$ is unaltered in $D_{v,b}^{alg}$.

6.2.1 Integer Down-Sampling

As already depicted in Figure 6.2, the integer method simply considers every n^{th} voxel from the original set of voxels $\mathcal{V}$, to get $\mathcal{V}^{int-n} \subset \mathcal{V}$. The pseudocode for populating $\mathcal{V}^{int-n}$ and $D_{v,b}^{int-n}$ is given in Algorithm 11.

Algorithm 11 Integer down-sampling pseudocode

```

1: input  $\mathcal{V}, \mathbf{D}, n$ 
2:  $\mathcal{V}^{int-n} \leftarrow \{\}, \mathbf{D}^{int-n} \leftarrow \{\}$ 
3: for  $v \leftarrow 1 : n : |\mathcal{V}|$  do
4:    $\mathcal{V}^{int-n} \leftarrow v$ 
5:    $\mathbf{D}^{int-n} \leftarrow \mathbf{D}[v, :]$ 
6: return  $\mathcal{V}^{int-n}, \mathbf{D}^{int-n}$ 

```

6.2.2 k -means Sampling

The k -means clustering algorithm is fairly standard in the literature; the number of clusters, k is given as input, and initially k random voxels are chosen as centroids. After the k -centroids are assigned, any distance metric, but in this case, the euclidean distance of each

voxel from each centroid is measured. A given voxel is then assigned to its closest centroid by distance. The mean of the entire set of voxels is recomputed as the new centroid. This process is repeated until the centroids stay the same between iterations.

Since each voxel has a set of physical location coordinates (x_v, y_v, z_v) , the k -means algorithm may cluster voxels based on these distances. The voxels also each have an associated row of the dose-influence matrix $D_{v,b}$, which may be used to evaluate similarity as well. Both a coordinate-based, and dose-influence matrix-based version of the algorithm are described in Algorithm 12. In both cases, the $\mathcal{V}^{km-c}$ and $D_{v,b}^{km-c}$ or $\mathcal{V}^{km-D}$ and $D_{v,b}^{km-D}$ returned are dummy voxels with representative centroids.

Algorithm 12 k -means sampling pseudocode

```

1: input  $\mathcal{V}, \mathbf{D}, K$ , type
2:  $\mathcal{V}^{km\text{-type}} \leftarrow \{1, \dots, K\}$ 
3: for  $k \leftarrow \{1, \dots, K\}$  do
4:    $\mathcal{C}'_k \leftarrow \{\}$  ▷ for convergence check
5:    $v \leftarrow \text{random}(\mathcal{V})$ 
6:    $\mathcal{C}_k \leftarrow v$  ▷ initialize centroid list with  $k$  random voxels
7:    $D_k^{km\text{-type}} \leftarrow D_v$  ▷ for both algs
8:   if type= $c$  do  $(x_k^c, y_k^c, z_k^c) \leftarrow (x_v, y_v, z_v)$ 
9:   while  $\mathcal{C}_k \neq \mathcal{C}'_k$  do
10:     $\mathcal{C}'_k \leftarrow \mathcal{C}_k$ 
11:    for  $v \leftarrow \{1, \dots, |\mathcal{V}|\}$  do
12:      for  $k \leftarrow \{1, \dots, K\}$  do
13:        if type= $c$  do  $d_{v,k} = \|(x_v, y_v, z_v) - (x_k^c, y_k^c, z_k^c)\|_2^2$ 
14:        if type= $D$  do  $d_{v,k} = \|D_v - D_k^{km\text{-type}}\|_2^2$ 
15:         $k^* \leftarrow \underset{k}{\text{argmin}} d_{v,k}$ 
16:         $\mathcal{C}_{k^*} \leftarrow v$ 
17:      for  $k \leftarrow \{1, \dots, K\}$  do
18:        if type= $c$  do  $(x_k^c, y_k^c, z_k^c) \leftarrow \text{mean}((x_v, y_v, z_v) \mid \forall v \in \mathcal{C}_k)$ 
19:         $D_k^{km\text{-type}} \leftarrow \text{mean}(D_v \mid \forall v \in \mathcal{C}_k)$  ▷ needed for both cases
20: return  $\mathcal{V}^{km\text{-type}}, \mathbf{D}^{km\text{-type}}, \mathcal{C}_k$ 

```

6.2.3 k -Neighbour Sampling

The k -neighbour approach is implemented based on [Mahn timer et al. \(2017\)](#), and it is nearly identical to the $km - D$ approach above, except that the authors begin by randomly assigning all voxels to clusters, instead of assigning them to the nearest cluster. They also make fewer checks for each voxel per iteration. They accomplish this by looking only at the centroids, defined by a row k of the D matrix, i.e., D_v^{kn} , that have been assigned to the voxel itself as well as its nearest neighbours, instead of comparing the distance between every voxel and every centroid. The method is summarized in Algorithm 13.

Algorithm 13 k -neighbour sampling pseudocode

```

1: input  $\mathcal{V}, \mathbf{D}, K$ ,
2:  $\mathcal{V} \leftarrow \{1, \dots, K\}$ 
3:  $\mathcal{C}'_k \leftarrow \{\}$   $\forall k \in \{1, \dots, K\}$  ▷ for convergence check
4:  $\mathcal{C}_k \leftarrow$  randomly assign all  $v \in \mathcal{V}$  ▷ assign all voxels to clusters randomly
5: while  $\mathcal{C}_k \neq \mathcal{C}'_k$  do
6:    $\mathcal{C}'_k \leftarrow \mathcal{C}_k$ 
7:   for  $k \leftarrow \{1, \dots, K\}$  do
8:      $D_k^{kn} \leftarrow \text{mean}(D_v \ \forall v \in \mathcal{C}_k)$ 
9:     for  $v \leftarrow \{1, \dots, |\mathcal{V}|\}$  do
10:      for  $k \leftarrow \text{nearest\_neighbour\_lookup}(v)$  do ▷ only centroids of neighbours
11:         $d_{v,k} = \|D_v - D_k^{kn}\|_2^2$ 
12:         $k^* \leftarrow \underset{k}{\text{argmin}} \ d_{v,k}$ 
13:         $\mathcal{C}_{k^*} \leftarrow v$ 
14: return  $\mathcal{V}^{kn}, \mathbf{D}^{kn}, \mathcal{C}_k$ 

```

6.2.4 Beamlet-Based Sampling

The final literature approach is implemented based on [Fountain et al. \(2022\)](#), and it uses the $D_{v,b}$ matrix exclusively to inform dose-selection. In the original paper, it was proposed that the voxels should first be clustered by beamlet, then each cluster's representative

voxel is the most influential (highest $D_{v,b}$ value) voxel in that cluster. In this chapter, that method was reproduced, however, since representative sampling is paramount, an approach that chose the median voxels within the cluster was also implemented for the trials. These approaches are denoted $bb - max$ and $bb - med$ in Algorithm 14, respectively.

Algorithm 14 Beamlet-based sampling pseudocode

```

1: input  $\mathcal{V}, \mathbf{D}, \text{type}$ 
2:  $\mathcal{V}^{bb\text{-type}} \leftarrow \{\}, \mathbf{D}^{bb\text{-type}} \leftarrow \{\}$ 
3:  $\mathcal{L}_b \leftarrow \{\}$ 
4: for  $v \leftarrow \{1, \dots, |\mathcal{V}|\}$  do
5:    $b^* \leftarrow \underset{b \in \mathcal{B}}{\operatorname{argmax}} \{D[v, :]\}$ 
6:    $\mathcal{C}_{b^*} \leftarrow v$ 
7: for  $b \leftarrow \{1, \dots, |\mathcal{B}|\}$  do
8:   if  $\text{type} = max$  then
9:      $v \leftarrow \max \mathcal{C}_b$ 
10:  elseif  $\text{type} = med$  then
11:     $v \leftarrow \text{med } \mathcal{C}_b$ 
12:   $\mathcal{V}^{bb\text{-type}} \leftarrow v$ 
13:   $\mathbf{D}^{bb\text{-type}} \leftarrow \mathbf{D}[v, :]$ 
14: return  $\mathcal{V}^{bb\text{-type}}, \mathbf{D}^{bb\text{-type}}$ 

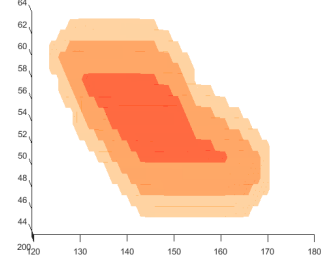
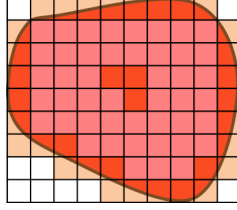
```

6.3 Proposed Layered Approach to Sampling

The layered approach to sampling takes the 3D structure of the target organ into consideration. Rather than sampling randomly throughout, it considers the path of an external beam hitting it. Several facts are invariably true:

- The beam must enter through a boundary and exit through a boundary.
- Outer voxels must be reached before inner voxels.

These two facts lead to the hypothesis, that rather than sampling randomly, it makes sense to take more complete representations of layers of structure, with less loss of information, as in Figure 6.3.



(a) 35 voxels selected, via every 3rd layer sampling (b) 3 layers of voxels on an actual 3D CTV

Figure 6.3: Layered approach to sampling. Chosen voxels are the darker/orange voxels.

Like the integer approach, this method takes an every n perspective on voxel sampling, however, it chooses every voxel in the n^{th} , rather than just some of the voxels, in order to maintain the shape of the object at hand. The approach is summarized in Algorithm 15.

Algorithm 15 Layered sampling pseudocode

```

1: input  $\mathcal{V}, \mathbf{D}, n$ 
2:  $\mathcal{V}^{lay-n} \leftarrow \{\}, \mathbf{D}^{lay-n} \leftarrow \{\}$ 
3:  $l \leftarrow 1$ 
4: while  $|\mathcal{V}| \neq \emptyset$  do
5:    $\mathcal{C}_l \leftarrow \text{peel\_outer}(v \in \mathcal{V})$  ▷ check for voxels with no neighbours
6:    $\mathcal{V} \leftarrow \mathcal{V} \setminus \mathcal{C}_l$  ▷ remove outer voxels
7:    $l \leftarrow l + 1$ 
8: for  $l \leftarrow 1 : n : |\mathcal{C}|$  do ▷ populate problem with every  $n^{th}$  layer
9:    $\mathcal{V}^{lay-n} \leftarrow \mathcal{C}_l$ 
10:   $\mathbf{D}^{int-n} \leftarrow \mathbf{D}[\mathcal{C}_l, :]$ 
11: return  $\mathcal{V}^{lay-n}, \mathbf{D}^{lay-n}$ 

```

6.4 Voxel Normalization Strategies

The goal of this work is to compare sampling approaches in a fair manner. However, it is clear that initially some of the five methods (seven, with variants) have very different clustering and selection criteria, and will therefore not be an apples-to-apples comparison. The following section provides guidance on getting all methods on an equal footing.

Ideally, each method is run with the same or similar numbers of voxels, regardless of whether these voxels are stand-in centroids, or voxels from the original structure. The nearest-neighbour and k -means algorithms all take in k as input, which makes their inclusions simpler (although due to the stochastic nature of the methods, it is not possible to enforce that all k clusters are used in each heuristic every time). For the remaining 3 classes of approaches, the following schemas are recommended for normalizing the plans.

Integer sampling: This method will be the ground truth for the sampling approaches. It dictates that the number of clusters will be $\frac{|\mathcal{V}|}{n}$. Therefore to be on equal footing, the three k -means-based algorithms must have $k \approx \frac{|\mathcal{V}|}{n}$, or more precisely, $k = \left\lfloor \frac{|\mathcal{V}|}{n} \right\rfloor$ voxels.

The layered approach: If only one voxel were to be considered using layered sampling, it would be the centre voxel of the structure, by volume. Assuming a more generous sampling scheme, a sphere packing-style problem is employed, where iteratively, the largest layer allowable (by number of voxels $|\mathcal{C}_l|$) is “packed” into the sample set, up to k total voxels. This method ensures that voxels may be greedily added in a rapid fashion, with a potential trade-off in terms of overall optimality and uniform dispersion of layers.

The beamlet-based approach: For this approach, one can simply continue to iterate through the highest voxel values from each cluster, and then next highest, as needed for the *bb-max* case, or take more voxels from the middle of the cluster in the *bb-med* case, until exactly k voxels is reached.

Taken together, these three approaches allow the different styles of sampling to be tested under similar conditions, better controlling for the implementation variations.

6.5 Results

The patient under consideration is a breast cancer patient with 36,196 relevant CT voxels. Of those voxels, 33,592 are in the CTV, whereas 2,604 are OAR. The problem was implemented using C++ and CPLEX 22.1.1, which was interfaced with MATLAB, in order to access the problem data. The model was run on a local 8 GB Windows computer. The following section is broken down to show visualizations of patient down-sampling approaches in Section 6.5.1 and plan results in Section 6.5.2.

6.5.1 Patient Sampling Visualizations

The examined breast cancer patient has a relatively small relevant OAR region due to an intentionally stylized tangential beam setup that excludes many OAR voxels from seeing the beam (Purdie et al., 2014). A visualization of the relevant volumes are shown in Figure 6.4. The prescribed dose treatment dose was 42.4 Gy to all CTV voxels and there are 1520 beamlets (760 from each of the two tangential angles).

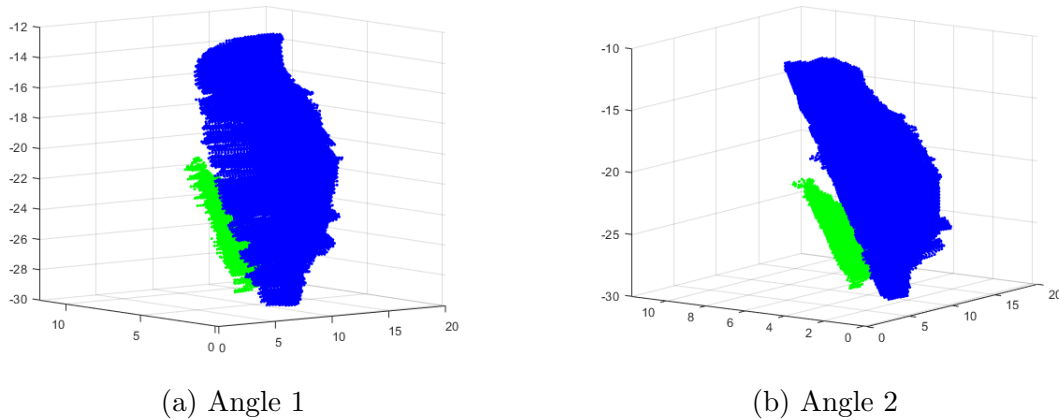


Figure 6.4: Two slightly different rotations of the same patient image. The light green is heart OAR that can be reached by the selected beams, while the dark blue is breast CTV

Note that the CT depicted in Figure 6.4 is a single phase of a 4D image, meaning the voxels have been deformed across multiple images for tracking purposes, and therefore do not have a strict grid-structure. This is called deformable registration and is noteworthy as it demonstrates that all algorithms may be implemented in a manner that is flexible enough to account for non-grid-based scans.

Each plan was run at $n = 2$, $n = 10$ and $n = 100$ sampling rates, to showcase the full effect of all approaches. It should be noted that real-world planning granularity for this case would be roughly equivalent to an $n = 32$ sampling rate, meaning true sampling likely falls somewhere between the middle and most aggressive rates. When the sampling is run on the integer plan, the series of sampling results looks as shown in Figure 6.5.

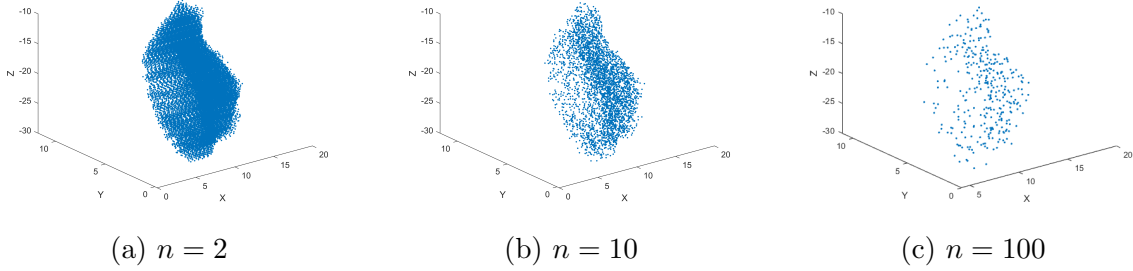


Figure 6.5: Integer down-sampling at an n of 2, 10 and 100.

The remaining six types (and subtypes) of sampling are all depicted at the mid-value $n = 10$ granularity in Figure 6.6. There are some notable properties from these images; foremost, the beamlet-based algorithms ($bb - max$ and $bb - med$) are hollow on the inside. $bb - max$ has a larger hollow region than $bb - med$, as the median-based approach shifts the selected voxels towards the center of the beamlet of interest, however in both cases, none of the sampling reaches the deepest portions of the volume.

The k -means based approaches ran much slower than the other alternatives, particularly the dose-influence-based implementations. The $km - D$ approach was the most time consuming, and as such, was given a heuristic speedup. At every iteration, the model

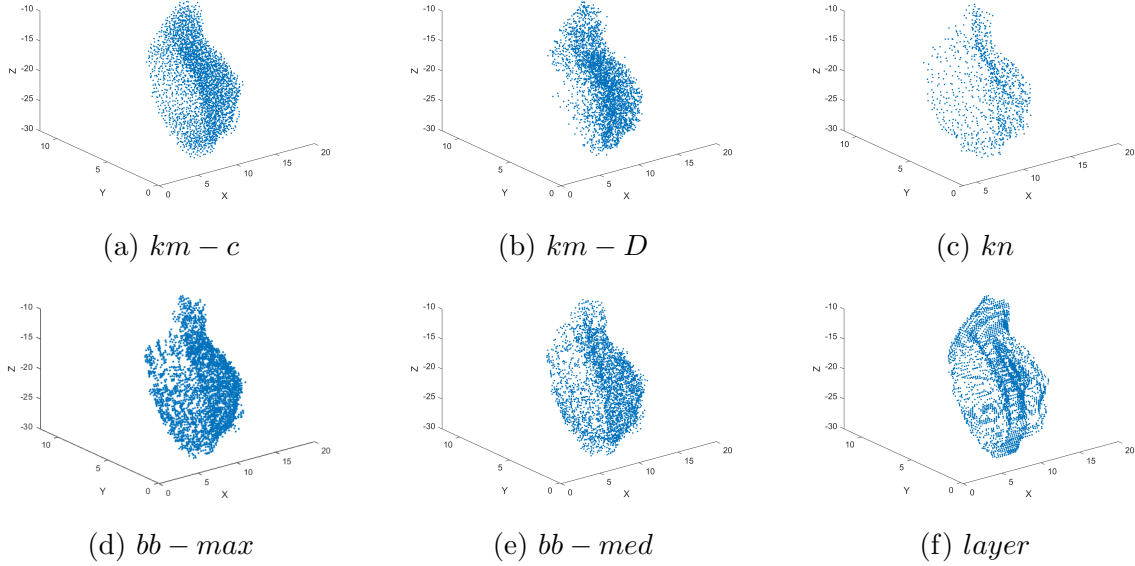


Figure 6.6: Every down-sampling approach but integer depicted at the $n = 10$ rate.

only considered centroids (aggregated $D_{v,b}$ values) that were, when summed over a given centroid's row, within the 20 closest to the sum of the current voxel's $D_{v,b}$ value. This could mean that in practice a voxel might fail to be assigned to its actual best centroid, however, it gains considerably in terms of timing, and the output became reliable. It was also given an iteration cutoff of 75.

The k -Neighbour approach required the definition of a neighbourhood for each voxel. Neighbourhoods were selected under the assumption that many voxels are the central element in a roughly $3 \times 3 \times 3$ voxel grid. As a result, each voxel's 26 nearest neighbours by euclidean distance were checked at each iteration of the algorithm.

Based on the visualization, the k -means algorithm with coordinate distances had its voxels spread out fairly evenly, most resembling the integer approach. The nearest neighbour approach also ended up rather uniformly distributed, possibly because voxels could not break out of their original neighbourhoods, while the $D_{v,b}$ -based approach seemed to concentrate in regions where there was more, likely differentiated, beamlet activity (i.e.,

the centre of the beam), choosing fewer representatives along the edges.

The layered model is perhaps the most visually interesting, with recurring shapes shifting towards the centre. It was implemented using the builtin `alphaShape()` and `boundaryFacets()` functions in MATLAB. `alphaShape` created a boundary volume that enveloping the voxel coordinates with a radius parameter set to 1.5mm. Following this envelopment, `boundaryFacets` returned all the points that made up the vertices of this volume, of which the unique ones were our outer voxels. Note, many implementations of this model are possible, this one was chosen as it works on both regular grids and deformable registration scans.

6.5.2 Numerical Results

As a baseline for assessing plan quality, the full FMO model (6.1) was run. The model converged in 2,591.3 seconds, or nearly 45 minutes. Recall that the FMO serves as an indicator problem for harder, more realistic implementations, and that the associated problems tend to scale exponentially (Ripsman et al., 2022). The sampling methods, on the other hand, will not scale with problem-difficulty, so slightly longer sampling times may be acceptable if they yield better results.

The metrics for the full plan are given in Table 6.1. They include the average dose to the heart (the OAR of interest), $\text{Avg}_{\mathcal{H}}$, as well as the hottest 1 and 10 cubic centimetre values (D1cc and D10cc). Note the average heart dose only applies to the small section (under 41cc) of the heart, which means that in reality, on the full heart dose grid of 42,342 voxels, this amount is actually a much more reasonable 0.3996 Gy. The higher amount will continue to be used for the purpose of comparison in this section, however. The D1cc and D10cc are unaffected by this conversion, by definition.

The dose metrics include the average and minimum dose to the target ($\text{Avg}_{\mathcal{T}}$ and $\text{Max}_{\mathcal{T}}$), the latter of which can be used to determine how well a sampling approach respects

the constraints given by the model (a minimum of 42.4 Gy indicates that even the coldest target voxel respects the minimum dose constraint). The remaining dose values indicate the dose to differing volumes of the **CTV** (D0.5%, D95% and D99%). Table 6.1 is the ideal values of each metric under the current model setup.

OAR			PTV				
$\text{Avg}_{\mathcal{H}}$	D1cc	D10cc	$\text{Avg}_{\mathcal{T}}$	$\text{Min}_{\mathcal{T}}$	D0.5%	D95%	D99%
6.497	39.865	7.894	45.865	42.400	95.502	42.471	42.400

Table 6.1: Full plan dose outcomes in Gy.

To compare the larger set of down-sampled results, the primary outcomes of concern were divided into two categories; run times, and solution quality. The outcomes of the different methods and their run times are summarized in Table 6.2. While the solution quality metrics are reported in Table 6.3.

Based on Table 6.2, there are some immediate observations. First, in terms of the FMO run time, all cases decreased as expected as the problem size got smaller in a largely proportional fashion. $n = 10$ sampling did take the runtime down by roughly an order of 10, and 100 did even further, due to the problem becoming less meaningful at such an intense sampling rate.

As expected, the integer sampling was the simplest and therefore the fastest. The beamlet-based approaches performed very nicely as well however, barely adding a meaningful amount of time. The k -Means- D and k -Neighbour algorithms took the longest time to run, and for k -Means- D , that run time does not seem particularly correlated with the number of samples requested. This variability is not ideal, and furthermore, the model is an approximation of the true k -Means- D , so to improve its performance, it would need to run even longer, ideally in parallel with more computing power. All other models appeared to roughly run in a speed that correlated with their sample size requirements.

The fastest FMO times were obtained using the k -Neighbour approach, and one can see why by looking at Table 6.2’s final column, which shows how many down-sampled

Method	Sampling Rate	Sample Time	FMO Time	Num Voxels
Full	n	1	-	33,592
	k	33,592		
Integer	n	2	0.126	16,796
		10	0.058	3,359
		100	0.012	335
k -Means- c	k	16,796	20.717	16,796
		3,359	11.905	3,359
		335	10.330	335
k -Means- D	k	16,796	183.055	16,783
		3,359	892.274	3,359
		335	530.949	335
k -Neighbour	k	16,796	365.395	2,584
		3,359	277.433	1,060
		335	246.572	281
Beamlet-max	k	16,796	1.217	16,796
		3,359	0.147	3,359
		335	0.100	335
Beamlet-med	k	16,796	0.420	16,796
		3,359	0.115	3,359
		335	0.077	335
Layered	k	16,796	12.528	16,773
		3,359	5.015	3,344
		335	0.077	314

Table 6.2: Run time information in seconds for the various sampling methods and granularities. A lower time can be seen as better, however, quality is much more important assuming tractability. The fastest run times across each category are shown in bold for reference.

voxels an algorithm returned. Given the implementation of this approach, it seems that voxels ended up in much fewer than the requested number of clusters, especially in the larger cases. Out of the 16,796 clusters randomly seeding the algorithm at $n = 2$, for example, only 2,584 clusters were actually formed. This explains why Figure 6.6c looks a bit sparse compared to the other plans; it contains roughly a third of the voxels in the competing algorithms.

It is worth noting that all of the remaining plans have exactly the allowable amount of clusters, or close in the case of Layered and k -Means- D sampling. At the $n = 2$ sampling level, the layered algorithm was able to pack in 16,773 voxels, at $n = 10$ that number was 3,344 and at $n = 100$ it was 314. Comparing this to the 16,796, 3,359 and 335 algorithm upper-limits, respectively, the packing algorithm seemed to do a good job fitting in a fairly large number of voxels.

Given these expected granularities, the second and third fastest FMO run-times for $n = 2$ went to the beamlet based methods, and the fastest FMO time for the $n = 10$ method was for the layered approach. By the $n = 100$ sampling, this metric was slightly trivial, but max-based beamlet-approach was ahead again, indicating that this is a fast method. Given the quick and predictable run times of most methods relative to the k -means methods, and the almost troubling speed of the k -Neighbour algorithm, it is interesting to investigate how these speeds impact the quality of the results.

The quality metrics in Table 6.3 are best augmented by the DVHs shown in Figure 6.7. These images show that overall, at $n = 2$ sampling, the methods are largely similar, whereas by $n = 100$ sampling, the plans barely resemble their prior patterns. In particular, the layered approach does not hold up well when arbitrarily set to a sampling rate.

Of the deterministic algorithms, the median beamlet-based approach seems to hold up the best in terms of dose conformity, as it does not lead to severe overdoses like the maximum approach but seems to have a nicer profile than the integer approach, by underdosing less. The beamlet-based approaches continue to dominate, with the median version

Sampling	Metric	Method						
		<i>int - n</i>	<i>km - c</i>	<i>km - D</i>	<i>kn</i>	<i>bb - max</i>	<i>bb - med</i>	<i>layer</i>
$n = 2$	Avg $\mathcal{H}$	6.050	5.967	6.034	5.298	6.806	6.720	7.119
	D1cc	37.287	35.928	36.089	32.362	40.248	39.854	40.168
	D10cc	7.413	7.278	7.446	6.459	8.526	8.501	9.061
	Avg $\mathcal{T}$	45.357	45.329	45.332	43.992	45.246	45.267	45.478
	Min $\mathcal{T}$	30.678	30.035	28.067	28.489	35.293	35.666	42.042
	D0.5%	91.243	91.024	87.539	77.043	91.292	91.450	92.080
	D95%	42.433	42.402	42.400	41.985	42.419	42.413	42.462
	D99%	42.297	42.095	42.118	40.638	42.277	42.192	42.400
$n = 10$	Avg $\mathcal{H}$	5.074	3.952	4.165	4.104	6.727	6.437	6.170
	D1cc	31.669	24.123	27.957	25.594	40.050	38.812	38.184
	D10cc	6.032	4.699	4.669	4.962	8.306	7.928	7.558
	Avg $\mathcal{T}$	44.411	43.558	44.019	43.302	44.380	44.347	45.272
	Min $\mathcal{T}$	20.869	19.894	16.411	20.460	32.193	34.014	29.377
	D0.5%	85.372	65.496	76.031	68.535	91.158	91.017	91.846
	D95%	42.085	41.687	40.518	41.013	41.984	42.114	41.272
	D99%	40.249	38.171	36.527	38.041	40.459	40.864	37.922
$n = 100$	Avg $\mathcal{H}$	2.352	2.357	3.034	2.729	5.702	4.488	0.143
	D1cc	14.594	12.828	19.884	18.695	38.451	30.941	0.424
	D10cc	2.750	2.911	3.359	3.121	6.423	4.999	0.234
	Avg $\mathcal{T}$	41.200	42.717	42.611	42.422	45.705	40.603	222.368
	Min $\mathcal{T}$	3.131	4.083	9.843	12.605	14.492	12.259	0.000
	D0.5%	75.715	82.589	86.719	62.988	96.340	71.255	3141.199
	D95%	25.604	30.560	29.614	32.952	28.563	28.450	0.174
	D99%	12.994	21.657	21.175	25.984	24.997	24.481	0.003

Table 6.3: Dose metrics in Gy for the various sampling methods and granularities. The component of interest in this case is how much the plan violates the required minimum dose, while keeping the rest of the metrics high-quality.

having a better show at the $n = 10$ level and the maximum version at the $n = 100$ stage.

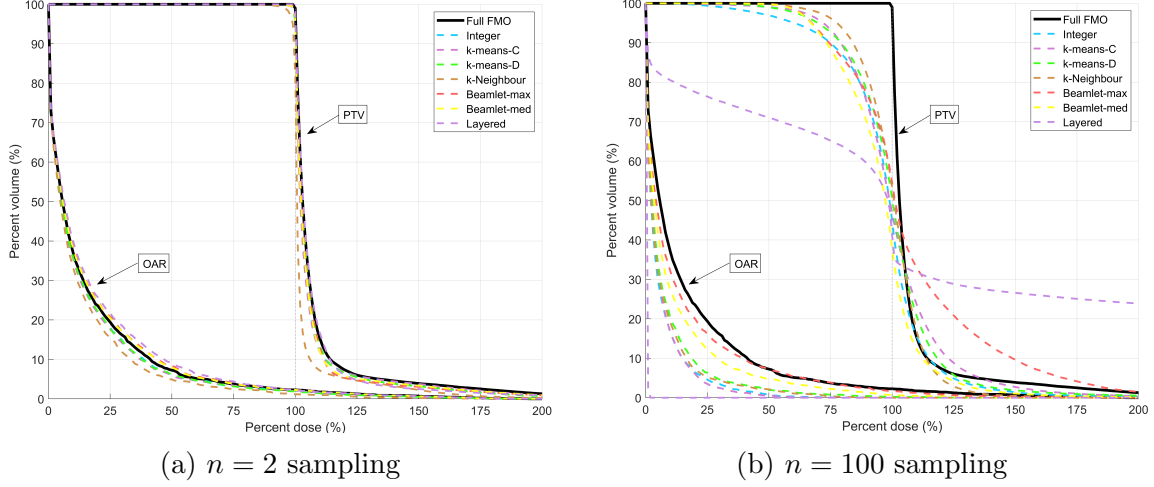
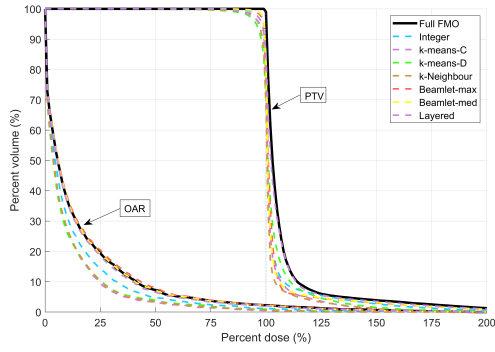


Figure 6.7: DVHs of the two more extreme sampling rates.

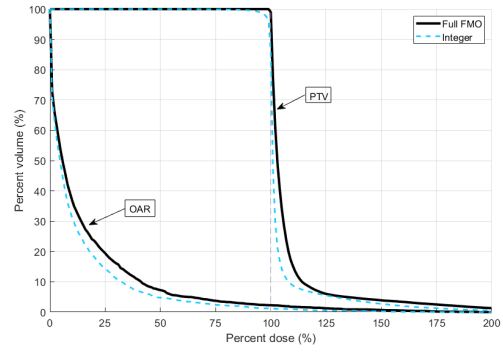
Once again, switching focus to the $n = 10$ rate, specifically, all plans are shown together in Figure 6.8a and then separately in the remainder of Figure 6.8. It can be seen that in terms of representing the original plan, beamlet-based or layered approaches are significantly closer than the other plans. What does jump out based on the DVHs however, is that while the layered approach better estimates the tail of the distribution (Figure 6.8h), the beamlet-based approaches seem to capture the critical constraints much more accurately at the upper right corner of the distribution. This is likely why they remain much better estimators at the more aggressive sampling levels as well.

In spite of dropping many voxels, the k-Neighbour approach performs quite well at the most aggressive $n = 100$ level of sampling. Using only 281 centroid $D_{v,b}$ values, it delivered a fairly high-quality plan. With that in mind, the roughly four minutes of pre-processing time required to run the sampling seems quite trivial. There is however, some significant worst-case loss at the higher granularities for this approach.

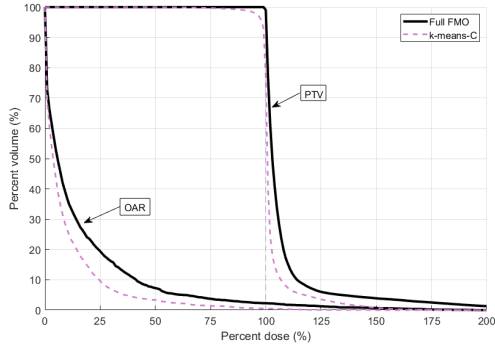
Going back to Table 6.3, the metrics tell a similar story. The layered approach does not perform well at the $n = 2$ or $n = 100$ sampling extremes. Moreover, at the $n = 2$



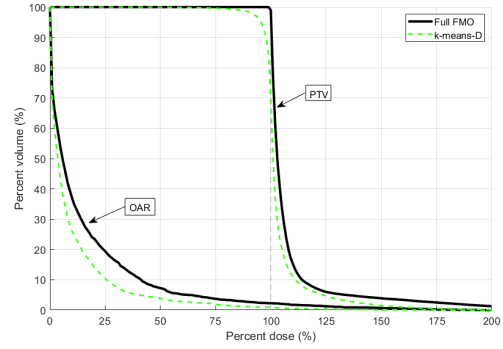
(a) All plans



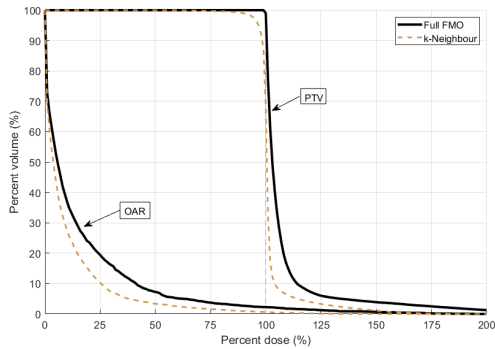
(b) Integer



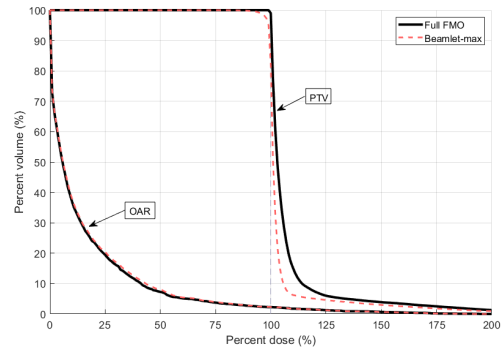
(c) k -means- c



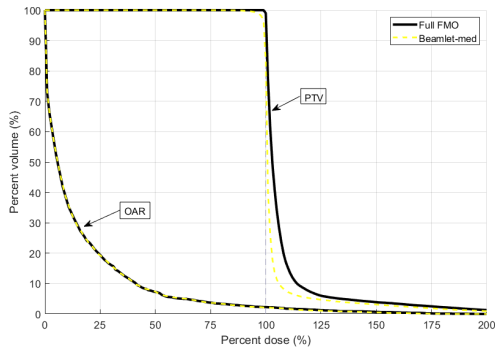
(d) k -means- D



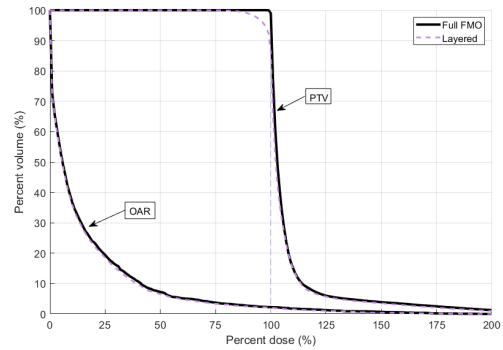
(e) k -Neighbour



(f) Beamlet-max



(g) Beamlet-med



(h) Layered

Figure 6.8: All DVH diagrams for the $n = 10$ sampling rate.

sampling level, it gives up on more heart dose than the original FMO model, for better tumour values, although it is far from the only method to do so. It does come the closest to respecting the minimum dose rate of 42.4 Gy at the $n = 2$ sampling rate but then loses out to the beamlet based median approach at $n = 10$ sampling and is the worst offender at the $n = 100$ rate. Overall, it would appear that the beamlet-based median selection algorithm was the most robust to change across all levels for the examined patient, and in particular, at the likely most-representative $n = 10$ level. Going forward, this method should be seriously considered for down-sampling implementations.

6.6 Conclusions

A set of sampling methods were thoroughly tested on a retrospective patient plan. A new proposed geometry-preserving method was proposed, and six additional methods were showcased, including adapted versions of methods previously introduced in the literature. The findings indicate that while the newly proposed layered approach preserved elements of the problem geometry at middle-of-the-road sampling rates, it did not preserve the most critical modelling components at more aggressive sampling rates. Rather, the adapted beamlet-based median approach worked quite well, and was a front runner in many of the relevant comparisons. With the downside of stochasticity and higher pre-processing times, the k -Neighbour approach was also shown to be a strong methodological contender, due to its incredible FMO model speed ups via voxel reduction, without a serious quality downside.

Moving forward, this chapter provides a compelling case for moving away from integer sampling, towards methods that better preserve problem geometry. It will be of interest to the research community to determine which methods continue to stand out as the approaches are tested on larger databases of case-studies, and potentially other steps of the [RTTP](#) process.

Chapter 7

Future Research Directions

This chapter touches on several directions for future work. These areas build on research done within and surrounding the thesis topics but were not completed in depth due to time and resource constraints. Section 7.1 introduces inverse optimization as it relates to parameter selection in the space of radiation therapy. Section 7.2 outlines a proposed methodology inspired by the warm start cutting plane approach in Chapter 2. Section 7.3 discusses future possibilities with regards to integrating uncertainty in radiation therapy, while Section 7.4 and 7.5 discuss the next steps for the BAO and DAO projects, respectively. Finally, Section 7.6 addresses the future work in sampling.

7.1 Parameter Selection in Radiation Therapy

Practical FMO formulations tend to have multiple, competing objectives due to the many trade-offs inherent to RTTP. Evaluating the relative importance of these objectives is typically left up to expert intuition in clinic, which is difficult to replicate in a laboratory, or automated setting. Weighting parameters can be used as a stand-in for expert opinion, in order to define the relative utility of an objective within a model, however, tuning these

parameters in the absence of experts can result in unfaithfully-calibrated models.

The use of [inverse optimization \(IO\)](#) has gained popularity in recent years, in part, due to its relationship to the ubiquitous machine learning applications in the literature. It is related to robust and stochastic methods in that it concerns parametric uncertainty, but unlike those cases that try to immunize against the uncertainty, [IO](#) tries to learn point estimates for the parameter values based on existing information. For radiation therapy, that might mean using databases of past radiation treatment plans, designed by experts, to learn the objective or constraint matrix weightings.

[IO](#) for finding objective function values of [LP](#) models was pioneered by [Ahuja and Orlin \(2001\)](#), with the assumption that any proposed solution is on the boundary of the feasible region. In the mathematical sense, optimal solution, $\mathbf{x}^*$ is given for the following equation, where vector γ is unknown,

$$\min\{\gamma^T \mathbf{x} \mid \mathbf{A}\mathbf{x} \geq \mathbf{b}, \mathbf{x} \in \mathbb{R}^{|x|}\}. \quad (\mathbf{FO})$$

This is referred to as the forward optimization model. In their paper, [Ahuja and Orlin \(2001\)](#) try to find the constraint value that given a optimal-point $\mathbf{x}^*$, minimizes the objective function γ as well. When this is formulated as the distance between γ and from some initial set of objective parameter values $\mathbf{c}$, with an L_1 and L_∞ norm and complementary slack properties, they show that it leads to a linear formulations of the inverse of [\(FO\)](#).

Building on this work, [Chan et al. \(2014a\)](#) generalized [Ahuja and Orlin \(2001\)](#)'s model to account for more realistic implications of noisy, imperfect data, where a past solution is not necessary optimal, or on the boundary, but remains feasible. More than one model is

presented in the paper, but the overall [IO](#) formulation looks as follows,

$$\begin{aligned}
& \min_{\gamma, \mathbf{y}, \epsilon} \epsilon \\
& \text{s.t. } \mathbf{A}^T \mathbf{y} \leq \gamma, \\
& \quad \mathbf{A} \mathbf{x}^* \geq \mathbf{b}, \\
& \quad \gamma^T \mathbf{x}^* = \mathbf{b}^T \mathbf{y} + \epsilon, \\
& \quad \|\gamma\| = 1,
\end{aligned} \tag{IO}$$

where ϵ is the deviation of an observed solution from optimality, $\mathbf{y}$ represent dual variables, the first constraint is dual feasibility, the second is primal feasibility, the third is strong duality, and the final constraint prevents the trivial solution of all zero values in the objective function. Depending on how ϵ is incorporated and or penalized, their model accounts for slack in different ways. [Chan et al. \(2014a\)](#) then go on to demonstrate their more flexible models on real, [RTTP](#) cases. More recently, [Chan et al. \(2019\)](#) looked at inverse models, and showed that a closed form-approach can be taken to solving inverse models with uncertain objective parameters, again, demonstrating this on [RTTP](#) examples. They further derived goodness-of-fit parameters for their models, to give inverse an equivalent measurement to other noisy data-fitting processes used in [OR](#), like regression analysis.

The goal of this section of the proposal, is to continue along this geometric perspective, seeing as much of the field of [IO](#) has continued to operate off optimization models, despite the possibility of generating closed-form solutions, that might greatly simplify solving them. The idea of this section is to proposed a 3-pronged outline for future work: 1. to create a more structural understanding of the [IO](#) problem, that allows for more straight-forward interpretations, making way for more complex generalizations, 2. to broaden these closed-form findings, and 3. to apply this intuition and determine benefits using various [IO](#) problems, including [RTTP](#).

7.1.1 Inverse Methods

In [Chan et al. \(2019\)](#), the closed-form solution of (IO) is derived through the use of the dual norm. While this was insightful work, it obfuscates a lot of the intuition behind the mechanisms of inverse, which has led OR practitioners to continue to use optimization models instead. Figure 7.1 instead shows the geometric intuition behind the proofs of performing IO in L_2 , with a feasible point. All IO models with unknown parameters, in essence, return the closest constraint to a given point (or centroid of a cluster of points, etc.), as the objective γ , with varying degrees of efficiency. The hypothesis in this section is that this can be found in the closed-form by projecting the point onto all constraints and taking the minimum value.

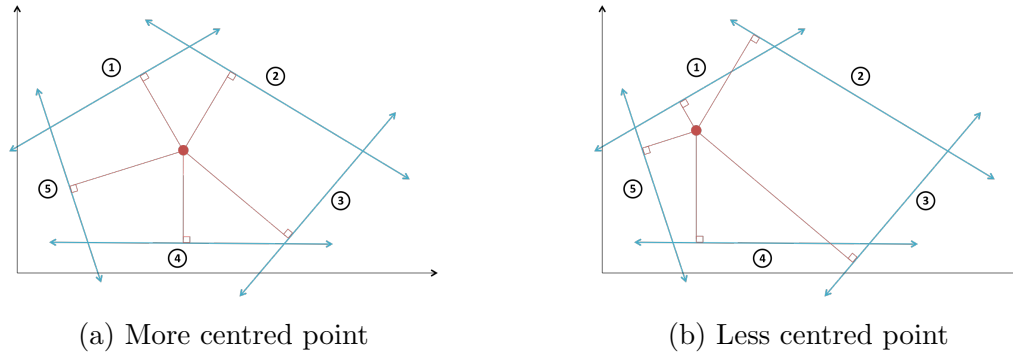


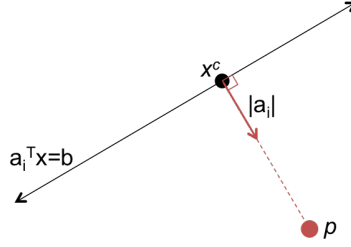
Figure 7.1: Two examples of an IO problem with a different feasible solution, where γ should be chosen to be exactly equal to constraint 1.

The following propositions keep this intuition intact, while proving the feasibility of such a method.

Proposition 4. *The distance d_i of an observation $\mathbf{p}$ from each constraint $\mathbf{a}_i^T \mathbf{x} = b_i$ can be calculated for any i using the closed-form formula:*

$$d_i = \frac{|b_i - \mathbf{a}_i^T \mathbf{p}|}{\sqrt{\mathbf{a}_i^T \mathbf{a}_i}} \quad (7.1)$$

Proof. Given a hyperplane of dimension n : $a_{i1}x_1 + \dots + a_{in}x_n = b_i$, and a point $\mathbf{p} = (p_1 \dots p_n)$, the closest euclidean distance between $\mathbf{p}$ and plane $\mathbf{a}_i^T \mathbf{x} = b_i$ is going to be the distance, perpendicular to the plane that joins the two, at point, $\mathbf{x}^c$.



To find this distance, follow the normal vector $\mathbf{a}_i$, which moves perpendicular to the hyperplane.

Any point $\mathbf{p}$ is some slack distance s from a given hyperplane, i.e.,

$$\mathbf{a}_i^T \mathbf{p} + s = b_i. \quad (7.2)$$

If $s = 0$, $\mathbf{p}$ is on the plane.

Since $\mathbf{x}^c$ is on the hyperplane, $\mathbf{a}_i^T \mathbf{x}^c = b_i$.

But there also $\exists t \in \mathbb{R}$ such that $\mathbf{x}^c = \mathbf{p} + \mathbf{a}_i t \implies \mathbf{a}_i^T (\mathbf{p} + \mathbf{a}_i t) = b_i$, or,

$$\mathbf{a}_i^T \mathbf{p} + \mathbf{a}_i^T \mathbf{a}_i t = b_i. \quad (7.3)$$

Combining equations (7.2) and (7.3) yields $s = \mathbf{a}_i^T \mathbf{a}_i t \implies t = \frac{s}{\mathbf{a}_i^T \mathbf{a}_i}$, or,

$$t = \frac{b - \mathbf{a}_i^T \mathbf{p}}{\mathbf{a}_i^T \mathbf{a}_i} \quad (7.4)$$

Then the final step is finding distance, $d_i = |\mathbf{a}_i t| = \sqrt{\mathbf{a}_i^T \mathbf{a}_i} |t| \implies |t| \sqrt{\mathbf{a}_i^T \mathbf{a}_i}$, or, the distance is,

$$d_i = \frac{|b - \mathbf{a}_i^T \mathbf{p}|}{\sqrt{\mathbf{a}_i^T \mathbf{a}_i}} \quad (7.5)$$

This calculation may be applied to every constraint to get a set of distances. $\square$

Proposition 5. *The projection with the smallest distance is guaranteed to be feasible.*

Proof. This can be proven by contradiction, based on convexity (no angles $> 180^\circ$). If the minimum perpendicular distance to a boundary in a convex shape is $\min_i \{d_i\}$, and that point is out of the feasible region, there must be a closer constraint, blocking that original constraint, with a perpendicular distance less than $\min_i \{d_i\}$, contradicting the original claim. $\square$

Therefore, the optimal solution to **(IO)** is $\gamma = \mathbf{a}_i$ where

$$i = \arg \min_{i \in \{1, \dots, m\}} \{d_i\}$$

and the objective function value at optimality is b_i .

7.1.2 Future Steps and Applications to RTTP

These propositions have broader inverse implications, if the geometrical intuition can be expanded into non-feasible inverse solutions. A similar intuition to Section 7.1.1 must be developed to project infeasible observations back into the feasible region. It is not immediately clear whether this can be done in constant time, however, there are a few observations that can be made geometrically, that may help this process along, based on Figure 7.2.

The first observation is that only infeasible constraints ($s < 0$) should need to be included in projection calculations, as the rest should remain feasible if a proper projection methodology is followed. The second observation is that the closest projection may not be feasible, unlike with points starting inside the feasible region. The third observation is that if a projection is feasible to all initially infeasible planes, then making the objective

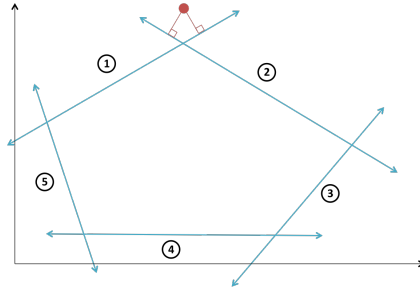


Figure 7.2: An infeasible example of an IO problem, where a corner point is optimal.

match the satisfying constraint is optimal. The final observation, however, is that if no projections are feasible to all violated constraints, the closest feasible projection lies on an intersection of at least two constraints, meaning there are infinite feasible objective function lines (ranging from the slope of one line to the other, i.e., the opposite of forward optimization, where a corner point means only one feasible solution, and a boundary point means infinite).

Moving forward, the goal is to flesh out the heuristic and then use these geometric principles to analyze other areas of inverse optimization, and generalizations including multi-point optimization and different distance metrics. Ideally, this will form a simpler baseline for inverse, so more complicated problems, such as those that currently require bi-level optimization, become much more accessible. Through these methods, the aim is to challenge some of the machine learning applications that have been growing in popularity outside, or even inside the field of OR.

To evaluate the findings, the goal is to test these solution methodologies on RTTP problems, to determine whether the closed-form approaches provide the speed-improvements and stability hypothesized. Ideally, if these methods are effective, applying some of them to the constraint matrix $\mathbf{A}$ may also provide learning opportunities in radiation therapy and other similar applications.

7.2 Extensions to the Warm Start Methodology

Chapter 2 introduced a series of cutting plane approaches for uncertainty in the [RTTP](#) problem. It looked at the structural elements of the modelling and tried to find faster ways to solve the model based on these elements. Ultimately, the [MILP](#) approach was found to be too large to solve efficiently using any of the methods proposed. The warm start, however, was not only quite fast, but it dominated hours of [MILP](#) run time for the [RTTP](#). This leads to the question of whether there is a way to design a better solver based on this warm start methodology that fully explores the region. The following section gives a sketch of what this alternative, custom branch-and-bound solver might look like.

Consider the original warm start objective function that penalizes both overdose and non-uniformity,

$$\alpha \sum_{\theta \in \Theta} f_{\theta} + (1 - \alpha) \cdot \sum_{s \in \mathcal{H}, \mathcal{T}} \sum_{v \in \mathcal{V}_s} \sum_{k \in \mathcal{K}} \frac{p_{v,k}}{|\mathcal{V}_s|} d_{k,v}.$$

The first term of the function is necessary in the warm start configuration, as it is the only component binding the maximum fluence to the outcome of the problem. In terms of the bigger picture, however, it is unclear whether minimizing this term always yields closer-to-optimal results, so it would need to be dropped. Reducing the problem essentially back to the ([FMO](#)) model, with an explicit maximum dose term f_{θ} .

$$\min \quad \sum_{s \in \mathcal{H}, \mathcal{T}} \sum_{v \in \mathcal{V}_s} \sum_{k \in \mathcal{K}} \frac{p_{v,k}}{|\mathcal{V}_s|} d_{k,v} \tag{7.6a}$$

$$\text{s.t.} \quad \sum_{k \in \mathcal{K}} \tilde{p}_{v,k} d_{k,v} \geq L_v \quad \forall \tilde{\mathbf{p}}_v \in \mathcal{P}_v, v \in \mathcal{V}_{\mathcal{T}}, \tag{7.6b}$$

$$\sum_{b \in \mathcal{B}} D_{b,v,k} w_b - d_{k,v} = 0 \quad \forall k \in \mathcal{K}, v \in \mathcal{V}_s, s \in \{\mathcal{H}, \mathcal{T}\}, \tag{7.6c}$$

$$w_b \leq f_{\theta} \quad \forall b \in \mathcal{B}_{\theta}, \theta \in \Theta. \tag{7.6d}$$

Normally, an additional set of variables that enforce the on or off state of a beamlet $\mathbf{x}$ and variables enforcing the left and right leaves $\mathbf{l}$ and $\mathbf{r}$, respectively, are introduced at this point, as in model 2.12. However, this proposed method involves leaving the model as is, and instead, iteratively branching on w_b values that lie in the range $(0, f_\theta)$. Note this is non-inclusive. One branch then fixes the value to f_θ and re-solves the problem, the other fixes the value to 0, as depicted in Figure 7.3.

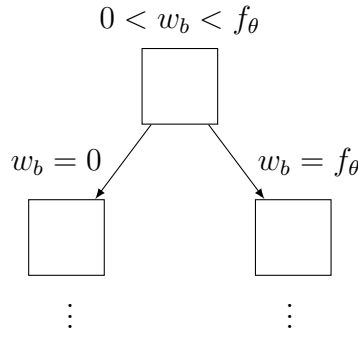


Figure 7.3: Proposed novel branch-and-bound approach.

There are several benefits to solving this problem iteratively, rather than explicitly including binary variables. The first, is that it achieves the same goal, without a huge amount of additional variables, that do not enforce the decision space in the LP relaxation. The second is that a heuristic approach may be added at each branch where values are fixed to f_θ , since if two non-adjacent beamlets are fixed to f_θ in a given row, the middle ones must be fixed to it as well. I.e., infeasible parts of the decision space can be implicitly discarded heuristically, throughout the algorithm, with no fear of pruning an optimal solution.

Since this method already incorporates robustness into the model, it may be solved iteratively using LP cutting plane approaches, which should maintain a fast run time, without needing to integrate these methods into the solvers' branch-and-bound frameworks. It may also benefit from a warm start generated with the original warm-start objective (2.12a) for pruning, and furthermore, would be an increasingly good candidate for the warm start fixing heuristic 5, as each iteration brings it closer to this feasible solution,

should the algorithm need to be stopped early.

This future direction aims to achieve roughly the same goals as Chapter 5’s DAO approach, and it would be interesting to compare the two, to see which method performs better in practice.

7.3 Future Light Pareto DRO Approaches to RTTP

Chapter 3 introduced the concept of LPRO which integrated optimality trade-offs and Pareto optimality into RO methodologies. The chapter also makes reference to DRO base-methods which are useful when more distributional information is available. In this section, the use cases for RO and DRO base-cases are clarified, and lead to proposals for future light Pareto distributionally robust optimization (LPDRO) work.

More specifically, while the intra-fractional polyhedral uncertainty discussed extensively in Chapter 2, lends itself to RO it may also be formulated as a DRO problem, which may be extended using LPDRO methods. In this section, the studied intra-fractional motion is further classified in Section 7.3.1, and the justification for keeping this method within the RO space is explained. An example of how the case could be extended to a DRO and then LPDRO settings are then shown in in Section 7.3.2. A proposal for a more appropriate inter-fractional future direction for LPDRO is then introduced in Section 7.3.3.

7.3.1 Intra-Fractional Motion and RO

The uncertainty or ambiguity in this thesis lies in the proportion of time spent in each breathing phase experienced by a patient. These phases are *finite*, as the images are taken using a 4D scan *a priori*. The uncertainty is on the proportion of time $\tilde{\mathbf{p}}$ spent in that phase, for each fixed location at phase k in the $D_{v,b,k}$ matrix, as shown in Figure 7.4.

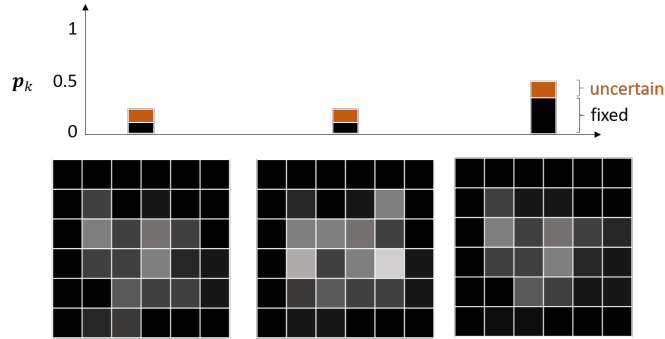


Figure 7.4: Certain $D_{v,b,k}$ but uncertain allocation across $|\mathcal{K}| = 3$ phases.

In terms of the breathing cycle, the patient's breathing and the polyhedral uncertainty set around the breathing PMF may be represented as in Figure 7.5. the uncertainty lies on what range will be realized within the orange section of the bars in the graph.

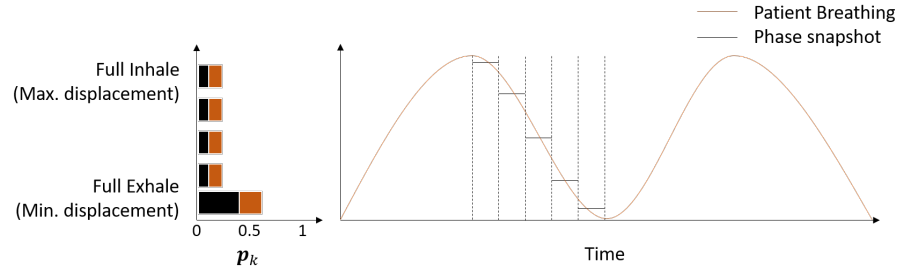


Figure 7.5: A PMF around the time spent in $|\mathcal{K}| = 5$ phases with uncertainty, where the orange regions are uncertain.

In a DRO setup, the uncertainty in phases may be taken as an ambiguity set in a number of ways. Using a pessimizing oracle, every single point in a given phase could be considered equally likely, in which case it can be seen as an example of a support set with a complete weighting at a single point, which collapses back into the RO case, as discussed in the introduction of Delage and Ye (2010). Alternatively, a distribution could be assumed, and the first and second moments could then be described. Say the distribution is assumed to be normal with an unknown mean and variance. This set of normal distributions could be addressed in a DRO framework, as in Figure 7.6

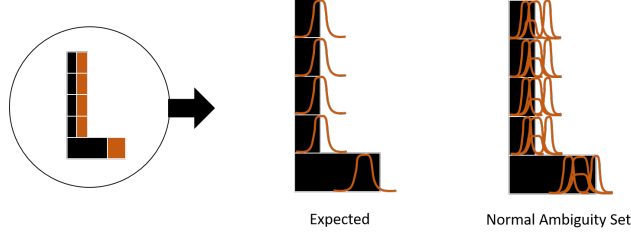


Figure 7.6: Ambiguity set based on a polyhedral uncertainty set.

The issue with these sets is that the nature of the [PMF](#) requires the results of a set of probabilities to sum to 1, and that is not always (or often) the case, as in Figure 7.7.

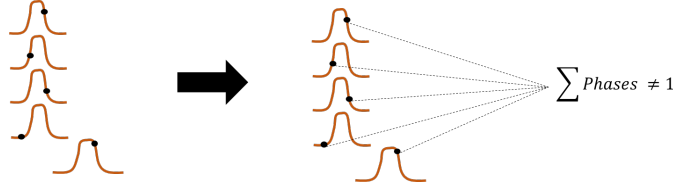


Figure 7.7: Choosing points from 5 separate random distributions will likely not sum to 1.

This means that in the broad distributional sense, there are only two ways we found to formulate the problem. Either considering *only* the L_∞ distance, which collapses back into the [RO](#) models. Or a scenario-based approach may be considered, e.g. by manually generating relevant scenarios that sum to 1, or by running Algorithm 6 to generate a database of cases. The manual approach is illustrated in the next section.

7.3.2 DRO Version of the Intra-Fractional Problem

The following is a form of a general [DRO](#) problem given by [Delage and Ye \(2010\)](#),

$$\min_{\mathbf{x} \in \mathcal{X}} \max_{f_\xi \in \mathcal{D}} \mathbb{E}_{f_\xi} [h(\mathbf{x}, \xi)], \quad (7.7)$$

where $h(\mathbf{x}, \xi)$ is a convex function, $\mathcal{X}$ is the set of all feasible $\mathbf{x}$ and $\mathcal{D}$ is an ambiguity set representing the set of all possible distributions f_ξ on the uncertain parameter ξ .

For a simple **DRO** that mirrors Section 3.3, assume that all breathing patterns are drawn from a set of feasible, uniform distributions, $\mathcal{D}$, where the feasibility check must ensure that any $\mathbf{p} \sim \mathbf{f}_{\mathbf{p}} \in \mathcal{D}$, also ensures $\sum \mathbf{p} = 1$, i.e., a polyhedral set. The uncertain function is as follows, for $f_{\tilde{\mathbf{p}}} \in \mathcal{D}$, and each voxel v ,

$$\mathbb{E}[h(\mathbf{w}, \tilde{\mathbf{p}})] = \mathbb{E} \left[\sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} \tilde{p}_{v,k} w_b \right],$$

in both basic **FMO** and **CVaR** models. The model is linear, so the expectation is only on the breathing, meaning the model may be simplified as follows,

$$\mathbb{E}[h(\mathbf{w}, \tilde{\mathbf{p}})] = \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} \mathbb{E}[\tilde{p}_{v,k}] w_b.$$

In the **CVaR** case, the **DRO** model looks as follows,

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{|\mathcal{H}|} \sum_{v \in \mathcal{H}} \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} p_{v,k} w_b & (7.8) \\ \text{subject to} \quad & \bar{d}_v \geq \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} \mathbb{E}[\tilde{p}_{v,k}] w_b - \bar{\zeta}, \quad \forall v \in \mathcal{T}, f_{\tilde{\mathbf{p}}} \in \mathcal{D} \\ & \underline{d}_v \geq \underline{\zeta} - \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} \mathbb{E}[\tilde{p}_{v,k}] w_b, \quad \forall v \in \mathcal{T}, f_{\tilde{\mathbf{p}}} \in \mathcal{D} \\ & \bar{\zeta} + \frac{1}{(1-\beta)|\mathcal{T}|} \sum_{v \in \mathcal{T}} \bar{d}_v \leq \lambda_U U, \\ & \underline{\zeta} - \frac{1}{(1-\beta)|\mathcal{T}|} \sum_{v \in \mathcal{T}} \underline{d}_v \geq \lambda_L L, \\ & w_b \geq 0 \quad \forall b \in \mathcal{B}. \end{aligned}$$

As usual, for the purpose of the derivation, the objective uses the nominal $\mathbf{p}$ value, while constraints contain the uncertainty. λ is a constant denoting some ≤ 1 value that allows deviation from the upper and lower bounds under worst-case uncertainty. As in the **RO**

models, a Pareto or light Pareto [DRO](#) model may be derived by fixing the original objective in a subsequent model,

$$\frac{1}{|\mathcal{H}|} \sum_{v \in \mathcal{H}} \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} p_{v,k} w_b \leq z^{\text{RO}} + \epsilon.$$

Which leads to a model with a new objective function, given a new $\hat{\mathbf{p}}$, which belongs to the reciprocal uncertainty set of $\mathbf{f}_{\hat{\mathbf{p}}} \in \mathcal{D}$, and new variables $\bar{d}_v$ and $\underline{d}_v$ that enforce the specific objective uncertainty set,

$$\begin{aligned} & \min_{\mathbf{w}, \mathbf{d}, \bar{\mathbf{d}}, \bar{\zeta}, \underline{\zeta}} \left[\bar{\zeta} + \frac{1}{(1-\beta)|\mathcal{T}|} \sum_{v \in \mathcal{T}} \bar{d}_v \right] - \left[\underline{\zeta} - \frac{1}{(1-\beta)|\mathcal{T}|} \sum_{v \in \mathcal{T}} \underline{d}_v \right] \quad (7.9) \\ & \text{subject to} \quad \frac{1}{|\mathcal{H}|} \sum_{v \in \mathcal{H}} \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} p_{v,k} w_b \leq z^{\text{RO}} + \epsilon \\ & \quad \bar{d}_v \geq \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} \hat{p}_{v,k} w_b - \bar{\zeta} \quad \forall v \in \mathcal{T}, \\ & \quad \underline{d}_v \geq \underline{\zeta} - \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} \hat{p}_{v,k} w_b \quad \forall v \in \mathcal{T}, \\ & \quad \bar{\zeta} + \frac{1}{(1-\beta)|\mathcal{T}|} \sum_{v \in \mathcal{T}} \bar{d}_v \leq \lambda_U U, \\ & \quad \underline{\zeta} - \frac{1}{(1-\beta)|\mathcal{T}|} \sum_{v \in \mathcal{T}} \underline{d}_v \geq \lambda_L L, \\ & \quad \bar{d}_v \geq \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} \mathbb{E}[\tilde{p}_{v,k}] w_b - \bar{\zeta}, \quad \forall v \in \mathcal{T}, \mathbf{f}_{\hat{\mathbf{p}}} \in \mathcal{D} \\ & \quad \underline{d}_v \geq \underline{\zeta} - \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} \mathbb{E}[\tilde{p}_{v,k}] w_b, \quad \forall v \in \mathcal{T}, \mathbf{f}_{\hat{\mathbf{p}}} \in \mathcal{D} \\ & \quad \bar{\zeta} + \frac{1}{(1-\beta)|\mathcal{T}|} \sum_{v \in \mathcal{T}} \bar{d}_v \leq \lambda_U U, \\ & \quad \underline{\zeta} - \frac{1}{(1-\beta)|\mathcal{T}|} \sum_{v \in \mathcal{T}} \underline{d}_v \geq \lambda_L L, \\ & \quad w_b \geq 0 \quad \forall b \in \mathcal{B}. \end{aligned}$$

For a demonstration of how the [CVaR](#) models in Section 3.3 could extend to [DRO](#), let

$\mathcal{D}_1 \subset \mathcal{D}$ be a set of five distinct breathing distributions drawn from the initial ambiguity (uncertainty) set, $\mathcal{D}_1 := \{\mathbf{f}_{\tilde{\mathbf{p}}_1}, \mathbf{f}_{\tilde{\mathbf{p}}_2}, \mathbf{f}_{\tilde{\mathbf{p}}_3}, \mathbf{f}_{\tilde{\mathbf{p}}_4}, \mathbf{f}_{\tilde{\mathbf{p}}_5}\}$. Based on the nature of polyhedral sets, the expectations of each phase may be characterized within the set, without claiming knowledge of the bounds of the set. Five feasible sets of expected values are selected, well within range of a typical breathing pattern in table 7.1.

Pattern	$\mathbb{E}[\tilde{p}(1)]$	$\mathbb{E}[\tilde{p}(2)]$	$\mathbb{E}[\tilde{p}(3)]$	$\mathbb{E}[\tilde{p}(4)]$	$\mathbb{E}[\tilde{p}(5)]$
1	0.125	0.125	0.125	0.125	0.50
2	0.13	0.13	0.13	0.13	0.48
3	0.12	0.12	0.12	0.12	0.52
4	0.1375	0.1375	0.1375	0.1375	0.45
5	0.1125	0.1125	0.1125	0.1125	0.55

Table 7.1: Expected values for five $\tilde{\mathbf{p}}$ distributions for the DRO model, making up $\mathcal{D}_1$.

Furthermore, by setting λ_L and λ_U to be .98 and $\frac{1}{.98}$, respectively, conservatism may be reduced within the CVaR model. The model was run at $\epsilon = 0.1, 0.2, 0.3, 0.4, 0.5$, as well as $\epsilon = 0$, representing a pure Pareto distributionally robust optimization (PDRO) solution, and the original model (7.8) alone, as the DRO model. Figure 7.8a shows the result of all 7 runs.

Based on the output, the DRO model behaves very similarly to the RO model, reducing conservatism as the ϵ value grows. Since the dose is in the constraints, any additional flexibility does little to improve the model's results over the more conservative RO model. What does improve the results is targeting the objective through PDRO and LPDRO.

It is also evident from Figure 7.8b that the difference between PRO and PDRO plans is negligible compared to the difference between PRO/PDRO and LPRO/LPDRO plans. As a matter of fact, with its narrower uncertainty set, the PDRO plans tend to overdose even more than PRO plans in the nominal model, since heart dose can be further reduced with the smaller uncertainty set. However, when flexibility is introduced, via LPDRO, it becomes even less conservative than the PRO model, since again, the uncertainty set is

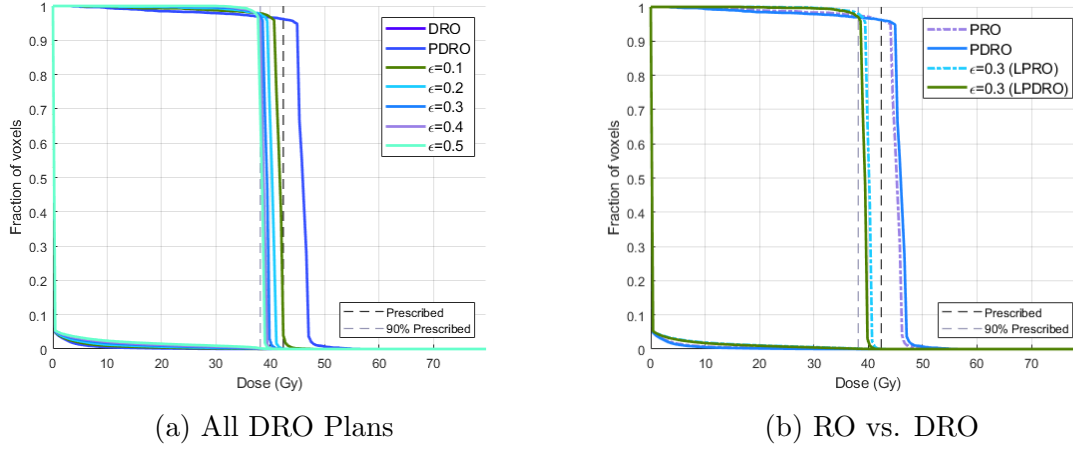


Figure 7.8: DVH plots for the DRO, PDRO and LPDRO model results with the $\mathcal{D}_1$ ambiguity set and a nominal realized $\mathbf{p}$. Note, the solid line plans have a DRO based model, and the broken lines have an RO base.

less restrictive.

The models are efficient to run, with the initial DRO model running in approximately 55 minutes, and subsequent Pareto and light Pareto models ranging from 20-35 minutes of additional runtime. Table 7.2 shows the results of each model run at the nominal, or expected breathing pattern. The results mirror Table 3.4, in that conservatism is reduced as the heart dose constraint becomes less restrictive.

Model	$\text{Avg}_{\mathcal{T}} \text{Dose}$	D0.5%	D10%	D90%	D95%	$\text{Avg}_{\mathcal{H}} \text{Dose}$	D10cc
DRO	45.2511	51.4251	46.9067	44.9915	44.8822	0.2299	15.4286
PDRO	45.2511	51.4251	46.9067	44.9915	44.8822	0.2299	15.4286
PDRO $_{\epsilon=.1}$	41.3994	43.8677	42.3685	41.0852	41.0412	0.3299	23.8178
PDRO $_{\epsilon=.2}$	40.0743	41.7213	40.7996	39.7292	39.6665	0.4299	31.2144
PDRO $_{\epsilon=.3}$	39.2066	40.8014	39.7914	38.7896	38.7658	0.5299	33.6346
PDRO $_{\epsilon=.4}$	38.6413	39.7455	39.1260	38.2205	38.2107	0.6299	36.6782
PDRO $_{\epsilon=.5}$	38.3830	39.4770	38.7935	38.0099	38.0099	0.7299	37.7698

Table 7.2: Dose metrics for the DRO models at nominal breathing pattern, $\mathbf{p}$.

7.3.3 Inter-Fractional Motion and DRO

While the previous section shows that [PDRO](#) and [LPDRO](#) methods can be applied to a polyhedral uncertainty set, this application does not fully showcase the flexibility of [DRO](#) methods as much as a case with truly distributional uncertainty would. In a paper by [Chu et al. \(2005\)](#), the motion considered is inter-fractional and continuously location-based, rather than being phase-based and intra-fractional. This occurs due to their prostate cancer data set, where geometric shifting potentially occurs between fractions of treatment delivery, due to organ relocation, size changes or other changes in patient physiology ranging from weight gain or loss, increased or decreased liquid consumption, etc. The region itself, however, is static during a given treatment session.

In their paper, they represent this uncertainty as a normally distributed uncertainty set around each element of the total dose-influence matrix for a given voxel at a certain selection of beam intensities $\mathbf{w}$, i.e., for a linear model, $\tilde{D}_v(\mathbf{w}) = \sum_{b \in \mathcal{B}} \tilde{D}_{v,b} w_b$. In this way, they assume $\tilde{D}_v(\mathbf{w}) \sim N(\mu, \sigma)$, and can neglect any phase-based uncertainty $\tilde{\mathbf{p}}_k$, that appears in the Chapter 3 models. They then approach this problem using [RO](#) methods.

The proposal in this section is that with the added flexibility of allowing each element of the matrix $\tilde{D}_{v,b}$ to operate separately, [DRO](#) would be a good candidate for extending the work of [Chu et al. \(2005\)](#). In that way, each element is represented as an independent, but not necessarily known distribution. Rather than assuming a fixed normal distribution, for example, the model could be extended to [DRO](#) by assuming an unknown distribution $\tilde{D}_v(\mathbf{w}) \sim N(\tilde{\mu}, \tilde{\sigma})$, or individual unknown distributions on each element of $\tilde{D}_{b,v} \sim N(\tilde{\mu}_{b,v}, \tilde{\sigma}_{b,v})$ as depicted in Figure 7.9 (or alternatively, to a range of potential distributions).

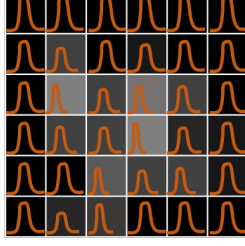


Figure 7.9: An uncertain $\tilde{D}_{v,b}$ matrix for inter-fractional motion.

A basic model may look as follows,

$$\min \sum_{v \in \mathcal{H}} \sum_{b \in \mathcal{B}} D_{v,b} w_b \quad (7.10)$$

$$\text{st. } \sum_{b \in \mathcal{B}} \mathbb{E}[\tilde{D}_{v,b} w_b] \geq \lambda \theta_v, \quad \forall v \in \mathcal{T}, \tilde{D}_{v,b} \sim N(\tilde{\mu}_{v,b}, \tilde{\sigma}_{v,b}), \quad (7.11)$$

$$w_b \geq 0 \quad b \in \mathcal{B}. \quad (7.12)$$

The λ is once again a constant denoting some ≤ 1 value that the system may deviate from the full dose in the case of uncertainty. Much like the constraints of the [RO](#) model may be replaced for each v by $\left\{ \min \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} D_{v,k,b} \hat{p}_{v,k} w_b \mid \hat{\mathbf{p}} \in \mathcal{U}_p \right\}$, the DRO constraints could be replaced by $\left\{ \min \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} \mathbb{E}[\hat{D}_{v,b}] w_b \mid \hat{D}_{v,b} \sim N(\tilde{\mu}_{v,b}, \tilde{\sigma}_{v,b}) \forall v, b \right\}$.

This model may then further include secondary moment functions, and be made into a [PDRO](#) model through the movement of the expectation to the constraints, and made light by loosening these constraints by ϵ .

7.4 Future Work in BAO

Chapter 4 provided a look at how one might choose beam angles when planning a radiation therapy treatment. There are a number of potential future directions. The first is an exploration into the tractability of heuristic approaches that do not require column generation. The goal of these heuristics are twofold, the first is to get the tools into hospitals. Clinicians do not generally have solver licenses, and this will prevent useful tools from becoming ubiquitous. The second is relaxing simplifications inherent in the models. Solving a column generation model is difficult and requires linear modelling.

A major concern of the radiation physicists that have seen the model presented in Chapter 4 is the lack of consideration of [percentage depth dose \(PDD\)](#)s. As radiation moves through the body, it attenuates, meaning the beam is typically less powerful further along the beam than when it enters the patient's body. This is demonstrated on a water phantom cube in Figure 7.10.

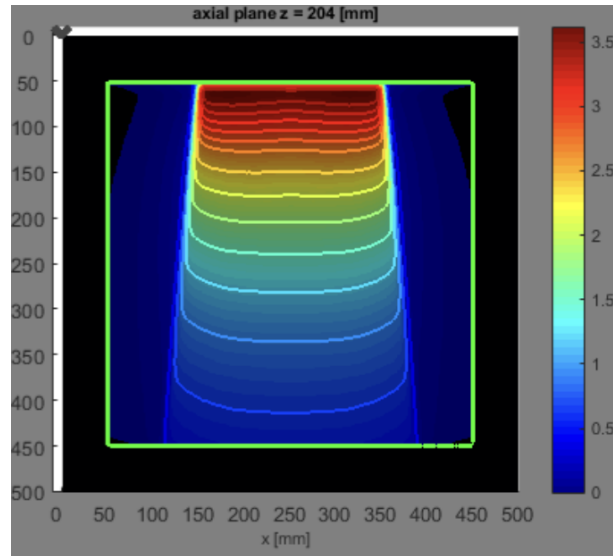


Figure 7.10: Water phantom PDD

Calibrating a dose-influence matrix generation algorithm to machine equipment typi-

cally relies on ensuring these PDDs match system output (which is calibrated against the actual linear accelerator in clinic). See Figure 7.11 for an attempted calibration of the CERR (Deasy et al., 2003) algorithm, vs. matRad’s algorithm (Wieser et al., 2017) in Figure 7.12.

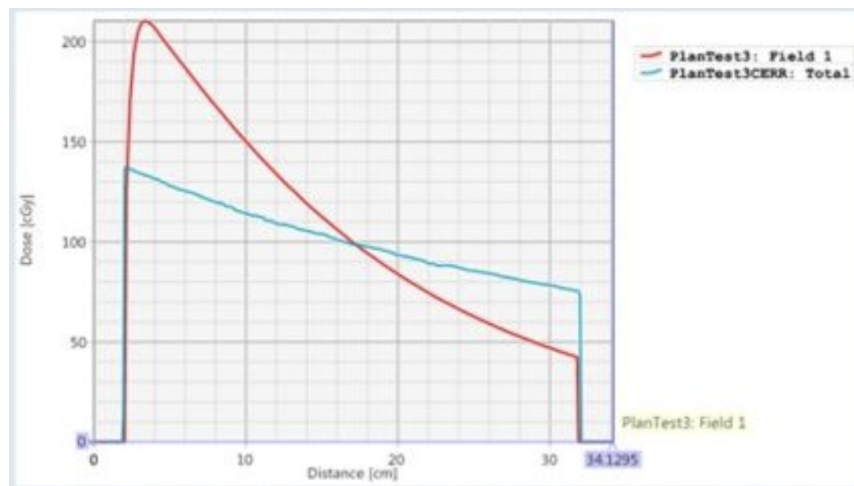


Figure 7.11: A PDD taken using CERR’s dose engine

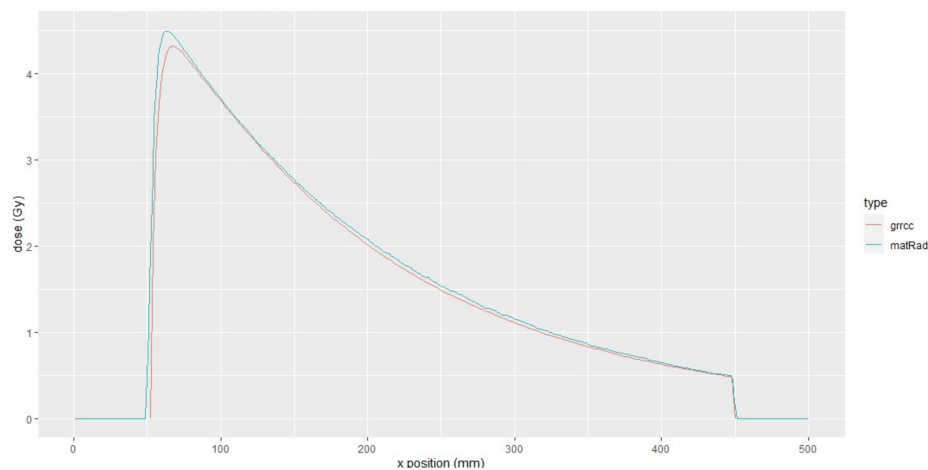


Figure 7.12: A PDD taken using matRad’s dose engine

A major takeaway from these images is that matRad matched far better than CERR

when calibrated, which is why it was integrated into our clinical pipeline (depicted in Figure 1.6 in the introduction). A second takeaway is that any fitting is better than no fitting. The current model would just be a straight horizontal line on both plots. Voxels are either dosed, or they are not dosed in the proposed models. However, in Figure 7.10, it can be seen that voxels towards the bottom of beam get nearly zero dose. This also does not account for any materials other than water that might be hit throughout the beam.

Ideally, if a fast heuristic is tractable on clinical equipment, it will also be tractable to integrate calibrated PDD information, and potentially even aggregates of density information that the beam might hit on the way into the body. Checking Algorithm 7 for speed and size and then simplifying it enough to add in as much dose and patient information as possible (since rough depth is already known at each intersection point) would be a major step towards making these methods useful for clinicians.

Other future steps include removing the point-voxel assumption (i.e., actually calculating what percent of each beamlet is present in a beam) could be accomplished by tracking voxel corner points instead of a single, central point. If the model performs well in the current setting, it could also be applied to other modalities of radiation treatment. These include choosing an optimal VMAT path, but also angle selection for Proton therapy and radiosurgery path selection, each of which differ from a more basic beam-angle selection. The proposed approaches may also be useful for more general 3D tracing problems, outside the sphere of RTTP, such as optimization problems in the fields of vision or biology.

7.5 Future Work in DAO

The DAO method has a considerable amount of implementation left before more future work is considered. That said, if the method performs well on the basic cases, integrating clinically relevant CVaR and RO constraints would be a good first step towards clinical viability.

7.6 Future Work in Sampling

The framework broken down in Chapter 6 is a good first step for choosing sampling methods, however, there is a lot more work that can be done in the future to evaluate the different approaches. First, a very large database of patients needs to be examined across multiple approaches to choose a single “best” method. Then it is highly possible that different approaches should be considered for different regions of the body or classes of cases.

Best practices should also be established in terms of non-uniform sampling, multiple organs at risk, or multiple CTV locations. Different objectives, constraints and steps of the RTTP process could also be examined across the different methods. Certain sampling methods may work better with more realistic constraints, such as CVaR and others might work better for methods that include RO considerations.

Finally, the proposed methods could each be independently optimized and examined in more detail. For example, the beamlet based approaches as implemented naively run through all beamlet locations until sufficient voxels have been added to the set of voxels under considerations. This means if an iteration chooses fewer than the number of beamlet-clusters available, they are not properly distributed throughout the beam. Improvements that address this shortcoming could improve the heuristic under more sparse sampling conditions without any meaningful downsides. The k -means approaches could also be parallelized and implemented in a faster language for improved convergence.

Taken together, these investigations would provide an even more thorough understanding of the impacts of voxel down-sampling on the RTTP problem space.

Chapter 8

Conclusions

The thesis presented a number of different topics in the field of radiation therapy treatment planning, each with unique challenges and problem structures. New ways to think about old problems were proposed, from conceptualizing beam angle positioning as a geometry problem, to considering the full planning process as a decision diagram. It also presented new perspectives on uncertainty, both from an implementation perspective, as well as plan quality assessment and trade-off tool.

Some major takeaways are as follows:

1. Uncertainty in optimization can be used to understand trade-offs and mitigate reasonable risk
2. Naive methods for [DAO](#) are slow and outdated, but there is promise for efficient methods that do not compromise treatment quality
3. Speeding up radiation therapy will require holistic views of many seemingly-unrelated [OR](#) problems

There is a lot of work to be done in order to make optimization in [RTTP](#) a viable staple in the industry. The planning process is largely an iterative one that requires many

re-optimizations and accurate methods for getting salient information and visualizing key trade-offs. In order to integrate more [OR](#) into these methods, the individual components must run faster and more memory-efficiently than ever before. In that vein, more studies, more cases, and faster models must continue to be developed in order to build to a gold standard of fast collaboration with experts.

In terms of next steps, integrating the work done so far into clinical systems for more comprehensive testing, and the unearthing of even more new projects is essential. Because the [RTTP](#) process is so large and encompasses so many steps, as previewed in the future work section, each project can easily lead to several more complex follow-up projects, which is why the core approaches must be clear and extensible. This thesis hypothesizes that a fast method which integrates all three major steps of the [RTTP](#) process will be built on efficient building blocks across each of the individual steps. Establishing a baseline of these smaller models that work well with open-source databases such as the CORT data instances ([Craft et al., 2014](#)) would act as a useful proof-of-concept for the broader radiation therapy community.

More large-scale directions for the presented research also includes extending the work done to allow for motion in [VMAT](#) and other popular systems, in order to keep up with the ever-evolving modern treatment standards. The integration of the current procedure and open sourcing of the proposed models into systems like Portpy ([Jhanwar et al., 2023](#)), is also crucial to involve the marketplace of ideas in new projects. On the other hand, specializing the models to work as a resource for individual procedures in specific cancer centres, as in the pipeline proposed in this document, may also be important for increasing the demand for optimal systems at a hospital level. Including highly visual explanations and decision support tools will ultimately be critical for one day formally integrating these tools into official treatment planning systems.

References

- Ravindra K Ahuja and James B Orlin. Inverse optimization. *Operations Research*, 49(5): 771–783, 2001.
- Dionne M Aleman, H Edwin Romeijn, and James F Dempsey. A response surface approach to beam orientation optimization in intensity-modulated radiation therapy treatment planning. *INFORMS Journal on Computing*, 21(1):62–76, 2009.
- Guy Amit, Thomas G Purdie, Alex Levinshtein, Andrew J Hope, Patricia Lindsay, Andrea Marshall, David A Jaffray, and Vladimir Pekar. Automatic learning-based beam angle selection for thoracic imrt. *Medical physics*, 42(4):1992–2005, 2015.
- James M Balter, Kwok L Lam, Cornealeus J McGinn, Theodore S Lawrence, and Randall K Ten Haken. Improvement of ct-based treatment-planning models of abdominal targets using static exhale imaging. *International Journal of Radiation Oncology* Biology* Physics*, 41(4):939–943, 1998.
- Walid Ben-Ameur and José Neto. Acceleration of cutting-plane and column generation algorithms: Applications to network design. *Networks: An International Journal*, 49(1): 3–17, 2007.
- Aharon Ben-Tal and Arkadi Nemirovski. Robust solutions of uncertain linear programs. *Operations research letters*, 25(1):1–13, 1999.
- Carmen Bergom, Julie A Bradley, Andrea K Ng, Pamela Samson, Clifford Robinson, Juan Lopez-Mattei, and Joshua D Mitchell. Past, present, and future of radiation-induced cardiotoxicity: refinements in targeting, surveillance, and risk stratification. *Cardio Oncology*, 3(3):343–359, 2021.
- Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations research*, 52(1): 35–53, 2004.

- Dimitris Bertsimas, David B Brown, and Constantine Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, 2011.
- Dimitris Bertsimas, Valentina Cacchiani, David Craft, and Omid Nohadani. A hybrid approach to beam angle optimization in intensity-modulated radiation therapy. *Computers & Operations Research*, 40(9):2187–2197, 2013.
- Dimitris Bertsimas, Iain Dunning, and Miles Lubin. Reformulation versus cutting-planes for robust optimization. *Computational Management Science*, 13(2):195–217, 2016.
- Michelle Böck, Kjell Eriksson, Anders Forsgren, and Björn Hårdemark. Toward robust adaptive radiation therapy strategies. *Medical physics*, 44(6):2054–2065, 2017.
- Natashia Boland, Horst W Hamacher, and Frank Lenzen. Minimizing beam-on time in cancer radiation treatment using multileaf collimators. *Networks*, 43(4):226–240, 2004.
- Thomas Bortfeld. IMRT: a review and preview. *Physics in Medicine & Biology*, 51(13):R363, 2006.
- Thomas Bortfeld, Timothy CY Chan, Alexei Trofimov, and John N Tsitsiklis. Robust management of motion uncertainty in intensity-modulated radiation therapy. *Operations Research*, 56(6):1461–1473, 2008.
- Maria Broderick, Michelle Leech, and Mary Coffey. Direct aperture optimization as a means of reducing the complexity of intensity modulated radiation therapy plans. *Radiation Oncology*, 4(1):8, 2009.
- Valentina Cacchiani and Paolo Toth. Nominal and robust train timetabling problems. *European Journal of Operational Research*, 219(3):727–737, 2012.
- Wenhua Cao, Gino J Lim, Andrew Lee, Yupeng Li, Wei Liu, X Ronald Zhu, and Xiaodong Zhang. Uncertainty incorporated beam angle optimization for impt treatment planning. *Medical physics*, 39(8):5248–5256, 2012.
- Fredrik Carlsson. Combining segment generation with direct step-and-shoot optimization in intensity-modulated radiation therapy. *Medical Physics*, 35(9):3828–3838, 2008.
- Timothy CY Chan, Thomas Bortfeld, and John N Tsitsiklis. A robust approach to IMRT optimization. *Physics in Medicine & Biology*, 51(10):2567, 2006.

- Timothy CY Chan, Tim Craig, Taewoo Lee, and Michael B Sharpe. Generalized inverse multiobjective optimization with application to cancer therapy. *Operations Research*, 62(3):680–695, 2014a.
- Timothy CY Chan, Houra Mahmoudzadeh, and Thomas G Purdie. A robust-cvar optimization approach with application to breast cancer therapy. *European Journal of Operational Research*, 238(3):876–885, 2014b.
- Timothy CY Chan, Taewoo Lee, and Daria Terekhov. Inverse optimization: Closed-form solutions, geometry, and goodness of fit. *Management Science*, 65(3):1115–1135, 2019.
- Ying Chen, Zhihong Yuan, and Bingzhen Chen. Process optimization with consideration of uncertainties—an overview. *Chinese journal of chemical engineering*, 26(8):1700–1706, 2018.
- Byungchul Cho. Intensity-modulated radiation therapy: a review with a physics perspective. *Radiation oncology journal*, 36(1):1, 2018.
- Millie Chu, Yuriy Zinchenko, Shane G Henderson, and Michael B Sharpe. Robust optimization for intensity modulated radiation therapy treatment planning under uncertainty. *Physics in Medicine & Biology*, 50(23):5463, 2005.
- IBM ILOG Cplex. V12.7: User’s manual for cplex. *User’s Manual (IBM ILOG CPLEX Division, Incline Village, NV)*, 2017.
- David Craft, Mark Bangert, Troy Long, Dávid Papp, and Jan Unkelbach. Shared data for intensity modulated radiation therapy (IMRT) optimization research: the cort dataset. *GigaScience*, 3(1):37, 2014.
- Chenxi Dai, Lei Wu, and Hongyu Wu. A multi-band uncertainty set based robust scuc with spatial and temporal budget constraints. *IEEE Transactions on Power Systems*, 31(6):4988–5000, 2016.
- Joseph O Deasy, Angel I Blanco, and Vanessa H Clark. Cerr: a computational environment for radiotherapy research. *Medical physics*, 30(5):979–985, 2003.
- Milad Dehghani Filabadi and Houra Mahmoudzadeh. Effective budget of uncertainty for classes of robust optimization. *INFORMS Journal on Optimization*, 2022.
- Erick Delage and Yinyu Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations research*, 58(3):595–612, 2010.

- James F Dempsey, H Edwin Romeijn, Jonathan G Li, Daniel A Low, and Jatinder R Palta. A fourier analysis of the dose grid resolution required for accurate imrt fluence map optimization. *Medical physics*, 32(2):380–388, 2005.
- Joana Dias, Humberto Rocha, Brígida Ferreira, and Maria do Carmo Lopes. A genetic algorithm with neural network fitness function evaluation for imrt beam angle optimization. *Central European Journal of Operations Research*, 22(3):431–455, 2014.
- Tesering Dorje. Limitation of pencil beam convolution (pbc) algorithm for photon dose calculation in inhomogeneous medium. *J Cancer Treat Res*, 2(1):1–4, 2014.
- Matthias Ehrgott, Çiğdem Güler, Horst W Hamacher, and Lizhen Shao. Mathematical optimization in intensity modulated radiation therapy. *Annals of Operations Research*, 175(1):309–365, 2010.
- Matteo Fischetti and Michele Monaci. Light robustness. In *Robust and online large-scale optimization*, pages 61–84. Springer, 2009.
- Matteo Fischetti and Michele Monaci. Cutting plane versus compact formulations for uncertain (integer) linear programs. *Mathematical Programming Computation*, 4(3):239–273, 2012.
- Matteo Fischetti and Domenico Salvagnin. An in-out approach to disjunctive optimization. In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems: 7th International Conference, CPAIOR 2010, Bologna, Italy, June 14-18, 2010. Proceedings 7*, pages 136–140. Springer, 2010.
- Lucy Fountain, Kourosh Khedrilliraviasl, Sina Mahmoudzadeh, and Houra Mahmoudzadeh. Dose-based constraint generation for large-scale imrt optimization. *INFOR: Information Systems and Operational Research*, 60(1):1–19, 2022.
- Haris Gavranović and Mirsad Buljubašić. Efficient robust linear optimization for large repositioning problems. In *International Conference on Network Optimization*, pages 553–558. Springer, 2011.
- Ralph E Gomory. An algorithm for integer solutions to linear programs. *Recent advances in mathematical programming*, 64(260-302):14, 1963.
- Bram L Gorissen, İhsan Yanıkoğlu, and Dick den Hertog. A practical guide to robust optimization. *Omega*, 53:124–137, 2015.

- Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2024. URL <https://www.gurobi.com>.
- Mostafa Hasan, İ Esra Büyüктаhtakın, and Elshami Elamin. A multi-criteria ranking algorithm (mcra) for determining breast cancer therapy. *Omega*, 82:83–101, 2019.
- Coen Hurkmans, Tommy Knöös, Per Nilsson, Gudrun Svahn-Tapper, and Hans Danielsson. Limitations of a pencil beam approach to photon dose calculations in the head and neck region. *Radiotherapy and Oncology*, 37(1):74–80, 1995.
- Dan A Iancu and Nikolaos Trichakis. Pareto efficiency in robust optimization. *Management Science*, 60(1):130–147, 2014.
- Gourav Jhanwar, Mojtaba Tefagh, Vicki T Taasti, Sadegh R Alam, Seppo Tuomaala, Saad Nadeem, and Masoud Zarepisheh. Portpy: An open-source python package for planning and optimization in radiation therapy including benchmark data and algorithms. *AAPM 65th Annual Meeting & Exhibition*, 2023.
- Noriyuki Kadoya, Yujiro Nakajima, Masahide Saito, Yuki Miyabe, Masahiko Kurooka, Satoshi Kito, Yukio Fujita, Motoharu Sasaki, Kazuhiro Arai, Kensuke Tani, et al. Multi-institutional validation study of commercially available deformable image registration software for thoracic images. *International Journal of Radiation Oncology* Biology* Physics*, 96(2):422–431, 2016.
- Gereon Kåver, Bengt K Lind, Johan Löf, Anders Liander, and Anders Brahme. Stochastic optimization of intensity modulated radiotherapy to account for uncertainties in patient sensitivity. *Physics in Medicine & Biology*, 44(12):2955, 1999.
- James E Kelley, Jr. The cutting-plane method for solving convex programs. *Journal of the society for Industrial and Applied Mathematics*, 8(4):703–712, 1960.
- Eva K Lee, Tim Fox, and Ian Crocker. Integer programming applied to intensity-modulated radiation therapy treatment planning. *Annals of Operations Research*, 119(1-4):165–181, 2003.
- Yongjie Li, Jonathan Yao, and Dezhong Yao. Genetic algorithm based deliverable segments optimization for static intensity-modulated radiotherapy. *Physics in Medicine & Biology*, 48(20):3353, 2003.
- Yongjie Li, Jonathan Yao, and Dezhong Yao. Automatic beam angle selection in imrt planning using genetic algorithm. *Physics in Medicine & Biology*, 49(10):1915, 2004.

- Hsiao-Ming Lu, Hanne M Kooy, Zach H Leber, and Robert J Ledoux. Optimized beam planning for linear accelerator-based stereotactic radiosurgery. *International journal of radiation oncology, biology, physics*, 39(5):1183–1189, 1997.
- Miles Lubin, Oscar Dowson, Joaquim Dias Garcia, Joey Huchette, Benoît Legat, and Juan Pablo Vielma. JuMP 1.0: Recent improvements to a modeling language for mathematical optimization. *Mathematical Programming Computation*, 2023. doi: 10.1007/s12532-023-00239-3.
- Houra Mahmoudzadeh. *Robust Optimization Methods for Breast Cancer Radiation Therapy*. PhD thesis, University of Toronto, 2015.
- Houra Mahmoudzadeh, Jenny Lee, Timothy CY Chan, and Thomas G Purdie. Robust optimization methods for cardiac sparing in tangential breast IMRT. *Medical Physics*, 42(5):2212–2222, 2015.
- Houra Mahmoudzadeh, Thomas G Purdie, and Timothy CY Chan. Constraint generation methods for robust optimization in radiation therapy. *Operations research for health care*, 8:85–90, 2016.
- Houra Mahmoudzadeh, Akram Mirahmadi Shalamzari, and Hossein Abouee-Mehrizi. Robust multi-class multi-period patient scheduling with wait time targets. *Operations Research for Health Care*, 25:100254, 2020.
- Mehdi Mahnam, Michel Gendreau, Nadia Lahrichi, and Louis-Martin Rousseau. Simultaneous delivery time and aperture shape optimization for the volumetric-modulated arc therapy (vmat) treatment planning problem. *Physics in Medicine & Biology*, 62(14):5589, 2017.
- Benjamin C Martin, Thomas R Bortfeld, and David A Castanon. Accelerating imrt optimization by voxel sampling. *Physics in Medicine & Biology*, 52(24):7211, 2007.
- MATLAB. *version 9.12.0 (R2022a)*. The MathWorks Inc., Natick, Massachusetts, 2022.
- Chunhua Men, H Edwin Romeijn, Z Caner Taşkın, and James F Dempsey. An exact approach to direct aperture optimization in IMRT treatment planning. *Physics in Medicine & Biology*, 52(24):7333, 2007.
- Marie-Pierre Milette. *Direct optimization of 3D dose distributions using collimator rotation*. PhD thesis, University of British Columbia, 2008.

- Almir Mutapcic and Stephen Boyd. Cutting-set methods for robust convex optimization with pessimizing oracles. *Optimization Methods & Software*, 24(3):381–406, 2009.
- Daryl P Nazareth, Stephen Brunner, Matthew D Jones, Harish K Malhotra, and Mohammad Bakhtiari. Optimization of beam angles for intensity modulated radiation therapy treatment planning using genetic algorithm on a distributed computing platform. *Journal of Medical Physics*, 34(3):129–132, 2009.
- Omid Nohadani and Arkajyoti Roy. Robust optimization with time-dependent uncertainty in radiation therapy. *IIEE Transactions on Healthcare Systems Engineering*, 7(2):81–92, 2017.
- Arinbjörn Olafsson and Stephen J Wright. Efficient schemes for robust IMRT treatment planning. *Physics in Medicine & Biology*, 51(21):5621, 2006.
- Bharath Pandu, Rajadurai Elavarasan, Vivek TR, Saro Jacob, Goutham Sunny, Hima Nainan, Tatarao Maddipati, et al. Dosimetric comparison and plan evaluation of different dose computing algorithms for different radiotherapy techniques in head and neck tumors. *Iranian Journal of Medical Physics*, 19(6):346–355, 2022.
- Fei Peng, Xun Jia, Xuejun Gu, Marina A Epelman, H Edwin Romeijn, and Steve B Jiang. A new column-generation-based algorithm for vmat treatment plan optimization. *Physics in Medicine & Biology*, 57(14):4569, 2012.
- Artur Pessoa, Ruslan Sadykov, Eduardo Uchoa, and Francois Vanderbeck. In-out separation and column generation stabilization by dual price smoothing. In *Experimental Algorithms: 12th International Symposium, SEA 2013, Rome, Italy, June 5-7, 2013. Proceedings 12*, pages 354–365. Springer, 2013.
- Laurent Poirrier and James Yu. On the depth of cutting planes. *arXiv preprint arXiv:1903.05304*, 2019.
- Gert-Jaap Polinder, Thomas Breugem, Twan Dollevoet, and Gábor Maróti. An adjustable robust optimization approach for periodic timetabling. *Transportation Research Part B: Methodological*, 128:50–68, 2019.
- Thomas G Purdie, Robert E Dinniwell, Daniel Letourneau, Christine Hill, and Michael B Sharpe. Automated planning of tangential breast intensity-modulated radiotherapy using heuristic optimization. *International Journal of Radiation Oncology* Biology* Physics*, 81(2):575–583, 2011.

- Thomas G Purdie, Robert E Dinniwell, Anthony Fyles, and Michael B Sharpe. Automation and intensity modulated radiation therapy for individualized high-quality tangent breast treatment plans. *International Journal of Radiation Oncology* Biology* Physics*, 90(3): 688–695, 2014.
- Rahimi, Fahimeh. *Reducing Conservatism in Pareto Robust Optimization*. PhD thesis, University of Waterloo, 2022. URL <http://hdl.handle.net/10012/18360>.
- RaySearch Laboratories. Deformable registration in raystation. <https://www.raysearchlabs.com/49e915/siteassets/about-overview/media-center/wp-re-ev-n-pdfs/white-papers/deformable-registration-in-raystation2.pdf>, 2017. Accessed: 2021-10-13.
- Danielle Ripsman. Robust direct aperture optimization methods for cardiac sparing in left-sided breast cancer radiation therapy. Master’s thesis, University of Waterloo, 2018.
- Danielle A Ripsman, Dionne M Aleman, and Kimia Ghobadi. Interactive visual guidance for automated stereotactic radiosurgery treatment planning. *Expert Systems with Applications*, 42(21):8337–8348, 2015.
- Danielle A Ripsman, Thomas G Purdie, Timothy CY Chan, and Houra Mahmoudzadeh. Robust direct aperture optimization for radiation therapy treatment planning. *INFORMS Journal on Computing*, 34(4):2017–2038, 2022. doi: <https://doi.org/10.1287/ijoc.2022.1167>.
- Danielle A Ripsman, Fahimeh Rahimi, Hossein Abouee-Mehrizi, and Houra Mahmoudzadeh. Light pareto robust optimization for imrt treatment planning. *Medical Physics*, 50(5):2637–2648, 2023.
- Humberto Rocha, Joana M Dias, Brigida C Ferreira, and Maria do Carmo Lopes. Influence of sampling in radiation therapy treatment design. In *International Conference on Computational Science and Its Applications*, pages 215–230. Springer, 2011.
- Humberto Rocha, Joana Matos Dias, Tiago Ventura, Brígida da Costa Ferreira, and Maria do Carmo Lopes. Beam angle optimization in imrt: are we really optimizing what matters? *International Transactions in Operational Research*, 26(3):908–928, 2019.
- H Edwin Romeijn, Ravindra K Ahuja, James F Dempsey, Arvind Kumar, and Jonathan G Li. A novel linear programming approach to fluence map optimization for intensity modulated radiation therapy treatment planning. *Physics in Medicine & Biology*, 48(21):3521, 2003.

- H Edwin Romeijn, Ravindra K Ahuja, James F Dempsey, and Arvind Kumar. A column generation approach to radiation therapy treatment planning using aperture modulation. *SIAM Journal on Optimization*, 15(3):838–862, 2005.
- H Edwin Romeijn, Ravindra K Ahuja, James F Dempsey, and Arvind Kumar. A new linear programming approach to radiation therapy treatment planning problems. *Operations Research*, 54(2):201–216, 2006.
- H.E. Romeijn and J.F. Dempsey. Intensity modulated radiation therapy treatment plan optimization. *TOP*, 16(2):215–243, 2008.
- Ouided Rouabhi, Brandie Gross, John Bayouth, and Junyi Xia. The dosimetric and temporal effects of respiratory-gated, high-dose-rate radiation therapy in patients with lung cancer. *Technology in Cancer Research & Treatment*, 18:1533033818816072, 2019.
- Arkajyoti Roy, Shaunak S Dabadghao, and Ahmadreza Marandi. Value of intermediate imaging in adaptive robust radiotherapy planning to manage radioresistance. *Annals of Operations Research*, pages 1–22, 2022.
- Isaac Rudich, Quentin Cappart, and Louis-Martin Rousseau. Improved peel-and-bound: Methods for generating dual bounds with multivalued decision diagrams. *Journal of Artificial Intelligence Research*, 77:1489–1538, 2023.
- Azar Sadeghnejad-Barkousaraie, Gyanendra Bohara, Steve Jiang, and Dan Nguyen. A reinforcement learning application of a guided monte carlo tree search algorithm for beam orientation selection in radiation therapy. *Machine learning: science and technology*, 2(3):035013, 2021.
- Ehsan Salari and Jan Unkelbach. A column-generation-based method for multi-criteria direct aperture optimization. *Physics in Medicine & Biology*, 58(3):621, 2013.
- Alexander Scherrer, Karl-Heinz Küfer, Thomas Bortfeld, Michael Monz, and Fernando Alonso. Imrt planning on adaptive volume structures—a decisive reduction in computational complexity. *Physics in Medicine & Biology*, 50(9):2033, 2005.
- Anita Schöbel. Generalized light robustness and the trade-off between robustness and nominal quality. *Mathematical Methods of Operations Research*, 80(2):161–191, 2014.
- David M Shepard, MA Earl, XA Li, S Naqvi, and Cedric Yu. Direct aperture optimization: A turnkey solution for step-and-shoot IMRT. *Medical Physics*, 29(6):1007–1018, 2002.

- Jens Sjölund, Stella Riad, Marcus Hennix, and Håkan Nordström. A linear programming approach to inverse planning in gamma knife radiosurgery. *Medical physics*, 46(4):1533–1544, 2019.
- S Soderstrom and A Brahme. Selection of suitable beam orientations in radiation therapy using entropy and fourier transform measures. *Physics in Medicine & Biology*, 37(4):911, 1992.
- Allen L Soyster. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations research*, 21(5):1154–1157, 1973.
- SCM Ten Eikelder, Ali Ajdari, Thomas Bortfeld, and Dick den Hertog. Adjustable robust treatment-length optimization in radiation therapy. *Optimization and Engineering*, 23(4):1949–1986, 2022.
- The Johns Hopkins University, The Johns Hopkins Hospital and Johns Hopkins Health System. Ct lung screening, 2024. URL <https://www.hopkinsmedicine.org/imaging/exams-and-procedures/screenings/ct-lung>. All rights reserved.
- Paras Babu Tiwari. Automating intensity modulated radiation therapy treatment planning by using hierarchical optimization. 2015.
- Jan Unkelbach, Markus Alber, Mark Bangert, Rasmus Bokrantz, Timothy CY Chan, Joseph O Deasy, Albin Fredriksson, Bram L Gorissen, Marcel Van Herk, Wei Liu, et al. Robust radiotherapy planning. *Physics in Medicine & Biology*, 63(22):22TR02, 2018.
- Lisanne V van Dijk, Lisa Van den Bosch, Paul Aljabar, Devis Peressutti, Stefan Both, Roel JHM Steenbakkers, Johannes A Langendijk, Mark J Gooding, and Charlotte L Brouwer. Improving automatic delineation for head and neck organs at risk by deep learning contouring. *Radiotherapy and Oncology*, 142:115–123, 2020.
- Varian Medical Systems, Inc. Image gallery, 2024. URL <https://www.varian.com/about-varian/newsroom/image-gallery>. All rights reserved.
- Steve Webb. *Intensity-modulated radiation therapy*. CRC Press, 2015.
- Ola Weistrand and Stina Svensson. The anaconda algorithm for deformable image registration in radiotherapy. *Medical physics*, 42(1):40–53, 2015.
- Hans-Peter Wieser, Eduardo Cisternas, Niklas Wahl, Silke Ulrich, Alexander Stadler, Henning Mescher, Lucas-Raphael Müller, Thomas Klinge, Hubert Gabrys, Lucas Burigo,

- et al. Development of the open-source dose calculation and optimization toolkit matrad. *Medical Physics*, 44(6):2556–2568, 2017. doi: 10.1002/mp.12251.
- Chiwei Yan and Jerry Kung. Robust aircraft routing. *Transportation Science*, 52(1): 118–133, 2018.
- Masoud Zarepisheh, Troy Long, Nan Li, Zhen Tian, H Edwin Romeijn, Xun Jia, and Steve B Jiang. A dvh-guided imrt optimization algorithm for automatic treatment planning and adaptive radiotherapy replanning. *Medical physics*, 41(6Part1):061711, 2014.