

Constraint Inference using Quality-based Robust-inverse Optimization for Radiation Therapy

by

Bradley Thomas Morley Hallett

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Management Sciences

Waterloo, Ontario, Canada, 2024

© Bradley Thomas Morley Hallett 2024

Author's Declaration

I hereby declare that I am the sole author of this 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.

Abstract

Traditional radiation therapy treatment planning involves manual, iterative adjustments to optimization model parameters, often guided by the oncologist’s judgment. This process can introduce implicit constraints that may not be obvious to treatment planners. One approach to infer the parameters of a problem from observed solutions is inverse optimization, which has been extensively studied in the context of radiation therapy planning. While traditional forward optimization relies on predefined problem parameters to find an optimal solution, inverse optimization seeks to determine these parameters based on one or multiple observations that may be optimal, feasible, or infeasible. When dealing with uncertain or noisy data, robust optimization can be employed to address uncertainty. Similarly, extensive research has been conducted on the application of robust optimization techniques to radiation therapy planning. This thesis introduces an inverse optimization approach that infers the complete constraint matrix of a linear forward optimization problem while considering multiple uncertain feasible observations. We discuss application-specific considerations for applying inverse optimization to radiation therapy treatment plans and present a formulation that adapts to uncertainty using a polyhedral uncertainty set. We propose an iterative single-constraint inference algorithm that demonstrates superior performance in terms of computational efficiency and the quality of inferred constraints compared to existing methods in the literature. The proposed algorithm is then evaluated on a dataset of prostate cancer treatment plans, showcasing its practical applicability in medical treatment planning. By leveraging the results from this algorithm, we aim to streamline the development of radiation therapy treatment plans by providing guidelines that better reflect the oncologist’s preferences, reducing the number of required iterations and conserving valuable resources such as time and cost.

Acknowledgments

I would like to express my deepest gratitude to all those who supported me throughout my master's journey.

I extend my heartfelt thanks to my supervisor, Professor Houra Mahmoudzadeh, for her invaluable guidance, encouragement, and patience throughout my graduate studies. Her insights and feedback were crucial in shaping this thesis.

I am grateful to Dr. Ernest Osei and Dr. Johnson Darko at the Grand River Regional Cancer Centre for their enthusiasm towards my research and for generously sharing their knowledge with me.

I would like to thank my committee members, Professor Safa Erenay and Professor Mehrdad Pirnia, for taking the time to review my thesis and providing me with valuable feedback.

I want to give special thanks to my colleagues and friends at UW, especially DR and SH, for their camaraderie, support, and stimulating discussions.

To my friends, RT, PP, and AG, thank you for the many moments of laughter that kept me grounded and focused.

I am deeply indebted to my family for their unconditional love, support, and understanding, SH, JH, JH, and JH. To my fiancée, Chelsea Jacob, your love, patience, and unwavering support have been a source of strength throughout this journey.

Thank you all.

Table of Contents

Author's Declaration	ii
Abstract	iii
Acknowledgments	iv
List of Figures	ix
List of Tables	xv
List of Abbreviations	xvi
1 Introduction	1
2 Literature Review	5
2.1 Inverse optimization background	6
2.2 Parameters inferred through inverse optimization	6
2.3 Structure of the forward problem	8

2.4	Characteristics of the data	10
2.5	Robust optimization	11
2.6	Robust-inverse optimization	12
2.7	Contributions	13
3	Data Description and Analysis	15
3.1	Existing dose constraints	16
3.2	Radiation therapy treatment plan quality	22
3.3	Data uncertainty	25
3.4	Summary	27
4	Inverse Optimization for Radiation Therapy Guidelines	28
4.1	Inverse optimization background	29
4.1.1	Forward optimization problem	29
4.1.2	Inverse optimization problem	31
4.2	Pairwise feature solution approach	33
4.3	Feature selection	36
4.4	Outlier detection	41
4.5	Preventing elimination of high-quality plans	46
4.6	Summary	49

5	Proposed Quality-based Robust-inverse Optimization Formulation	50
5.1	Loss functions	51
5.2	Proposed Robust-inverse optimization formulation	52
5.2.1	Robust-inverse optimization implementation	56
5.3	Quality-based robust-inverse optimization formulation	59
5.4	Summary	62
6	Fairness Measure Formulation	64
6.1	Formulation	64
6.2	Results	69
6.3	Discussion	72
7	Compactness Measure Formulation	74
7.1	Formulation	74
7.2	Results	76
7.3	Discussion	79
8	Proposed Iterative Single-constraint Inference Algorithm	81
8.1	Formulation	82
8.2	Results	90
8.3	Discussion	93
8.4	Summary	94

9 Discussion	95
10 Conclusions	101
References	105
APPENDICES	114
A The Complete Set of Constraints Inferred	115

List of Figures

1.1	The Radiation Therapy (RT) treatment planning process	2
3.1	Box and whisker plots of the features from the planning objectives (the planning objectives are in green)	20
3.2	Two plan features with potential outliers (red) and a potential implicit trade-off constraint (green)	21
3.3	Box and whisker plots of the HI, UI and CI values	23
3.4	Box and whisker plot of the RT treatment plan quality values	24
4.1	An example of the visualization of a two-dimensional constraint	34
4.2	A three-dimensional space separated into three two-dimensional spaces . .	35
4.3	Correlation matrix heatmap of RT treatment plan features	38
4.4	Rectum - V20Gy (%) vs. Bladder - V25(%) with a potential trade-off constraint	39
4.5	The feature pairs selected (blue) and not selected (red)	40
4.6	Bladder - V 5000 cGy (%) vs. Rectum - V 5000 cGy (%) scatter plot with potential outliers indicated by the red arrows	42

4.7	Bladder - V 5000 cGy (%) vs. Rectum - V 5000 cGy (%) scatter plot with outliers (red)	45
4.8	An example of preferred constraints (green), undesirable constraints (orange), and two hypothetical observations (p_2 and p_4) to motivate corner restricting constraints	46
5.1	An example of a two-dimensional problem with box uncertainty	54
5.2	An example of a two-dimensional problem with the box uncertainty corner points as observations and potential inferred constraints	57
5.3	Inferred constraints without uncertainty (black) and with constant uncertainty size (green)	60
5.4	Inferred constraints without uncertainty (black) and with quality-based uncertainty size (green)	62
6.1	An example of the F-e-RIO formulation result on a single pair of features	71
6.2	An example of an odd inferred constraint produced by the F-e-RIO formulation	72
7.1	An example of the C-e-RIO formulation result when inferring a single constraint	78
7.2	An example of the C-e-RIO formulation result when inferring two constraints	79
8.1	An example problem for the iterative single-constraint inference algorithm	87
8.2	A visualization of the iterative single-constraint inference algorithm on the example problem	88

8.3	A comparison of the solutions from the proposed algorithm and the F-e-RIO formulation	92
9.1	PTV - V 6000 cGy (%) vs. Rectum - V 4600 cGy (%) with inferred constraints (green) and current guidelines (orange)	98
A.1	Bladder - V 5000 cGy (%) vs. Rectum - V 3850 cGy (%)	116
A.2	Bladder - V 5000 cGy (%) vs. Rectum - V 4000 cGy (%)	116
A.3	Bladder - V 5000 cGy (%) vs. Rectum - V 4600 cGy (%)	116
A.4	Bladder - V 5000 cGy (%) vs. Rectum - V 5000 cGy (%)	116
A.5	Bladder - V 5000 cGy (%) vs. Rectum - V 5380 cGy (%)	117
A.6	Bladder - V 5000 cGy (%) vs. Rectum - V 5770 cGy (%)	117
A.7	Bladder - V 5000 cGy (%) vs. Rectum - V 6000 cGy (%)	117
A.8	Bladder - V 5380 cGy (%) vs. Rectum - V 3850 cGy (%)	117
A.9	Bladder - V 5380 cGy (%) vs. Rectum - V 4000 cGy (%)	118
A.10	Bladder - V 5380 cGy (%) vs. Rectum - V 4600 cGy (%)	118
A.11	Bladder - V 5380 cGy (%) vs. Rectum - V 5000 cGy (%)	118
A.12	Bladder - V 5380 cGy (%) vs. Rectum - V 5380 cGy (%)	118
A.13	Bladder - V 5380 cGy (%) vs. Rectum - V 5770 cGy (%)	119
A.14	Bladder - V 5380 cGy (%) vs. Rectum - V 6000 cGy (%)	119
A.15	Bladder - V 5770 cGy (%) vs. Rectum - V 3850 cGy (%)	119
A.16	Bladder - V 5770 cGy (%) vs. Rectum - V 4000 cGy (%)	119

A.17 Bladder - V 5770 cGy (%) vs. Rectum - V 4600 cGy (%)	120
A.18 Bladder - V 5770 cGy (%) vs. Rectum - V 5000 cGy (%)	120
A.19 Bladder - V 5770 cGy (%) vs. Rectum - V 5380 cGy (%)	120
A.20 Bladder - V 5770 cGy (%) vs. Rectum - V 5770 cGy (%)	120
A.21 Bladder - V 5770 cGy (%) vs. Rectum - V 6000 cGy (%)	121
A.22 CTV - D99% (%) $\geq$ 99% vs. Bladder - V 5000 cGy (%)	121
A.23 CTV - D99% (%) $\geq$ 99% vs. Bladder - V 5380 cGy (%)	121
A.24 CTV - D99% (%) $\geq$ 99% vs. Bladder - V 5770 cGy (%)	121
A.25 CTV - D99% (%) $\geq$ 99% vs. Rectum - V 3850 cGy (%)	122
A.26 CTV - D99% (%) $\geq$ 99% vs. Rectum - V 4000 cGy (%)	122
A.27 CTV - D99% (%) $\geq$ 99% vs. Rectum - V 4600 cGy (%)	122
A.28 CTV - D99% (%) $\geq$ 99% vs. Rectum - V 5000 cGy (%)	122
A.29 CTV - D99% (%) $\geq$ 99% vs. Rectum - V 5380 cGy (%)	123
A.30 CTV - D99% (%) $\geq$ 99% vs. Rectum - V 5770 cGy (%)	123
A.31 CTV - D99% (%) $\geq$ 99% vs. Rectum - V 6000 cGy (%)	123
A.32 CTV - D99% (%) $\leq$ 101.5% vs. Bladder - V 5000 cGy (%)	123
A.33 CTV - D99% (%) $\leq$ 101.5% vs. Bladder - V 5380 cGy (%)	124
A.34 CTV - D99% (%) $\leq$ 101.5% vs. Bladder - V 5770 cGy (%)	124
A.35 CTV - D99% (%) $\leq$ 101.5% vs. Rectum - V 3850 cGy (%)	124
A.36 CTV - D99% (%) $\leq$ 101.5% vs. Rectum - V 4000 cGy (%)	124

A.37 CTV - D99% (%) $\leq$ 101.5% vs. Rectum - V 4600 cGy (%)	125
A.38 CTV - D99% (%) $\leq$ 101.5% vs. Rectum - V 5000 cGy (%)	125
A.39 CTV - D99% (%) $\leq$ 101.5% vs. Rectum - V 5380 cGy (%)	125
A.40 CTV - D99% (%) $\leq$ 101.5% vs. Rectum - V 5770 cGy (%)	125
A.41 CTV - D99% (%) $\leq$ 101.5% vs. Rectum - V 6000 cGy (%)	126
A.42 PTV - D99% (%) vs. Bladder - V 5000 cGy (%)	126
A.43 PTV - D99% (%) vs. Bladder - V 5380 cGy (%)	126
A.44 PTV - D99% (%) vs. Bladder - V 5770 cGy (%)	126
A.45 PTV - D99% (%) vs. Rectum - V 3850 cGy (%)	127
A.46 PTV - D99% (%) vs. Rectum - V 4000 cGy (%)	127
A.47 PTV - D99% (%) vs. Rectum - V 4600 cGy (%)	127
A.48 PTV - D99% (%) vs. Rectum - V 5000 cGy (%)	127
A.49 PTV - D99% (%) vs. Rectum - V 5380 cGy (%)	128
A.50 PTV - D99% (%) vs. Rectum - V 5770 cGy (%)	128
A.51 PTV - D99% (%) vs. Rectum - V 6000 cGy (%)	128
A.52 PTV - Max Dose (cGy) vs. Bladder - V 5000 cGy (%)	128
A.53 PTV - Max Dose (cGy) vs. Bladder - V 5380 cGy (%)	129
A.54 PTV - Max Dose (cGy) vs. Bladder - V 5770 cGy (%)	129
A.55 PTV - Max Dose (cGy) vs. Rectum - V 3850 cGy (%)	129
A.56 PTV - Max Dose (cGy) vs. Rectum - V 4000 cGy (%)	129

A.57 PTV - Max Dose (cGy) vs. Rectum - V 4600 cGy (%)	130
A.58 PTV - Max Dose (cGy) vs. Rectum - V 5000 cGy (%)	130
A.59 PTV - Max Dose (cGy) vs. Rectum - V 5380 cGy (%)	130
A.60 PTV - Max Dose (cGy) vs. Rectum - V 5770 cGy (%)	130
A.61 PTV - Max Dose (cGy) vs. Rectum - V 6000 cGy (%)	131
A.62 PTV - V 6000 cGy (%) vs. Bladder - V 5000 cGy (%)	131
A.63 PTV - V 6000 cGy (%) vs. Bladder - V 5380 cGy (%)	131
A.64 PTV - V 6000 cGy (%) vs. Bladder - V 5770 cGy (%)	131
A.65 PTV - V 6000 cGy (%) vs. Rectum - V 3850 cGy (%)	132
A.66 PTV - V 6000 cGy (%) vs. Rectum - V 4000 cGy (%)	132
A.67 PTV - V 6000 cGy (%) vs. Rectum - V 4600 cGy (%)	132
A.68 PTV - V 6000 cGy (%) vs. Rectum - V 5000 cGy (%)	132
A.69 PTV - V 6000 cGy (%) vs. Rectum - V 5380 cGy (%)	133
A.70 PTV - V 6000 cGy (%) vs. Rectum - V 5770 cGy (%)	133
A.71 PTV - V 6000 cGy (%) vs. Rectum - V 6000 cGy (%)	133

List of Tables

3.1	Radiation dose guidelines for the bladder (a) and the rectum (b) for prostate cancer	16
3.2	The planning objectives and the proportion of patients meeting each criterion	18
3.3	The maximum allowable uncertainty for each RT treatment plan feature .	26
4.1	The direction of optimization for each RT treatment plan feature	43
6.1	Computational performance of the F-e-RIO solution approach	69
7.1	Computational performance of the C-e-RIO solution approach (inferring one constraint)	77
8.1	Computational performance of the iterative single-constraint inference algorithm	91
9.1	Performance summary of the solution approaches	96

List of Abbreviations

CI Conformity Index [22–25](#), [33](#), [36](#)

CT Computed Tomography [1](#)

CTV Critical Target Volume [3](#), [17](#), [18](#), [26](#), [36](#), [41](#), [43](#)

DVH Dose-volume Histogram [17](#), [33](#), [36](#), [39](#)

FO Forward Optimization [3–6](#), [8](#), [9](#), [11](#), [13](#), [28–30](#), [49](#), [57](#), [101](#), [102](#)

GRRCC Grand River Regional Cancer Centre [15–17](#), [19](#), [23](#), [25](#), [27](#), [33](#), [34](#), [39](#), [41](#), [42](#),
[44](#), [96](#), [97](#), [101–103](#), [115](#)

HI Homogeneity Index [22–24](#), [33](#), [36](#)

IMRT Intensity Modulated Radiation Therapy [12](#)

IO Inverse Optimization [3–11](#), [13](#), [14](#), [25](#), [27–33](#), [35](#), [37](#), [39](#), [48–50](#), [52](#), [57](#), [59](#), [60](#), [63](#), [64](#),
[95](#), [101](#), [102](#)

KKT Karush–Kuhn–Tucker [9](#), [10](#)

LHS Left-hand Side 7, 8, 11, 13, 32, 34, 47, 48, 55, 67, 76, 101

MILP Mixed-integer Linear Program 9, 10

MIP Mixed-integer Program 14, 75

ML Machine Learning 100, 104

MRI Magnetic Resonance Imaging 1

OAR Organ at Risk 3, 18, 26, 33, 37, 40, 41, 43, 102

OR Operations Research 85

PCA Principle Component Analysis 104

PTV Planning Target Volume 3, 17, 18, 22, 26, 36, 41, 43

RHS Right-hand Side 7, 8, 11, 13, 47, 48, 67, 76, 101

RIO Robust-inverse Optimization 3–5, 14, 26, 49, 50, 62, 65, 100–102

RO Robust Optimization 3–5, 11–13

RT Radiation Therapy ix, xv, 1–4, 8, 14–17, 19, 22–28, 33–36, 38, 39, 41–49, 51, 59, 61, 63, 67, 76, 93, 94, 96–103, 115

UI Uniformity Index 22–24, 33, 36

VMAT Volumetric-modulated Arc Therapy 15

Chapter 1

Introduction

Prostate cancer is the second most common cancer diagnosis made in men worldwide, with 1,466,680 new cases, causing 396,792 deaths (4.1% of all deaths caused by cancer) in 2022 (Bray et al., 2024). Approximately 50% of cancer patients will receive Radiation Therapy (RT) as part of their treatment journey (Delaney et al., 2005; Begg et al., 2011; Baskar et al., 2012), with an estimated 40% of curative treatment attributed to RT (Barnett et al., 2009).

Radiation therapy is a cancer treatment method that uses radiation beams to eradicate cancerous tumor cells. We will focus on external beam RT, where a linear accelerator mounted on a rotating gantry is used to treat a patient with high-energy photon beams. External beam RT aims to find the shape, direction, and intensities of a set of radiation beams such that the tumor receives a prescribed dose of radiation while sparing the surrounding healthy organs. The process for this treatment often starts with a planning session in which data on the internal anatomy are acquired (e.g., a Magnetic Resonance Imaging (MRI) or Computed Tomography (CT) scan). Using these data, along with ob-

jectives and constraints (or sometimes just rough guidelines) provided by a physician, a treatment planner determines a suitable treatment plan with the help of a software system. This plan is then forwarded to an oncologist, who will inspect it and either approve or return it to the planner for further adjustments. Producing and approving treatment plans is complex as many patterns of radiation beams may be proposed in different directions, shapes, and intensities. Typically, a few iterations of the RT treatment plan are rejected before the oncologist approves the plan. Figure 1.1 aids in describing the iterative interaction of the treatment planner and oncologist in the RT treatment planning process. Once the treatment plan is approved, it is typically delivered daily (five days a week) over several weeks (typically four to six) (Bortfeld et al., 2008). Each delivery of the RT treatment plan is called a fraction.

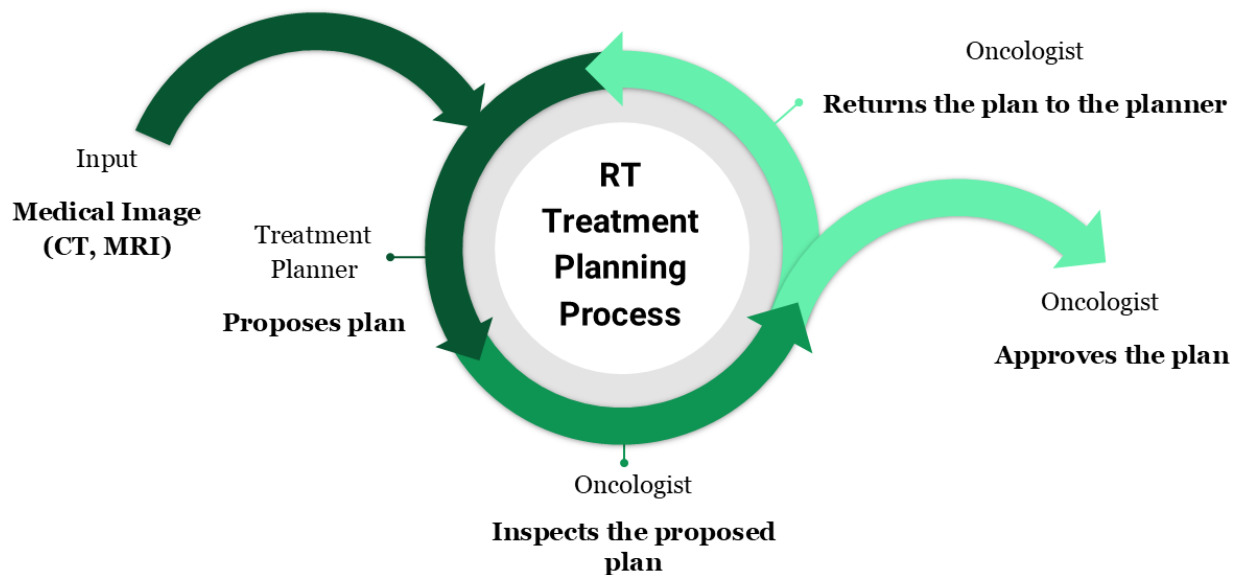


Figure 1.1: The RT treatment planning process

RT treatment plans are traditionally designed by treatment planners manually and iteratively fine-tuning the parameters of an underlying optimization model. Generally,

the accepted treatment plan violates one or more radiation constraint criteria for either the [Critical Target Volume \(CTV\)](#), [Planning Target Volume \(PTV\)](#), or an [Organ at Risk \(OAR\)](#) ([Darko et al., 2019](#)). This process can be considered subjective as the radiation plan’s approval depends on the oncologist’s opinion, which implies that there may be implicit constraints involved that may not be obvious to the treatment planners.

Hospitals may consider two common radiation guidelines for dose limitations when designing radiation therapy treatment plans: the RTOG ([RTOG, 2024](#)) and Quantec ([Marks et al., 2010](#)) guidelines; however, these guidelines are not universally agreed upon. This causes hospitals to create their own unique set of guidelines or dose goals/limitations when designing radiation plans. Another thing to note is that these RTOG and Quantec guidelines only include upper bounds for a single organ but do not consider trade-offs between organ-to-organ pairs or organ-to-target volume pairs.

One method for inferring the parameters of a problem from observed solutions is [Inverse Optimization \(IO\)](#). Significant research has been conducted on the application of [IO](#) techniques in the field of [RT](#) treatment planning ([Chan et al., 2014](#); [Boutilier et al., 2015](#); [Babier et al., 2018](#); [Chan and Lee, 2018](#); [Ghate, 2020a](#); [Ajayi et al., 2022](#); [Mahmoudzadeh and Ghobadi, 2024](#)). When dealing with uncertain or noisy data, [Robust Optimization \(RO\)](#) can be employed to address uncertainty. Similarly, extensive research has been conducted on the application of [RO](#) techniques applied to [RT](#) planning ([Chan et al., 2006](#); [Bortfeld et al., 2008](#); [Bertsimas et al., 2010](#); [Cromvik and Patriksson, 2010](#); [Chan and Mišić, 2013](#)).

To the best of our knowledge, no study has combined [IO](#) and [RO](#) techniques into a [Robust-inverse Optimization \(RIO\)](#) solution approach to infer the complete set of constraint parameters for a linear [Forward Optimization \(FO\)](#) problem while considering multiple uncertain observations. As mentioned previously, the [RT](#) treatment planning process

can be considered subjective as the radiation plan’s approval depends on the oncologist’s opinion, which implies that there may be implicit constraints involved that may not be obvious to the treatment planners. In this thesis, a quality-based [RIO](#) approach is used to impute the feasible region of the [RT](#) treatment plans and discover implicit constraints to aid the planning process of [RT](#). We identify application-specific considerations when using an [IO](#) approach for imputing the feasible region of the [RT](#) treatment plans. We then propose a quality-based [RIO](#) formulation, apply two different loss functions, and discuss the performance and quality of the results. Finally, we propose an iterative single-constraint inference algorithm that performs better than the previous formulation, and we present the inferred constraints using the iterative algorithm.

The structure of the remainder of this thesis is as follows. In Chapter [2](#), we discuss the existing literature of [IO](#) and the many variations that have been considered, as well as [RO](#) and [RIO](#). In Chapter [3](#), we discuss the data and the analysis of the data used in this study. We define the problem and outline specific considerations essential for using [RIO](#) to learn from previously accepted [RT](#) treatment plans in Chapter [4](#). In Chapter [5](#), we build the proposed quality-based robust-inverse optimization formulation from the [FO](#) problem. In Chapters [6](#) and [7](#), we provide the formulations, results, and discussions for the fairness measure formulation and compactness measure formulation, respectively. In Chapter [8](#), the proposed iterative single-constraint inference algorithm is presented, as well as the results and discussion of the solution approach similar to the previous chapters. The discussion of this thesis is presented in Chapter [9](#), followed by the conclusions in Chapter [10](#).

Chapter 2

Literature Review

IO has attracted significant attention and recognition in both research and practical applications. The literature presents numerous variations of IO problems, such as different inferred parameters, different FO problem structures, and the characteristics of the data used (Chan et al., 2023). RO has also been a steadfast method of considering uncertainty in observations for traditional optimization problems. More recently, solution approaches considering both robust and inverse optimization techniques have been proposed (Gabrel et al., 2014). Section 2.1 provides a high-level description of IO and highlights the pioneering studies in this research area. Section 2.2 examines the various parameters that can be inferred using IO techniques. Section 2.3 explores the different structures of the forward problems to which IO approaches have been applied in the literature. Section 2.4 reviews the types of observations considered in IO literature. Section 2.5 and 2.6 discuss Robust Optimization (RO) and Robust-inverse Optimization (RIO), respectively. Finally, Section 2.7 identifies a research gap revealed by this literature review and outlines the specific contributions of this thesis.

2.1 Inverse optimization background

In traditional mathematical optimization, often called the **FO** problem, an objective function and set of constraints are used as inputs to solve for an optimal solution. **IO** focuses on reverse engineering traditional mathematical optimization by considering one or more solutions as given input to solve for either the objective function, a set of constraints, or both. The solutions used for the **IO** problem are typically referred to as observations. The first study to leverage this concept was [Zhang et al. \(1995\)](#), which proposed an inverse shortest path problem that aimed to find the arc weights for the objective function of the **FO** problem while minimizing the deviations from the estimated arc weights. Shortly after, [Zhang and Liu \(1996\)](#) developed a method to solve general inverse linear programming problems to adjust the cost coefficients associated with decision variables, including specific applications to the inverse minimum cost flow and inverse assignment problems. [Ahuja and Orlin \(2001\)](#) introduced a unified framework for solving inverse optimization problems to estimate the objective coefficients of a linear optimization model and demonstrated that these inverse problems can be solved as linear programming problems. These studies paved the way for the current **IO** research we have today.

2.2 Parameters inferred through inverse optimization

The existing research on **IO** has focused on solving for the objective function of a single-objective **FO** problem ([Keshavarz et al., 2011](#); [Tavashoğlu et al., 2018](#); [Aswani et al., 2018](#); [Babier et al., 2018](#); [Ghate, 2020b](#); [Shahmoradi and Lee, 2022](#)) as well as focused on solving for the objective weights for a multi-objective **FO** problem ([Roland et al., 2013](#); [Sayre and Ruan, 2014](#); [Chan et al., 2014](#); [Boutillier et al., 2015](#); [Goli et al., 2018](#); [Chan and Lee, 2018](#);

Naghavi et al., 2019; Babier et al., 2021; Gebken and Peitz, 2021).

While **IO** studies have primarily focused on objective function inference with the assumption that the constraint parameters are known, several papers have also addressed imputing the **Right-hand Side (RHS)** parameters of the constraints in addition to the objective function parameters. Dempe and Lohse (2006) focused on finding both the objective function and **RHS** parameters by minimizing the distance between an observed point and the optimal solution. Chow and Recker (2012) proposed a method that jointly estimates both the objective coefficients and the **RHS** of the constraints for calibrating the household activity pattern problem such that perturbation from the initial set of parameters under the $\mathbf{L}_1$ norm is minimized. Černý and Hladík (2016) investigated inverse linear programming with intervals to find the optimal parameter set for three scenarios: when parameters appear on the **RHS**, when parameters appear in the objective function, and when parameters appear in both the **RHS** and the objective function. Saez-Gallego and Morales (2017) leveraged **IO** to estimate the objective function and **RHS** parameters for time series forecasting for load prediction.

Shifting focus from **RHS** parameter estimation, Chan and Kaw (2020) proposed two **IO** models to impute the **Left-hand Side (LHS)** constraint parameters in addition to the objective function parameters based on a single observed optimal solution. Ghatrani and Ghate (2024) studied inverse semidefinite programs where the **LHS** constraint parameters are unknown and three sub-cases where both the **RHS** and objective parameters are unknown or only one of these two is unknown. Their study differs from our work as they consider semidefinite programs, which are specific types of optimization problems that extend traditional optimization. Most closely related to this study, Ghobadi and Mahmoudzadeh (2021) and Mahmoudzadeh and Ghobadi (2024) proposed **IO** models to infer the complete feasible region, both the **LHS** and **RHS** constraint parameters, of a linear

program, provided a known objective function and multiple observations. Although the models we propose in this study are similar to that of [Ghobadi and Mahmoudzadeh \(2021\)](#) and [Mahmoudzadeh and Ghobadi \(2024\)](#), we consider uncertainty in the observations as well as propose an iterative single-constraint inference algorithm to improve the solution time. We also consider problem-specific details to aid in inferring constraints to our specific application of learning from past successful [RT](#) treatment plans.

[IO](#) problems are structurally and mathematically different depending on whether the model is inferring the constraint parameters or only the objective function parameters. When solving for the objective function parameters, the imputed objective function needs to maintain the optimality of the observed optimal point. This may be done by enforcing dual feasibility and strong duality to ensure the [FO](#) has an optimal solution. [Ahuja and Orlin \(2001\)](#) leveraged dual feasibility and complementary slackness optimality conditions to propose their [IO](#) formulation. On the other hand, [Chan et al. \(2019\)](#) used strong duality optimality conditions instead of complementary slackness. When inferring the constraint parameters, it is crucial to consider primal feasibility alongside dual feasibility and strong duality optimality conditions. It is essential to consider primal feasibility, as the inferred constraints must not be violated by the provided observations ([Ghobadi and Mahmoudzadeh, 2021](#)). A result of inferring both the [LHS](#) and [RHS](#) of the constraint parameters is that the formulation becomes non-linear, making it more complex and, therefore, less studied in the literature.

2.3 Structure of the forward problem

It is essential to consider the structure of the forward problem as it plays a role in the tractability of the [IO](#) problem. The forward problem structure impacts the complexity

and mathematical techniques used to solve the [IO](#) problem. Possible structures of the [FO](#) problem include linear, conic, convex, and discrete optimization models. Sequential decision-making models (e.g., Markov decision processes (MDPs)) are also possible ([Chan et al., 2023](#)). For a linear forward problem, both the objective function and constraints are linear. In the literature, there have been studies that have focused on imputing the objective function of a linear [FO](#) problem ([Ahuja and Orlin, 2001](#); [Chan et al., 2019](#); [Ghate, 2020b](#); [Babier et al., 2021](#); [Shahmoradi and Lee, 2022](#)) as well as studies that have focused on solving for the constraints parameters of a linear [FO](#) problem ([Güler and Hamacher, 2010](#); [Černý and Hladík, 2016](#); [Saez-Gallego and Morales, 2017](#); [Chan and Kaw, 2020](#); [Ghobadi and Mahmoudzadeh, 2021](#)).

The complexity of the inverse problem increases when the forward problem is not linear. The [Karush–Kuhn–Tucker \(KKT\)](#) conditions serve as sufficient optimality conditions when the structure of the forward problem is convex or conic ([Mahmoudzadeh and Ghobadi, 2024](#)). In conic and convex models, it’s possible to optimize any combination of linear, second-order, and quadratic objectives over a linear, second-order, or semidefinite cone. [Iyengar and Kang \(2005\)](#) discussed the [IO](#) problem of a conic program. They leverage the [KKT](#) conditions to estimate the parameters of the objective function of a conic [FO](#) problem. [Zhang and Zhang \(2010\)](#) considered an inverse quadratic programming problem where the objective function is perturbed as little as possible so that a known observation becomes optimal. They presented an inverse conic programming problem with a quadratic objective and linear constraints. [Keshavarz et al. \(2011\)](#) and [Aswani et al. \(2018\)](#) utilized [KKT](#) conditions to solve the objective function parameters of a convex optimization problem. [Zhang et al. \(2010\)](#) and [Zhang et al. \(2015\)](#) discussed the inverse quadratic and second-order cone quadratic programming problems, respectively.

Discrete optimization models refer to [Mixed-integer Linear Program \(MILP\)](#) problems.

The optimality of [MILP](#) forward problems cannot be easily represented by a small number of equations (e.g., the [KKT](#) conditions) ([Chan et al., 2023](#)). In the literature, [IO](#) methods for discrete optimization models that used strong duality and dual feasibility conditions, similar to the linear forward problem, led to an [IO](#) formulation with an exponential number of variables and constraints ([Schaefer, 2009](#)). An alternative approach for discrete forward problems is to apply cutting plane algorithms that enhance the accuracy of the approximate solution for the inverse model ([Wang, 2009](#)). Further studies have considered [IO](#) methods where the forward problem is discrete, and the cost vector is unknown ([Lamperski and Schaefer, 2015](#); [Bulut and Ralphs, 2021](#); [Moghaddass and Terekhov, 2021](#); [Bodur et al., 2022](#)).

2.4 Characteristics of the data

The input of an [IO](#) problem can be one or many observations. In the case of many observations, they can be optimal, near-optimal, feasible, or infeasible. The single-point [IO](#) problem takes a single observation as input, whereas the multi-point [IO](#) problem uses a finite number of multiple observations as input ([Ghobadi and Mahmoudzadeh, 2021](#)). When imputing the objective function parameters, single-point [IO](#) is often used where the input is a feasible and optimal observation ([Zhang and Liu, 1996, 1999](#); [Iyengar and Kang, 2005](#); [Zhang et al., 2010](#); [Ghate, 2020b](#)) or a feasible and near-optimal observation ([Aswani et al., 2018](#); [Chan et al., 2019](#)). In the literature, there have been studies that have focused on the single-point problem while acknowledging that the observation might be infeasible ([Chan et al., 2014](#); [Naghavi et al., 2019](#)). There also exists research that has considered multi-point [IO](#) for objective inference given the observations are near-optimal ([Keshavarz et al., 2011](#); [Bertsimas et al., 2015](#); [Ajayi et al., 2022](#)). [Mohajerin Esfahani](#)

et al. (2018) and Ahmadi et al. (2023) considered using infeasible observations that are made to be near-optimal. Shahmoradi and Lee (2022) leveraged multiple infeasible near-optimal observations to impute the cost function of a linear FO problem. Babier et al. (2021) developed an IO model that considers both feasible and infeasible observations to infer the objective function.

Dempe and Lohse (2006) and Chow and Recker (2012) proposed IO models to determine the objective function and RHS parameters using a single feasible observation while taking the sub-optimality gap into account. Chan and Kaw (2020) developed an IO model to infer the cost vector and LHS parameters, also using a single observation and considering the sub-optimality gap. Similarly, previous research has considered the optimality gap in inverse problems and used a single observation to infer the RHS parameters (Güler and Hamacher, 2010). When inferring the complete set of constraint parameters (both the LHS and RHS parameters), Ghobadi and Mahmoudzadeh (2021) utilized multiple feasible observations. Shortly after, Mahmoudzadeh and Ghobadi (2024) considered an IO method to leverage both feasible and infeasible observations to infer the complete set of constraint parameters.

2.5 Robust optimization

RO is a modeling approach for solving optimization problems while considering uncertainty in the data. The first study to introduce RO was by Soyster (1973) and has been extensively studied over the years. For comprehensive reviews of robust optimization, readers are referred to Beyer and Sendhoff (2007), Ben-Tal et al. (2009), Bertsimas et al. (2011), and Gabrel et al. (2014). RO requires only the uncertainty range for the uncertain parameters. In RO, the uncertainty range is usually expressed as an uncertainty set that includes all

possible realizations of the uncertain data.

A common criticism of RO is that it is overly conservative and only optimal for the most extreme scenarios of uncertainty. Soyster (1973) introduced a linear optimization model to solve for an optimal solution that remains feasible for all possible scenarios of the uncertain data. El Ghaoui and Lebret (1997) and Ben-Tal and Nemirovski (2000) proposed less conservative approaches by considering the uncertain parameters to be within ellipsoidal uncertainty sets. Bertsimas and Sim (2004) then proposed a new approach to combat the issue of being overly conservative. This approach introduced the concept of “budget of uncertainty,” which offers full control over the level of uncertainty in the model. By considering the budget of uncertainty, it provides a parameter in the model that determines the maximum amount of uncertainty that is allowed.

Chan et al. (2006) proposed a RO formulation to deal with breathing uncertainty in Intensity Modulated Radiation Therapy (IMRT) planning. This approach includes uncertainty in the constraint parameters such that an infinite number of constraints must be considered. Using the theory of linear programming duality, Chan et al. (2006) presented that an equivalent linear formulation, which has a finite number of constraints called the robust counterpart, may be considered.

2.6 Robust-inverse optimization

More recently, research has considered both inverse and robust optimization techniques in a single solution approach. Due to the broad term “inverse optimization,” which can be used to describe imputing different parameters of an optimization model as presented in Section 2.2, many studies have described their models as robust inverse optimization models; however, their work differs from the proposed solution approach in this study.

As mentioned previously, [Chan and Kaw \(2020\)](#) proposed two [IO](#) models to impute the [LHS](#) constraint parameters in addition to the objective function parameters based on a single observed optimal solution. This study applied these models to estimate the uncertainty set parameters in robust linear optimization problems. Our study differs from their work in two distinct ways. The first is that [Chan and Kaw \(2020\)](#) considers only a single point, whereas we consider multiple observations. The second is that we employ [RO](#) techniques to the [IO](#) formulation as we consider uncertainty in the observations themselves, whereas [Chan and Kaw \(2020\)](#) consider the forward problem to be an [RO](#) problem.

Another study by [Ghobadi et al. \(2018\)](#) discussed an [IO](#) approach that imputes the cost vector while considering uncertainty in the observation. Their work differs in two ways as well. The first is that their approach estimates the objective function parameters rather than the constraint parameters. This difference, as stated in [Section 2.2](#), creates a problem that is significantly structurally and mathematically different. The second difference is that only a single uncertain observation is considered, whereas this study considers multiple uncertain observations as input.

[Aswani et al. \(2018\)](#) proposed an [IO](#) approach to impute the objective function parameters by using multiple uncertain observations as input. Similar to the previous study, their model differs from this approach as the [IO](#) approach in this study imputes both the [LHS](#) and [RHS](#) constraint parameters rather than the cost function.

2.7 Contributions

Among all the studies previously mentioned, none have employed an [IO](#) approach that simultaneously infers both the [LHS](#) and [RHS](#) constraint parameters of a linear [FO](#) problem

while considering multiple uncertain observations. In this thesis, a quality-based robust-inverse optimization approach is applied to previously accepted [RT](#) treatment plans for patients with prostate cancer to uncover the implicit logic of the oncologist. We aim to provide these inferred constraints to the treatment planners, enabling them to design better treatment plans that can be approved with fewer iterations. To the best of my knowledge, this solution approach has not been addressed by previous research in the literature. Among all the papers discussed in the literature review, [Chan and Kaw \(2020\)](#) and [Ghobadi and Mahmoudzadeh \(2021\)](#) are the two most relevant studies to this work; however, the research in this study introduces a few novel contributions:

- First, we identify application-specific considerations that must be taken into account when using [IO](#) to learn from past accepted [RT](#) treatment plans and apply these considerations to the following solution approaches.
- Second, we propose a quality-based [Robust-inverse Optimization \(RIO\)](#) formulation and apply two different loss functions that may be used in the formulation. We then discuss the performance of each of these loss functions applied to the quality-based [RIO](#) formulation.
- Third, we propose an iterative single-constraint inference algorithm that is faster and more efficient than the previous quality-based [RIO](#) formulation by mitigating the computational complexity of solving the full [Mixed-integer Program \(MIP\)](#) model. We solve for an updated set of guidelines and trade-offs between organ-to-organ or organ-to-target volume pairs. Finally, we present interesting findings based on the results of the iterative algorithm.

Chapter 3

Data Description and Analysis

The data used in this study was collected from the [Grand River Regional Cancer Centre \(GRRCC\)](#). The dataset consists of values from a set of [RT](#) treatment plan features for 273 intact prostate cancer patient [RT](#) treatment plans, spanning from the years 2021 to 2023. The unit Gy is used to denote Gray, which is used as a unit of radiation dose. One Gy is defined as the absorption of one joule of radiation energy per kilogram of matter. The unit cGy is used to denote Centigray. 1 Gy is equivalent to 100 cGy. The dose prescription of these radiation plans was 60 Gy, which was delivered in 20 fractions (3.0 Gy per fraction).

The delivery of these [RT](#) treatment plans was done so using a [Volumetric-modulated Arc Therapy \(VMAT\)](#) technique. The [VMAT](#) technique is a form of [RT](#), which involves continuous irradiation at a constant or variable dose rate using a gantry that rotates around the patient using one or more arcs ([Darko et al., 2019](#)).

3.1 Existing dose constraints

As mentioned previously, hospitals may use the RTOG (RTOG, 2024) and Quantec (Marks et al., 2010) guidelines to reference safe radiation absorption in different organs; However, these guidelines are not universally agreed upon. For instance, the current dose goals being used in the GRRCC, as seen in Table 3.2, do not match with either the RTOG or Quantec guidelines.

Table 3.1: Radiation dose guidelines for the bladder (a) and the rectum (b) for prostate cancer

(a) Bladder			(b) Rectum		
Dose	RTOG	Quantec	Dose	RTOG	Quantec
V 5000 cGy $\leq$	60%	-	V 5000 cGy $\leq$	50%	50%
V 6500 cGy $\leq$	50%	50%	V 6000 cGy $\leq$	50%	35%
V 6660 cGy $\leq$	40%	-	V 6500 cGy $\leq$	35%	25%
V 7000 cGy $\leq$	35%	35%	V 6660 cGy $\leq$	25%	-
V 7500 cGy $\leq$	25%	25%	V 7000 cGy $\leq$	25%	20%
V 8000 cGy $\leq$	15%	15%	V 7500 cGy $\leq$	15%	15%

Table 3.1 displays the radiation dose guidelines for the bladder and rectum with respect to treating prostate cancer with RT, according to RTOG and Quantec. Looking at the first line in Table 3.1a, it says that the percent volume of the bladder that receives 5000 cGy or more must not exceed 60%. These guidelines are created to ensure certain organs, such as the bladder and rectum, are not dosed with dangerous levels of radiation when treating prostate cancer with radiation therapy. It can be seen that some information is conflicting or even missing between the two guidelines. For example, the values for V 6000 cGy for

the rectum in Table 3.1b do not match (35% vs. 25%). Also, note that these guidelines only include upper bounds on each organ independently but do not consider trade-offs.

When creating the RT treatment plans for patients with prostate cancer, GRRCC keeps a number of dose constraints or planning objectives in mind that are specific to their cancer center. In a perfect world, every treatment plan would meet every planning objective; however, this is not the case in practice. The dose-volume constraints that were considered by the GRRCC when creating the RT treatment plans from the dataset in this study can be seen in Table 3.2. These values are taken from a Dose-volume Histogram (DVH). A DVH is a histogram that relates radiation dose to tissue volume in RT planning. The dose-volume criteria in Table 3.2 is displayed similarly to the RTOG and Quantic dose constraints. The D stands for dose for a dose constraint, and the V stands for volume for a volume constraint. The first percentage value in a dose constraint (D) refers to the percent of the target dose; in this case, the target dose is 60 Gy or 6000 cGy. The second percentage value in a dose constraint (D) is the dose limit. The radiation value in a volume constraint (V) refers to the volume that receives that given amount of radiation or more. The percentage value in a volume constraint (V) is the percent volume limit. For instance, $D\ 99.0\% \geq 99.0\%$ for the CTV means the dose received by 99.0% of the CTV must be larger than or equal to 99.0% of the target dose (in this case 99% of 60 Gy is 59.4 Gy). Similarly, the volume constraint $V\ 6000\ \text{cGy} \leq 1.0\%$ for the rectum means the volume of the rectum that receives 6000 cGy or more must be less than or equal to 1.0%.

Table 3.2 shows dose-volume constraints for the Critical Target Volume (CTV), Planning Target Volume (PTV), rectum, bladder, left femoral head and right femoral head used by GRRCC for the 273 RT plans we collected. The right column of the table shows the percentage of patients meeting each criterion. Note, not all RT plans that have been accepted meet every planning objective. Most of the planning objectives have a high per-

Table 3.2: The planning objectives and the proportion of patients meeting each criterion (D: dose constraint, V: volume constraint)

Target volume/ OAR	Dose constraint	Percentage of patients meeting criteria
CTV	D 99.0% $\geq$ 99.0%	100.00%
	D 99.0% $\leq$ 101.5%	96.34%
PTV	D 99.0% $\geq$ 95.0%	90.11%
	Dmax $\leq$ 105.0%	69.23%
	V 6000 cGy $\geq$ 75.0%	88.28%
Rectum	V 6000 cGy $\leq$ 1.0%	51.28%
	V 5770 cGy $\leq$ 15.0%	99.63%
	V 5380 cGy $\leq$ 20.0%	97.44%
	V 5000 cGy $\leq$ 22.0%	80.59%
	V 4600 cGy $\leq$ 35.0%	98.53%
	V 4000 cGy $\leq$ 38.0%	79.49%
	V 3850 cGy $\leq$ 50.0%	98.17%
Bladder	V 5770 cGy $\leq$ 15.0%	86.45%
	V 5380 cGy $\leq$ 35.0%	98.90%
	V 5000 cGy $\leq$ 40.0%	98.90%
Left femoral head	V 4300 cGy $\leq$ 2.0%	100.00%
Right femoral head	V 4300 cGy $\leq$ 2.0%	99.63%

centage of patients meeting the criteria; however, some dose constraints, such as V 6000 cGy $\leq$ 1.0% for the rectum in the sixth row, have a much lower percentage (51.25%). This

motivates the question of where and why RT plans violate these dose-volume constraints. There could be dose-volume constraints that are too strict, or possibly the current guidelines GRRCC are currently using do not accurately represent the RT they are accepting. An interesting area to investigate is how much these RT treatment plan features deviate from the planning objectives. This may be done by looking at the distribution of each RT plan feature used to check these planning objectives.

Figure 3.1 displays the distribution of each feature presented in the dose-volume constraints in Table 3.2 using box and whisker plots. Figure 3.1 also includes the dose-volume constraints from Table 3.2 as green lines with an arrow to point in the direction of the constraint. As seen in Figure 3.1, some features from the planning objectives violate the dose goals significantly. Particularly for the dose constraint, Rectum - V 6000 cGy $\leq$ 1.0%, we can see that the RT treatment plans that were accepted and violated this objective are relatively far from the dose goal. The fact that not all planning objectives are met for RT plans suggests that, in some cases, the dose-volume constraints are too strict; however, it does not show that relaxing the dose-volume constraints would better represent the plans being accepted. This still raises the question of why one or more dose-volume constraints are violated in RT plans.

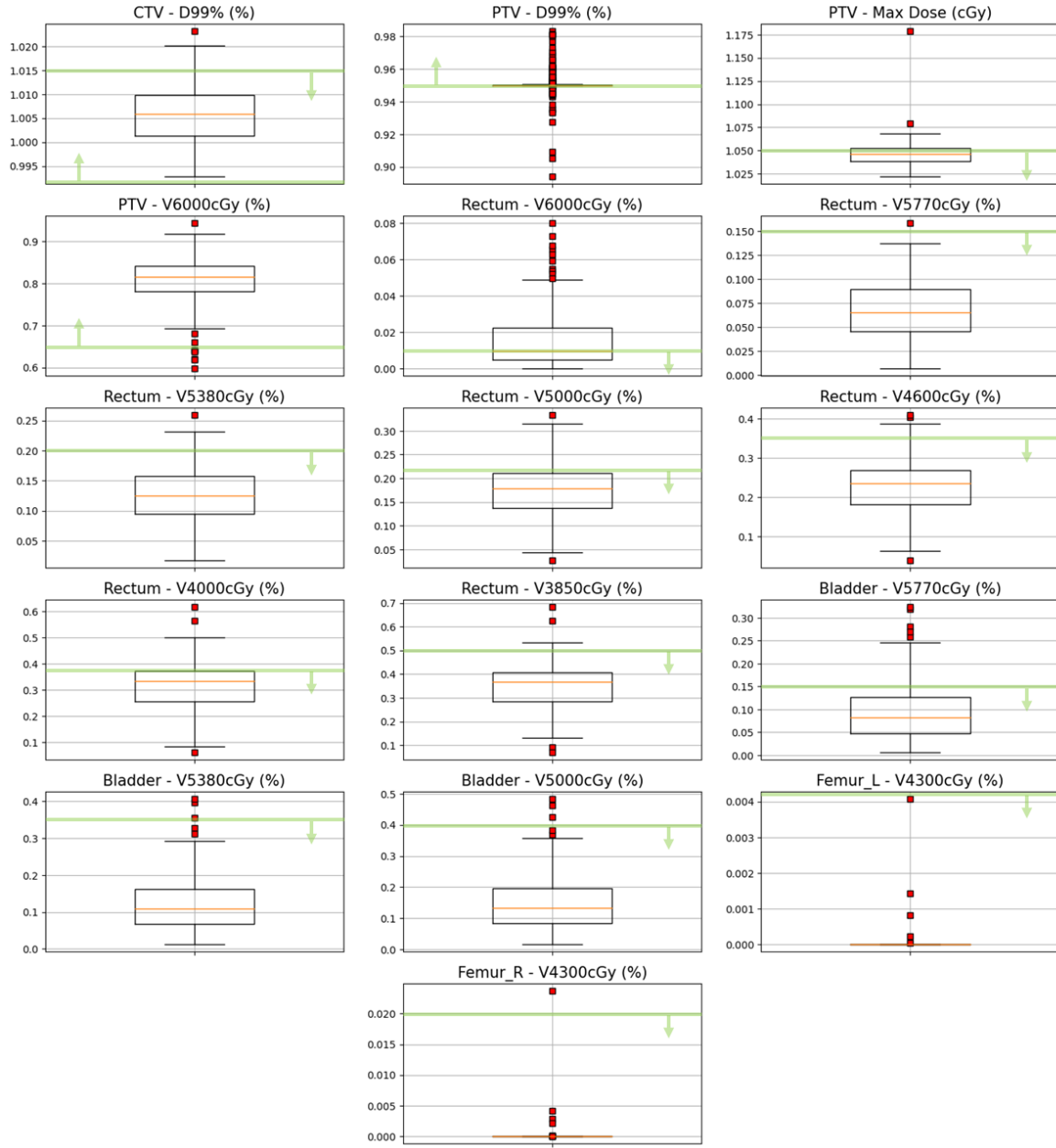


Figure 3.1: Box and whisker plots of the features from the planning objectives (the planning objectives are in green)

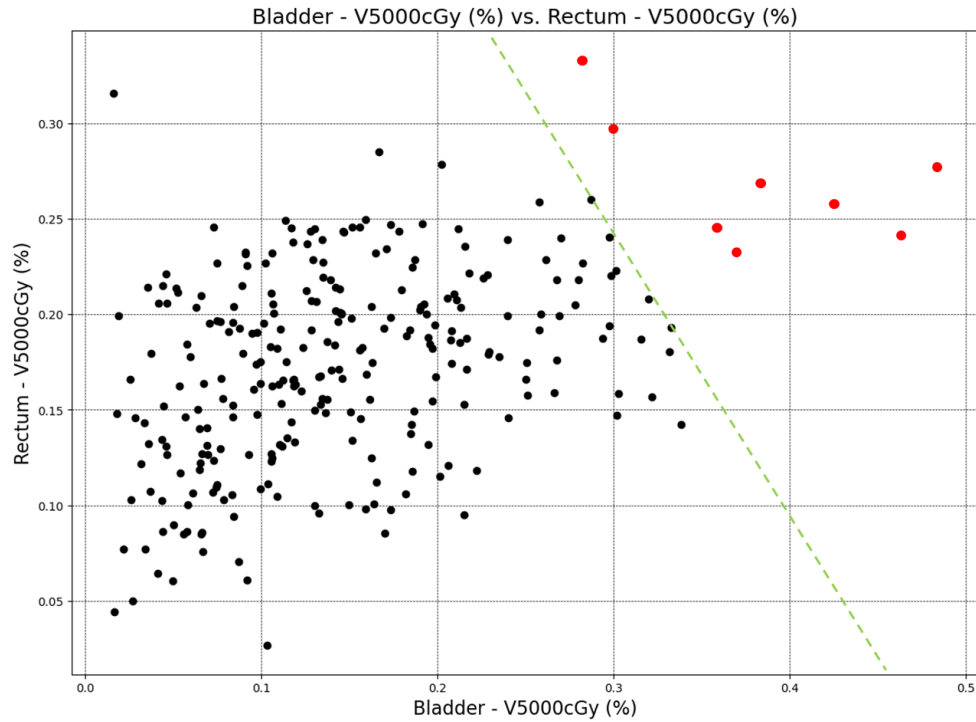


Figure 3.2: Two plan features with potential outliers (red) and a potential implicit trade-off constraint (green)

Figure 3.2 shows a scatter plot of two random plan features from Table 3.2. With the understanding that both of these features must be minimized, there are two interesting insights to point out. The first is that there are observations in the scatter plot that seem to be potential outliers (the red observations). The second is that, with the outliers in mind, there appears to be an implicit trade-off constraint that could be clinically meaningful. These two insights will be discussed in more detail later in the study.

3.2 Radiation therapy treatment plan quality

An interesting topic that will be leveraged in later chapters is using the quality of an RT plan. The quality of an RT plan may be determined by how well the prescribed dose conforms to the target volume and the variability of the dose distribution within the target volume (Darko et al., 2019). In this study, the Homogeneity Index (HI), Uniformity Index (UI), and Conformity Index (CI) values for the PTV are utilized to quantitatively evaluate the quality of an RT treatment plan. The Homogeneity Index (HI) value describes how homogeneous the plan is based on how close the minimum doses received by 2% of the PTV volume (D_2) and the minimum doses received by 98% of the PTV volume (D_{98}) are to each other with respect to the prescribed dose (D_{PD}) and has an ideal value of zero. A HI value greater than zero indicates a plan with less homogeneity. Therefore, the HI value should be minimized. The Uniformity Index (UI) value is defined as the ratio between minimum doses reached in 5% of the PTV volume (D_5) and the minimum dose reached in 95% of PTV volume (D_{95}). The ideal value for the UI is one. In fact, the UI value will always be one or larger as the D_5 is always equal or larger to the D_{95} . Similar to the HI value, the UI value should be minimized. The Conformity Index (CI) value is the ratio of the volume of the PTV covered by the reference isodose line (95% isodose line) (V_{RI}) and the target volume of the PTV (TV). A CI value greater than one indicates that the irradiated volume exceeds the target volume and covers part of the healthy tissue, whereas a CI less than one indicates that the irradiated volume is not covered completely. Therefore, the deviation of the CI value from one should be minimized. The index values stated above are calculated as follows:

$$HI = \frac{D_2 - D_{98}}{D_{PD}}, \quad (3.1)$$

$$\text{UI} = \frac{D_5}{D_{95}}, \quad (3.2)$$

$$\text{CI} = \frac{V_{RI}}{TV}. \quad (3.3)$$

The dose values used to calculate the **HI**, **UI**, and **CI** values are the default dose values used by the **GRRCC**. There is much discussion on quantifying the quality of an **RT** plan. However, there is no universally agreed-upon calculation or method. For the purpose of this study, we decided to use a weighted sum of the **HI**, **UI**, and **CI** values to determine the quality of an **RT** plan as seen in Equation 3.4.

$$\text{RT treatment plan quality} = C_1\text{HI} + C_2\text{UI} + C_3\text{CI}. \quad (3.4)$$

Equation 3.4 was considered acceptable by the experts at the **GRRCC**. Other measures of **RT** plan quality could be interchanged in the solution approaches presented in this study. This equation is not the primary contribution of this study. Rather, the **RT** plan quality

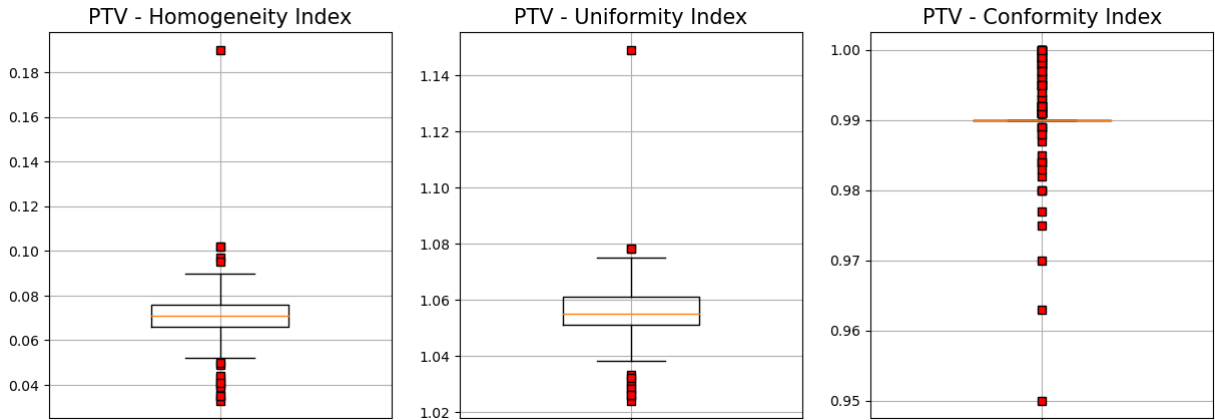


Figure 3.3: Box and whisker plots of the **HI**, **UI** and **CI** values

equation presented serves to demonstrate a quality-based robust-inverse optimization solution approach, which will be discussed in later chapters. Equation 3.4 is stated with the assumption that a lower value corresponds to a better quality plan. Figure 3.3 illustrates the distribution of the HI, UI, and CI values for the dataset. As mentioned previously, lower values are considered better for the HI, and UI values, and a lower deviation of the CI value from one is considered better. Figure 3.3 displays the distribution of the HI, UI, and CI values from the 273 RT plans used in this study. Looking at the distribution of the CI values in Figure 3.3, it shows that no CI values are greater than one, so it is safe to say that higher CI values are better for this set of data specifically. The weights for the RT treatment plan quality we used was $[C_1, C_2, C_3] = [1, 1, -1]$.

Keeping in mind a lower quality treatment value is better, lower values are better for the HI and UI values and higher values are better for the CI values (in this case specifically), it is clearer to see why the signs of the weights for the HI, UI and CI values in Equation

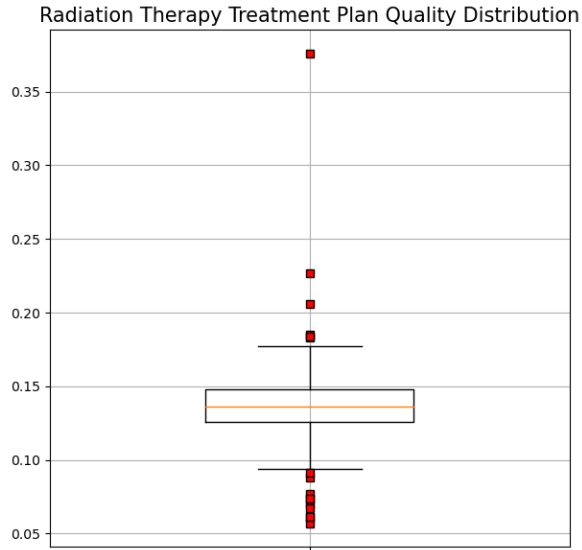


Figure 3.4: Box and whisker plot of the RT treatment plan quality values

3.4 are positive for the first two index values, and negative for the CI value. The values of the weights in Equation 3.4 themselves were chosen to equally distribute the importance of each index value for the quality calculation. Again, Equation 3.4 and the weights used in the equation were considered acceptable by the experts at the GRRCC. Figure 3.4 displays the distribution of the quality values of the 273 RT plans using Equation 3.4.

3.3 Data uncertainty

Another interesting topic that will be leveraged in later chapters is integrating uncertainty around the RT plan feature values for each observation in the IO model. Two reasons for considering uncertainty in the observations of RT treatment plans are the accuracy of the data and to increase the likelihood of including the optimal solution in the imputed feasible region. The latter reason will be discussed in more detail later in the study.

From an IO perspective, past decisions are broken down into data points. When evaluating these data points, it is essential to consider their accuracy. The dose values provided in the RT plans are determined by an RT treatment planning software. It is reasonable to consider slight differences in the values provided in that data and the actual doses received by the patients in real life.

In later chapters, the concept of quality-based uncertainty is explored, where the better quality plans have a larger uncertainty. It can be easy to tell if an RT plan is good or bad by comparing the individual plan features to the guidelines used to design the plan. However, it is hard to tell if the optimal plan exists within the set of observations. It is possible that the best RT plan ever to be designed is not in the set of observations. We do not want to infer a feasible region that potentially cuts off better plans from being considered in the future. To increase the likelihood of including the optimal solution in the

set of observations, a larger margin of uncertainty is considered for higher-quality plans. This also provides room in the feasible region to allow for further improvements.

Table 3.3 displays the values of maximum uncertainty for the features that are considered in the RIO formulations in later chapters. We determined the maximum uncertainty values in Table 3.3 based on a visual analysis of the distribution of each of the RT treat-

Table 3.3: The maximum allowable uncertainty for each RT treatment plan feature

Target volume/OAR	Dose constraint	Maximum Uncertainty
CTV	$D\ 99.0\% \geq 99.0\%$	$\pm 0.15\%$
	$D\ 99.0\% \leq 101.5\%$	$\pm 0.15\%$
PTV	$D\ 99.0\% \geq 95.0\%$	$\pm 0.50\%$
	$D_{\max} \leq 105.0\%$	$\pm 50\text{ cGy}$
	$V\ 6000\text{ cGy} \geq 75.0\%$	$\pm 2.00\%$
Rectum	$V\ 6000\text{ cGy} \leq 1.0\%$	$\pm 0.20\%$
	$V\ 5770\text{ cGy} \leq 15.0\%$	$\pm 0.50\%$
	$V\ 5380\text{ cGy} \leq 20.0\%$	$\pm 1.00\%$
	$V\ 5000\text{ cGy} \leq 22.0\%$	$\pm 2.00\%$
	$V\ 4600\text{ cGy} \leq 35.0\%$	$\pm 2.00\%$
	$V\ 4000\text{ cGy} \leq 38.0\%$	$\pm 2.00\%$
	$V\ 3850\text{ cGy} \leq 50.0\%$	$\pm 2.00\%$
Bladder	$V\ 5770\text{ cGy} \leq 15.0\%$	$\pm 2.00\%$
	$V\ 5380\text{ cGy} \leq 35.0\%$	$\pm 2.00\%$
	$V\ 5000\text{ cGy} \leq 40.0\%$	$\pm 2.00\%$

ment plan features. The values we chose were considered acceptable by experts at the [GRRCC](#). The values used in Table [3.3](#) are not the main focus of this thesis, and they serve to demonstrate a quality-based robust-inverse optimization solution approach, which will be discussed in later chapters. Note that the left and right femoral head uncertainty values are not included in this table. The reasoning for this removal of these features is mentioned in Section [4.3](#), which discusses feature selection. It should also be noted that these uncertainty values may be fine-tuned by hospital staff as required.

3.4 Summary

Chapter [3](#) provided a comprehensive description and analysis of the data used in this study, which was sourced from the [Grand River Regional Cancer Centre \(GRRCC\)](#). The dataset is comprised of 273 intact prostate cancer patient [RT](#) treatment plans from 2021 to 2023. The guidelines that were used by the [GRRCC](#) to design the [RT](#) treatment plans were presented and compared against the data. The concepts of treatment plan quality and uncertainty in the data were discussed. This analysis sets the stage for the following chapters in this thesis. In the forthcoming chapter, we delve into the inverse optimization problem for radiation therapy guidelines, addressing specific application-based considerations necessary for learning from [RT](#) treatment plans when using an [IO](#) approach.

Chapter 4

Inverse Optimization for Radiation Therapy Guidelines

Certain application-based considerations must be taken into account when using an [Inverse Optimization \(IO\)](#) approach to learn from [RT](#) treatment plans. Section [4.1](#) in this chapter discusses background information on the [IO](#) formulation for a linear [FO](#) problem in order to explain the concepts in the following sections. Section [4.2](#) then presents a pairwise feature solution approach to increase the ability to interpret the inferred constraints. The pairwise solution approach then motivates Section [4.3](#), where we discuss the features as well as pairs of features that will be used in the following [IO](#) approaches in later chapters. Section [4.4](#) outlines the need for an outlier detection algorithm and presents the outlier detection algorithm that will be used throughout the rest of the study. Then, in Section [4.5](#), we present additional constraints to be applied to the [IO](#) formulation to ensure the imputed constraints do not exclude better [RT](#) treatment plans from being considered.

4.1 Inverse optimization background

In traditional mathematical optimization, an objective function and set of constraints are used as input to solve for an optimal solution. [Inverse Optimization \(IO\)](#) focuses on reverse engineering traditional mathematical optimization by considering one or more solutions as given input to solve for either the objective function, a set of constraints, or both. More specifically, it leverages prior solutions to a [Forward Optimization \(FO\)](#) problem to rebuild the mathematical model that would render these solutions approximately or exactly optimal. Solving for these parameters itself is an optimization model known as the “inverse model” ([Chan et al., 2023](#)). [IO](#) is structured in its solution methodology, which allows the user to clearly see the logic of the model through an optimization problem that is solved to aid in making future decisions.

The notation for the remainder of this paper is as follows. All vectors are column vectors, denoted as bold lowercase letters. Scalars are denoted as regular text lowercase letters. Matrices are denoted as bold uppercase letters. Sets are denoted as calligraphic uppercase letters. All inequalities are row-wise.

4.1.1 Forward optimization problem

To understand the [Inverse Optimization \(IO\)](#) problem, let’s start with the [Forward Optimization \(FO\)](#) problem. The [FO](#) problem consists of an objective function, two sets of inequality constraints, and unrestricted decision variables. Let $\mathbf{c}$ be the objective function parameters. Let $\mathbf{A}$ and $\mathbf{b}$ be the unknown constraint parameters we are solving for in the [FO](#) model. Let $\mathbf{G}$ and $\mathbf{h}$ be the parameters of the known constraints. Let $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{m_1 \times n}$, $\mathbf{b} \in \mathbb{R}^{m_1}$, $\mathbf{G} \in \mathbb{R}^{m_2 \times n}$, and $\mathbf{h} \in \mathbb{R}^{m_2}$. We define our [FO](#) problem as:

$$\mathbf{FO} : \quad \underset{\mathbf{x}}{\text{minimize}} \quad \mathbf{c}'\mathbf{x} \quad (4.1a)$$

$$\text{subject to} \quad \mathbf{Ax} \geq \mathbf{b}, \quad (4.1b)$$

$$\mathbf{Gx} \geq \mathbf{h}, \quad (4.1c)$$

$$\mathbf{x} \in \mathbb{R}^n. \quad (4.1d)$$

Constraints (4.1b) and (4.1c) represent the set of unknown and known constraints for the **FO** problem, respectively. Throughout this study, we index the unknown and known constraints using the sets $\mathcal{I}_1 = \{1, \dots, m_1\}$ and $\mathcal{I}_2 = \{1, \dots, m_2\}$, respectively. The set $\mathcal{J} = \{1, \dots, n\}$ denotes the indices of the decision variable $\mathbf{x}$ in the forward problem. Let $\mathbf{a}'_i$ be the i^{th} row of the constraint matrix $\mathbf{A}$ and a_{ij} as the element in the i^{th} row and j^{th} column of the constraint matrix $\mathbf{A}$, for $i \in \mathcal{I}_1$ and $j \in \mathcal{J}$. This means that the matrix $\mathbf{A}$ contains all the row vectors $\mathbf{a}'_i$ for $i \in \mathcal{I}_1$, and each element a_{ij} for $i \in \mathcal{I}_1$ and $j \in \mathcal{J}$. These are different representations of matrix $\mathbf{A}$ referenced throughout this study. The i^{th} element of vector $\mathbf{b}$ is denoted as b_i for $i \in \mathcal{I}_1$. For the **IO** problem, we would like to consider multiple feasible solutions, which we call observations. The set of the finite number of observations is defined as $\mathcal{K} = \{1, \dots, p\}$, where $\mathbf{x}^k$ is the k^{th} observation to the forward problem and x_j^k is the j^{th} element of the k^{th} observation, for $j \in \mathcal{J}$ and $k \in \mathcal{K}$. The bold number $\mathbf{0}$ is used to denote an all-zeros vector. We make the following assumption that the objective function of the **FO** formulation is known.

Assumption 1 *The objective function vector, $\mathbf{c}$, is known.*

4.1.2 Inverse optimization problem

Given that the objective function is known, we can determine the preferred solution by finding the observation with the best objective function value. We denote the preferred solution or observation as $\mathbf{x}^0$, which is defined as:

$$\mathbf{x}^0 \in \arg \min_{\mathbf{x}^k, k \in \mathcal{K}} \{\mathbf{c}'\mathbf{x}^k\}. \quad (4.2)$$

If there are multiple optimal observations, without loss of generality, we arbitrarily choose one as $\mathbf{x}^0$. Let $\mathbf{y}$ and $\mathbf{w}$ be the dual vectors associated with constraints (4.1b) and (4.1c), respectively. Based on Ghobadi and Mahmoudzadeh (2021), a multi-point IO formulation which aims to find unknown parameters $\mathbf{A}$ and $\mathbf{b}$ such that $\mathbf{x}^0$ is the optimal observation for the FO formulation, and all $\mathbf{x}^k$, for $k \in \mathcal{K}$ are feasible, is as follows:

$$\mathbf{MIO} : \quad \underset{\mathbf{A}, \mathbf{b}, \mathbf{y}, \mathbf{w}}{\text{minimize}} \quad \mathcal{F}(\mathbf{A}, \mathbf{b}; \mathcal{D}) \quad (4.3a)$$

$$\text{subject to} \quad \mathbf{A}\mathbf{x}^k \geq \mathbf{b}, \quad \forall k \in \mathcal{K}, \quad (4.3b)$$

$$\mathbf{c}'\mathbf{x}^0 = \mathbf{b}'\mathbf{y} + \mathbf{h}'\mathbf{w}, \quad (4.3c)$$

$$\mathbf{A}'\mathbf{y} + \mathbf{G}'\mathbf{w} = \mathbf{c}, \quad (4.3d)$$

$$\|\mathbf{a}'_i\| = 1, \quad \forall i \in \mathcal{I}_1, \quad (4.3e)$$

$$\mathbf{y} \in \mathbb{R}^{m_1}, \quad \mathbf{w} \in \mathbb{R}^{m_2}, \quad (4.3f)$$

$$\mathbf{A} \in \mathbb{R}^{m_1 \times n}, \quad \mathbf{b} \in \mathbb{R}^{m_1}. \quad (4.3g)$$

The objective function, or loss function, of the MIO formulation, $\mathcal{F}(\mathbf{A}, \mathbf{b}; \mathcal{D})$, can be selected to drive the desired properties of the inferred feasible region based on some given input parameter $\mathcal{D}$. For example, a prior belief about the shape of the feasible region can be input as parameter $\mathcal{D}$, with the objective function, $\mathcal{F}$, set to minimize the deviation from this prior belief. Specific examples of loss functions or objective functions for the MIO

formulation are discussed in Section 5.1. Constraint (4.3b) enforces primal feasibility for all observations. Constraint (4.3c) is the strong duality constraint, which ensures $\mathbf{x}^0$ is the optimal solution to the forward problem. Constraints (4.3d) and (4.3f) are dual feasibility constraints. Finally, the normalization constraint, constraint (4.3e), prevents the LHS of each unknown constraint from finding a trivial solution ($\mathbf{a}'_i = \mathbf{0}'$) and prevents redundancy by omitting multiple equivalent solutions (e.g., $x_1 + 2x_2 \geq 4$ is equivalent to $2x_1 + 4x_2 \geq 8$). Norm evaluations such as the $\mathbf{L}_2$ norm, the $\mathbf{L}_1$ norm, or a proxy for the $\mathbf{L}_1$ norm may be used for constraint (4.3e). More details on the normalization constraint are provided in Chapter 6.

The **MIO** formulation (4.3) is nonlinear due constraints (4.3c) and (4.3d) which both include the multiplication of variables. Ghobadi and Mahmoudzadeh (2021) developed a tractable reformulation of the **MIO** formulation (4.3) that eliminates this bilinearity. They proved that by adding the half-space $\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{c}'\mathbf{x} \geq \mathbf{c}'\mathbf{x}^0\}$ to the set of known constraints, that strong duality and dual feasibility conditions, (4.3c) and (4.3d), respectively, are guaranteed. This reduces the complexity of the formulation, leading to an equivalent reformulation of the **MIO** formulation as follows:

$$\text{e-MIO : } \underset{\mathbf{A}, \mathbf{b}}{\text{minimize}} \quad \mathcal{F}(\mathbf{A}, \mathbf{b}; \mathcal{D}) \quad (4.4a)$$

$$\text{subject to } \mathbf{a}'_i \mathbf{x}^k \geq b_i, \quad \forall i \in \mathcal{I}_1, k \in \mathcal{K}, \quad (4.4b)$$

$$\|\mathbf{a}'_i\| = 1, \quad \forall i \in \mathcal{I}_1, \quad (4.4c)$$

$$\mathbf{A} \in \mathbb{R}^{m_1 \times n}, \quad b_i \in \mathbb{R}. \quad (4.4d)$$

In this thesis, this formulation from Ghobadi and Mahmoudzadeh (2021) is considered the nominal formulation with no uncertainty. The robust **IO** formulation will be built from the **e-MIO** formulation in Chapter 5. We elaborate on the objective functions (4.3a) and (4.4a) in Section 5.1 in the next chapter.

If the **e-MIO** formulation were to be applied to infer the constraint parameters of **RT** treatment plans in its current form, the resulting inferred constraints might lack interpretability and be too restricting. The inferred constraints might include features that are not useful to treatment planners or oncologists, they could be overly conservative by considering outlier observations, and they could exclude better **RT** treatment plans from being considered. In the following sections, we will introduce practical considerations required for inferring the feasible region of **RT** treatment plans while using an **IO** approach.

4.2 Pairwise feature solution approach

There are many features that may be considered in an **RT** treatment plan, such as different **DVH** values, index values (e.g., **HI**, **UI** and **CI** values), volumes, minimum doses, maximum doses and mean doses for each **OAR** or target volume. The current guidelines the **GRRCC** use, as well as the RTOG and Quantec guidelines, are all bounds on a single **RT** plan feature. From an optimization perspective, these are understood as one-dimensional constraints.

When solving for the constraint parameters for **RT** treatment plans, it is important to consider who will receive the resulting information. If an **IO** approach is used with no restrictions on the number of features considered in an inferred constraint, it can infer a constraint with up to n -dimensions, which can be hard to interpret. Consider an **IO** formulation, which takes eight plan features. Constraint (4.5) is an example of an eight-dimensional constraint that could be inferred.

$$0.32x_1 + 0.13x_2 + 0.68x_3 + 0.01x_4 + 0.76x_5 + 0.54x_6 + 0.12x_7 + 0.11x_8 \geq 0.85 \quad (4.5)$$

It is difficult to understand the information this constraint is conveying. There are coeffi-

cients of different values on each of the eight plan features, and the constraints specify that the [LHS](#) is greater than or equal to the value 0.85. Moreover, it is difficult to visualize an eight-dimensional constraint using a single plot.

Given that treatment planners or oncologists are some of the primary users of the constraints generated by such a model, the inferred constraints must be easily interpretable. A one-dimensional constraint can be understood as the minimum or maximum value of a single plan feature. This information is considered useful but is trivial to find. More interestingly and less sought after, a two-dimensional constraint may be understood as the trade-off constraint between two plan features. Moreover, a two-dimensional constraint can easily be visualized on a two-dimensional plot. Figure [4.1](#) shows an example of the visualization of a two-dimensional constraint for two random [RT](#) plan features from the [GRRCC](#) guidelines stated in Chapter [3](#).

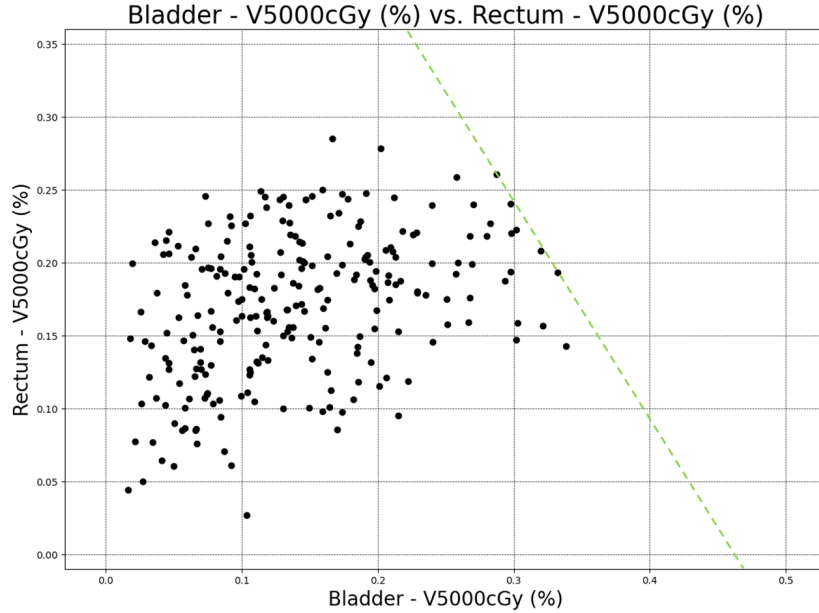


Figure 4.1: An example of the visualization of a two-dimensional constraint

Trade-offs between three [RT](#) plan features, or constraints of three dimensions, are more difficult to understand, and constraints of four dimensions and higher are both difficult to interpret and visualize. Considering interpretability and visualization of the inferred constraints are of such importance, a maximum of two-dimensional constraints would be deemed more acceptable. By solving for a maximum of two-dimensional constraints, the results of the [IO](#) approach are more easily visualized, seen in [Figure 4.1](#), and interpreted by the treatment planners or oncologists.

An added benefit to solving for a maximum of two-dimensional constraints is that many smaller two-dimensional [IO](#) problems may be considered instead of one larger n -dimensional problem. [Figure 4.2](#) shows an example of a three-dimensional space of five observations

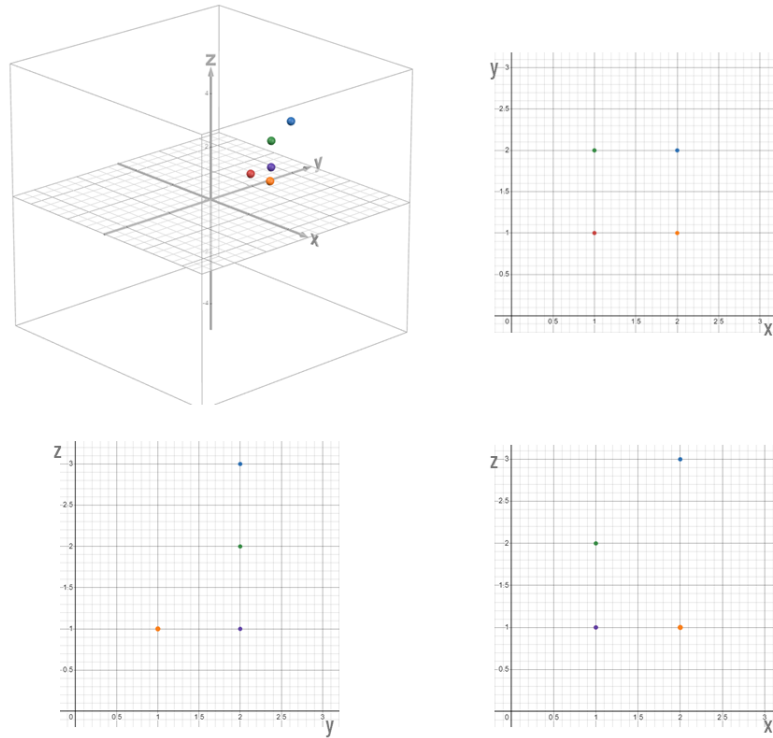


Figure 4.2: A three-dimensional space separated into three two-dimensional spaces

separated into three two-dimensional spaces. The example in Figure 4.2 includes three features: x , y , and z . By separating the three-dimensional space into many two-dimensional spaces by considering the pairs of features, in this case, (x, y) , (y, z) , and (x, z) , one can infer constraints for these two-dimensional spaces, then superimpose the inferred two-dimensional constraints back onto the three-dimensional space. This approach can be generalized to an n -dimensional space where n features for each RT plan are considered. This pairwise feature approach ensures that the constraints inferred from the n -dimensional space are a maximum of two dimensions. Moreover, it decreases the size of the problem, resulting in decreased computational times required to solve for the constraint parameters for RT treatment plans. The remainder of this study will use this pairwise feature solution approach to ensure that a maximum number of two-dimensional constraints are solved.

4.3 Feature selection

Although many features are available to use, it is wise not to use them all. The number of constraints that could be inferred may be very high, and some of the features used in the constraints may not be useful to a treatment planner or oncologist. This raises the question of which features, as well as pairs of features, should be considered.

Initially, 144 features were considered, which included the volume, maximum dose, minimum dose, mean dose, various DVH values (V5Gy to V85Gy in increments of 5), and the HI, UI and CI values for the CTV, PTV, rectum, bladder, left femoral head and right femoral head. The features that had a standard deviation of less than 0.01 were removed as it was assumed that if a feature were a similar value for all of the plans provided in the data, then it would not need to be considered. This resulted in removing 68 of the original 144 features, leaving 76 features left to consider.

One approach to selecting the best feature pairs for evaluating trade-offs is to analyze the correlation between each pair. Figure 4.3 displays a correlation heatmap that shows the correlation value between each pair of the remaining 76 features. Each row and each column corresponds to a single plan feature. The color scale indicates the correlation value where a correlation value of 1.0 is represented as red, and a correlation value of -1.0 is represented as blue. Any correlation value between 1.0 and -1.0 is represented on the color scale on the right of the figure. As seen on the diagonal of the heatmap, each cell is red, indicating a correlation value of 1.0. This is understandable as each feature intersects with itself at the diagonal. The features are grouped together on each axis to show which target volume or OAR they belong to. The feature pairs within the black boxes in Figure 4.3 are not considered as potential pairs of features to use in an IO approach. The reasoning for this is discussed later in this section.

Using the correlation matrix heatmap from Figure 4.3, it can be seen that the features Rectum - V20Gy (%) and Bladder - V25Gy (%) have a high positive correlation of approximately 0.75. When looking at the two features individually, the values for these features are considered better if they are lower (lower radiation to the bladder and rectum is preferred). With the knowledge that both features are to be minimized, the trade-off constraint would be in the upper right of the scatter plot of the two features.

Figure 4.4 shows the scatter plot of the pair of features Rectum - V20Gy (%) and Bladder - V25Gy (%) as well as a potential trade-off constraint in the top right of the plot. Looking at the raw data of the two features, it can be seen that the potential trade-off constraint is not very useful. It does not remove very much feasible region from the top right corner of the data. The result in Figure 4.4 shows that correlation may not be the best indication of whether a pair of features has a better potential trade-off constraint or not. In the case of two features that are to be minimized, it might be tempting to say

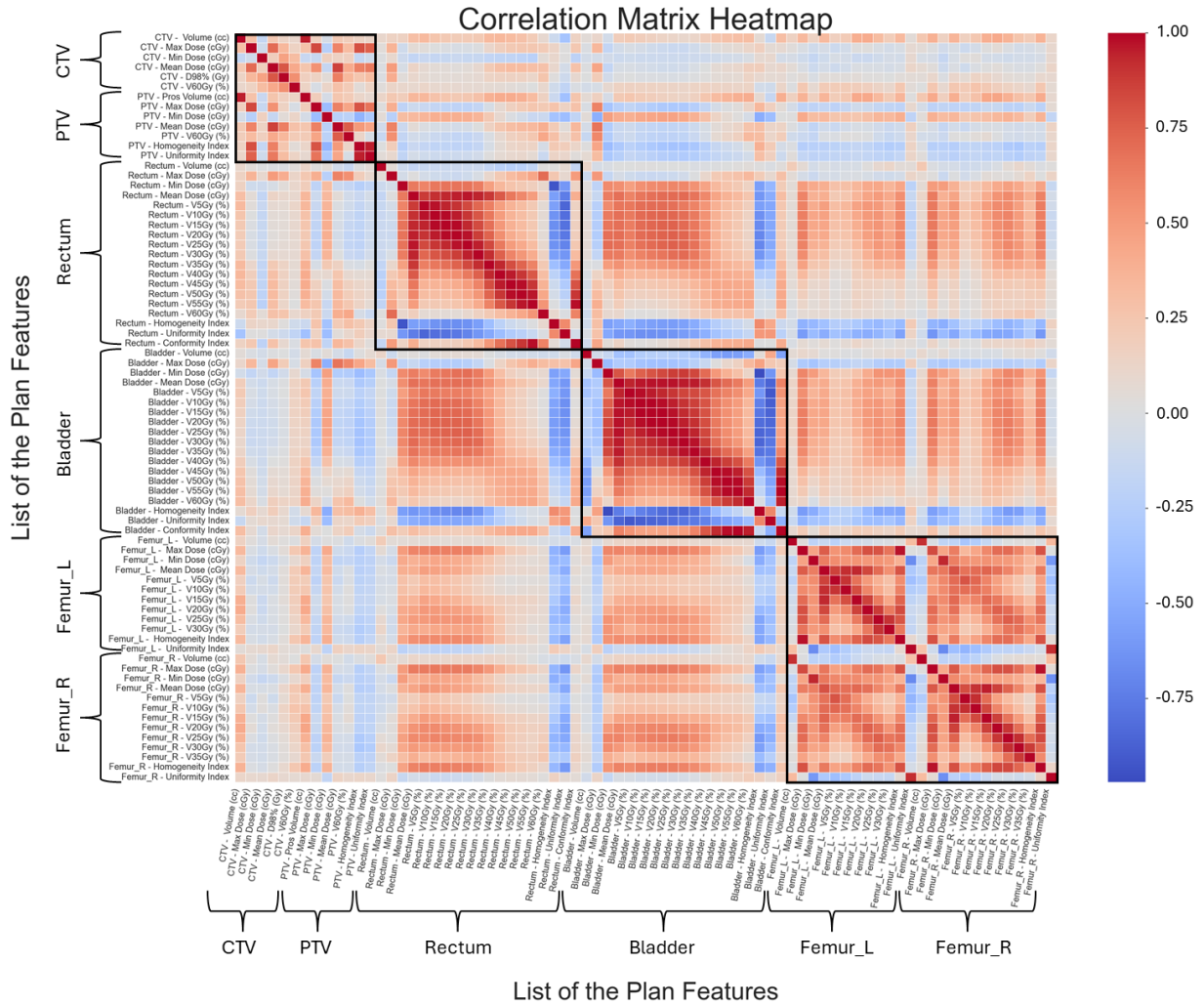


Figure 4.3: Correlation matrix heatmap of RT treatment plan features

a negative correlation could indicate a good trade-off constraint; however, this might not be the case as a scatter plot in the shape of a triangle spanning from the top left to the bottom right to the bottom left corner could have a good trade-off constraint but a bad correlation value.

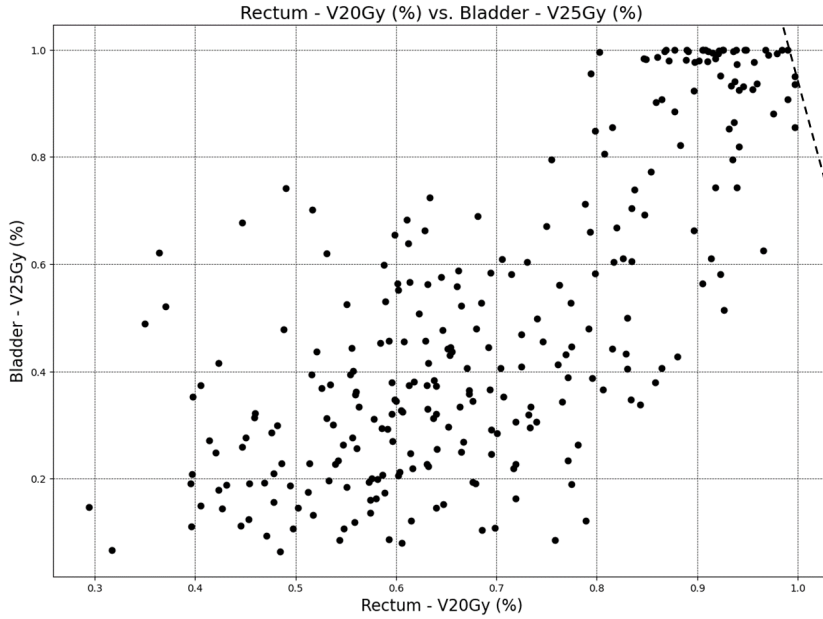


Figure 4.4: Rectum - V20Gy (%) vs. Bladder - V25Gy (%) with a potential trade-off constraint

Alternatively, one could visually inspect the two-dimensional scatter plots of all possible feature pairs to determine if a pair of features has good trade-off constraints, but there are an infinite number of features to be explored as the DVH values may be fractional (e.g., Bladder - V20.25Gy (%)).

Based on these results, we determined that the best course of action for choosing features to use in an IO approach to infer trade-off constraints was to leverage the features in the current dose goals the GRRCC use from Table 3.2. This decision was made because the treatment planners and oncologists were already accustomed to using these features to assess an RT treatment plan. Additionally, utilizing these features will enable GRRCC to identify necessary updates to their current dose goals and explore interesting trade-offs among the features they currently use in those goals.

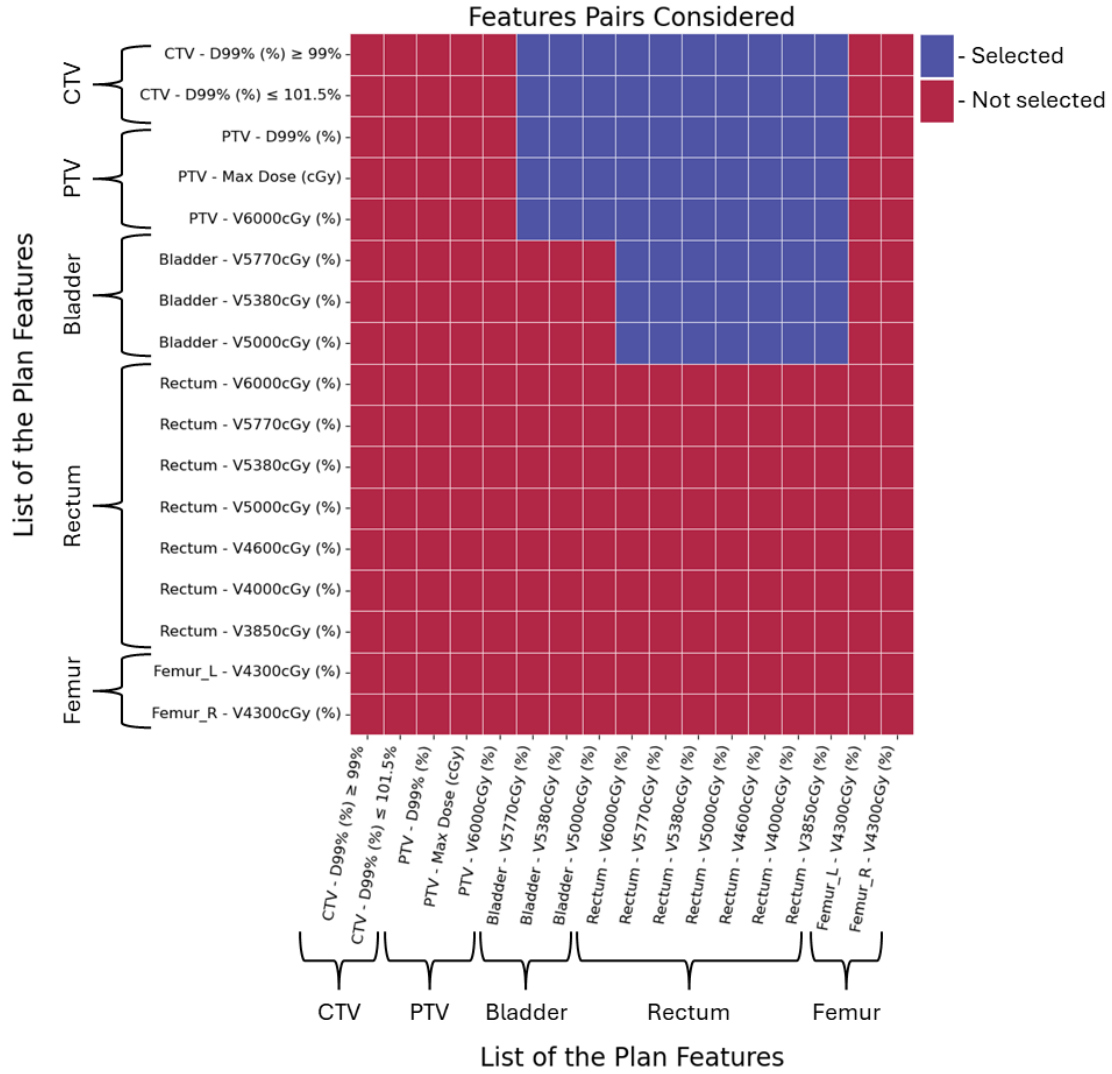


Figure 4.5: The feature pairs selected (blue) and not selected (red)

Only a subset of the pairs of features from the dose goals from Table 3.2 were considered. Figure 4.5 shows the pairs of features that were selected in blue and the pairs that were not in red. The features are also grouped into their respective target volume or OAR on each axis. Features from the same OAR or target volume were not considered, as well as

PTV and CTV pairs. These pairs of features were not considered as it does make sense to consider a trade-off between two features of the same target volume or OAR. The pairs between the PTV and CTV features were not considered as the PTV is the physical area around the CTV. The left and right femur features were omitted as the standard deviation of the features was less than 0.01. In addition to these pairs that were omitted, it must also be noted that the trade-off constraints imputed from the pair of features (x, y), for example, would be the same trade-off constraints imputed for the pair of features (y, x). Therefore, only one set of these duplicate pairs was considered (the upper triangular matrix). With all these pair restrictions in mind, a total of 71 pairs of features were considered. Considering these pairs of RT plan features makes sense, as these plan features are explicitly referenced in the volume and dose constraints used by the GRRCC (as seen in Table 3.2), and there may be implicit logic (trade-offs) applied by the oncologist when requesting adjustments to the RT treatment plans.

4.4 Outlier detection

As previously discussed in Chapter 3 and illustrated in Figure 4.6 below, there are potential outliers in the data (indicated by the red arrows in the figure). There are many reasons why outliers may exist in the data of accepted RT treatment plans. For instance, an RT plan could be an outlier due to a patient who has diabetes as they are required to have adjusted RT plans. Unfortunately, patient information for each RT plan collected was not available, so this type of information could not be used to identify such outliers, and other methods were explored.

Before applying an outlier detection algorithm, each feature must be individually assessed, and the direction of optimization must be specified. The direction of optimization

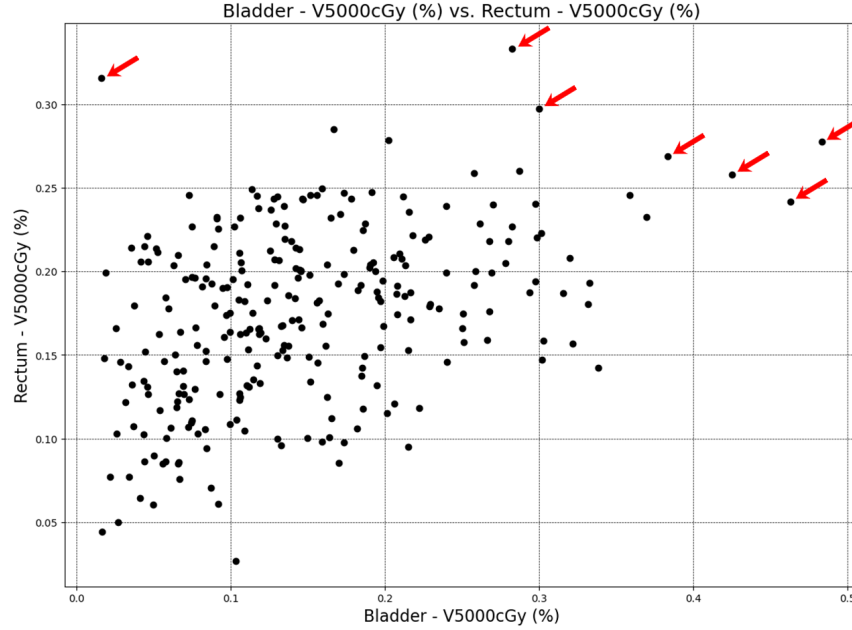


Figure 4.6: Bladder - V 5000 cGy (%) vs. Rectum - V 5000 cGy (%) scatter plot with potential outliers indicated by the red arrows

can be either to minimize or maximize a feature value. Using the current dose goals in Table 3.2, it can easily be seen which directions are preferred for each feature. Table 4.1 shows the directions of optimization for each dose goal which the GRRCC used to create the 273 RT plans in the data collected. Note that there is one plan feature, D 99.0%, which is used in two separate dose constraints in the first and second rows of Table 4.1. In this case, we consider this plan feature to have a direction of optimization of both minimize and maximize. We explain how we deal with this particular case in detail later in this section. Also, note that the directions for the left and right femoral head optimization are not included in this table. The reasoning for this removal of these features is mentioned in Section 4.3.

Table 4.1: The direction of optimization for each RT treatment plan feature

Target volume/OAR	Dose constraint	Direction of Optimization
CTV	D 99.0% $\geq$ 99.0%	maximize $\uparrow$
	D 99.0% $\leq$ 101.5%	minimize $\downarrow$
PTV	D 99.0% $\geq$ 95.0%	maximize $\uparrow$
	Dmax $\leq$ 105.0%	minimize $\downarrow$
	V 6000 cGy $\geq$ 75.0%	maximize $\uparrow$
Rectum	V 6000 cGy $\leq$ 1.0%	minimize $\downarrow$
	V 5770 cGy $\leq$ 15.0%	minimize $\downarrow$
	V 5380 cGy $\leq$ 20.0%	minimize $\downarrow$
	V 5000 cGy $\leq$ 22.0%	minimize $\downarrow$
	V 4600 cGy $\leq$ 35.0%	minimize $\downarrow$
	V 4000 cGy $\leq$ 38.0%	minimize $\downarrow$
	V 3850 cGy $\leq$ 50.0%	minimize $\downarrow$
Bladder	V 5770 cGy $\leq$ 15.0%	minimize $\downarrow$
	V 5380 cGy $\leq$ 35.0%	minimize $\downarrow$
	V 5000 cGy $\leq$ 40.0%	minimize $\downarrow$

Three different types of outlier detection algorithms were considered. They are described as follows:

- **Top X%:** The top X% outlier detection algorithm considers an observation to be an outlier if the feature value is in X% of the worst observations.

- **Standard Deviation:** The standard deviation outlier detection algorithm considers an observation to be an outlier if the feature value is beyond the mean of the feature distribution $\pm$ some constant value, Z , times the standard deviation given the direction of optimization of the feature is minimize/maximize.
- **Interquartile Range:** The interquartile range (IQR) outlier detection algorithm considers an observation to be an outlier if the value is beyond the mean of the feature distribution $\pm$ some constant value, Z , times the IQR given the direction of optimization of the feature is minimize/maximize.

The standard deviation outlier detection algorithm was chosen and will be used for all outlier detection throughout this study. The constant value multiplied by the standard deviation was decided to be 2.5. We chose the value of 2.5 as it removed all the obvious outliers that were noted by the experts at the [GRRCC](#). The outlier detection algorithm chosen and the constant value multiplied by the standard deviation in the outlier detection algorithm (2.5) may be interchanged in the solution approaches presented in this study. The outlier detection presented here is not the primary contribution of this study. Rather, the outlier detection was used to aid in the demonstration of the quality-based robust-inverse optimization solution approach, which will be discussed in later chapters.

Note that if the feature values were assumed to follow a normal distribution, only the worst 0.62% of the data would be flagged as outliers using the standard deviation outlier detection algorithm with a value of 2.5. As mentioned previously, one plan feature was specified to have an optimization direction of both minimize and maximize. For this feature, we applied the outlier detection algorithm in both directions.

The standard deviation outlier detection algorithm was run against all features, and 30 out of the 273 [RT](#) treatment plans included at least one feature value that was considered an

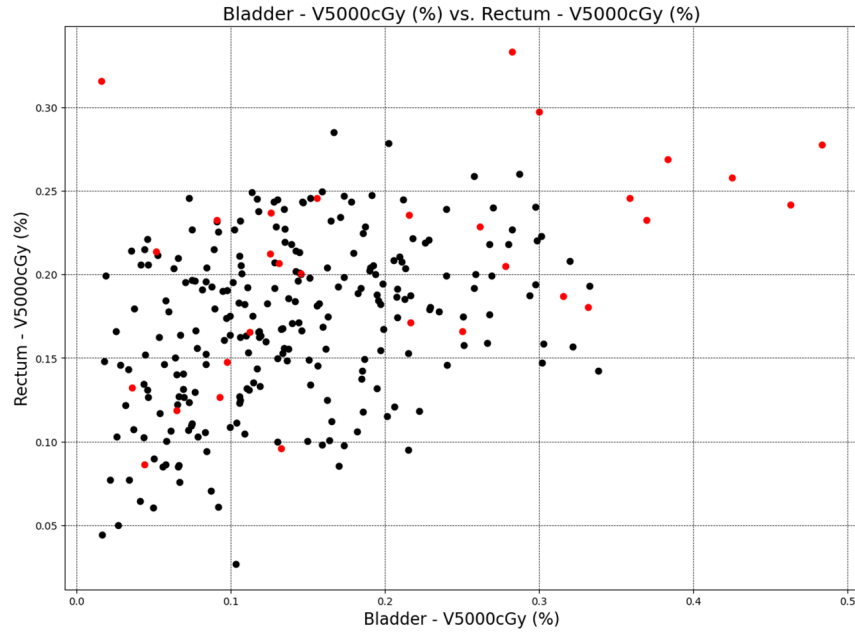


Figure 4.7: Bladder - V 5000 cGy (%) vs. Rectum - V 5000 cGy (%) scatter plot with outliers (red)

outlier. These 30 plans were removed from the set of observations used to infer constraints. Figure 4.7 displays the same observations as Figure 4.6, but with the outliers identified in red. In Figure 4.7, both features have a minimized direction of optimization. It is clear why the observation at the top or right of the plot would be considered outliers, but it is less apparent why some observations in the middle of the data distribution are also classified as outliers. The reason for this is that these observations are outliers for a different RT plan feature. One can imagine adding a third dimension to the set of observations by considering a third feature plan. In this new dimension, certain observations might be outliers for this additional feature, even if they are not outliers for the two features shown in Figure 4.7.

4.5 Preventing elimination of high-quality plans

When inferring constraints for RT treatment plans, it may not be wise to infer constraints all the way around the observations for each pair of features. Figure 4.8 shows a set of potential constraints that could be inferred for a particular pair of features. The figure demonstrates that some of the potential inferred constraints restrict the possibility of considering better RT plans in the future.

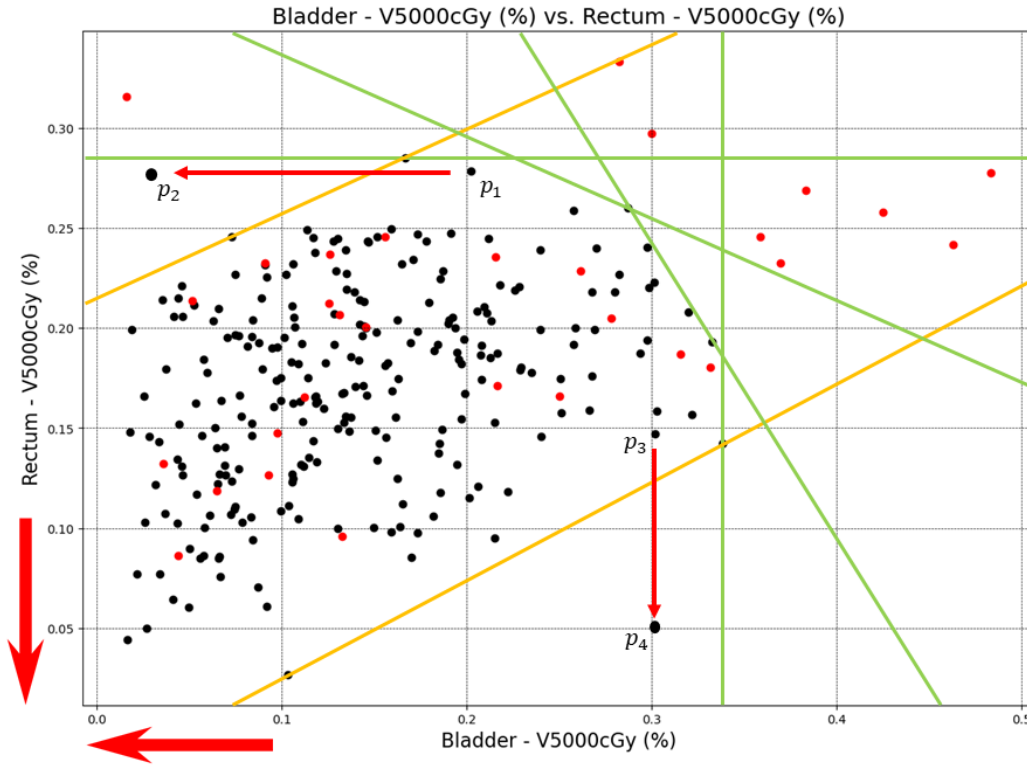


Figure 4.8: An example of preferred constraints (green), undesirable constraints (orange), and two hypothetical observations (p_2 and p_4) to motivate corner restricting constraints

As seen in Figure 4.8, there are five potential inferred constraints for the feature pair Bladder - V 5000 cGy (%) and Rectum - V 5000 cGy (%), and two hypothetical observations, p_2 and p_4 . Referencing Table 4.1, both features have a minimization direction of optimization. This is visualized by the red arrows in the bottom left of the figure. Without additional restrictions for inferring constraints around the feature pair, the five constraints shown in Figure 4.8 could be inferred. Focusing on the orange constraint at the top-left of the figure, it is shown that it excludes the hypothetical observation p_2 . It is evident that p_2 has the same Rectum - V 5000 cGy (%) value as p_1 , but a lower (better) Bladder - V 5000 cGy (%) value than p_1 . This means that p_2 dominates p_1 and should not be excluded from being a potential observation in the future. A similar statement can be made for p_4 and p_3 , showing p_4 dominates p_3 . In Figure 4.8, the green constraints do not cause this issue of excluding better RT treatment plans, whereas the orange constraints do. This motivates the case for restricting the constraints inferred for RT treatment plans.

In the example above, since both features have a minimization direction of optimization, only constraints in the top right of the observations should be inferred. If one or both of the features had a maximization direction of optimization, the inferred constraints would need to be positioned in a different corner of the observations. This can be understood using the same logic as described earlier. To derive the restrictions that need to be applied to the LHS parameters of the inferred constraints to focus the constraints on a specific corner of the observations, some notation must be defined. Let a_1 and a_2 be the constraint coefficients of a given constraint for the feature on the x-axis (x_1) and y-axis (x_2), respectively. Let b be the RHS value for that constraint. Assume the constraint would be as follows:

$$a_1x_1 + a_2x_2 \geq b. \quad (4.6)$$

In the example in Figure 4.8 where features in the x-axis and y-axis both have a minimization direction of optimization, the two extreme constraints that may be considered are the

horizontal constraint and vertical constraint in the following forms:

$$-1 \times x_1 + 0 \times x_2 \geq b, \quad (4.7)$$

$$0 \times x_1 - 1 \times x_2 \geq b. \quad (4.8)$$

Note that the [RHS](#) value b is not restricted to be non-negative. Any constraint that falls between these two constraints (including the constraint shapes themselves) is preferred for constraint inference for [RT](#) treatment planning as these constraints will not exclude better [RT](#) treatment plans from being considered (the green constraints in Figure 4.8). With this information in mind, the following restrictions on the a_1 and a_2 constraint parameters can be added to the [IO](#) formulation:

$$-1 \leq a_1 \leq 0, \quad (4.9)$$

$$-1 \leq a_2 \leq 0. \quad (4.10)$$

In the general case, the restrictions required for the [LHS](#) constraint parameter for feature $j \in \mathcal{J}$ of constraint $i \in \mathcal{I}_1$ where feature j has a minimization direction of optimization is as follows:

$$-1 \leq a_{ij} \leq 0. \quad (4.11)$$

By applying the same derivation process as outlined earlier, we can also determine the restrictions on the [LHS](#) parameter for a feature with a maximization direction of optimization. Conversely, the restrictions required for the [LHS](#) constraint parameter for feature $j \in \mathcal{J}$ of constraint $i \in \mathcal{I}_1$ where feature j has a maximization direction of optimization is as follows:

$$0 \leq a_{ij} \leq 1. \quad (4.12)$$

In some cases, one may want to infer constraints that do not include the extreme cases of the constraints, such as one-dimensional constraints (vertical and horizontal constraints). In this case, the constraints (4.11) and (4.12) may be adjusted to be tighter by considering strict inequality signs. This would cause the IO formulation to only infer two-dimensional constraints between the two extreme cases (the horizontal and vertical constraints).

4.6 Summary

Chapter 4 outlines the specific practical considerations necessary when applying an Inverse Optimization (IO) approach to Radiation Therapy (RT) treatment plans. This chapter introduces the IO model for a linear FO problem, presents a pairwise feature solution approach to increase interpretability, and discusses the features and feature pairs that will be used in the pairwise feature solution approach for the following IO approaches in this thesis. Additionally, the outlier detection algorithm used throughout the rest of the study is presented, along with additional constraints, to ensure the imputed constraints do not preclude the consideration of better RT treatment plans.

Thus far, we have demonstrated the required considerations for applying IO techniques to RT treatment plans. Using these methodologies, we have laid a foundation for developing a more comprehensive optimization model that incorporates quality-based uncertainty. The following chapters will build upon these ideas by developing a quality-based Robust-inverse Optimization (RIO) formulation, applying various loss functions, and evaluating their performance.

Chapter 5

Proposed Quality-based Robust-inverse Optimization Formulation

This chapter discusses the proposed quality-based [Robust-inverse Optimization \(RIO\)](#) formulation. Section [5.1](#) explores two potential loss functions and explains the effects the objective functions have on the imputed constraints. Section [5.2](#) follows with the development of the general [RIO](#) formulation, building on the [IO](#) formulation discussed in the previous chapter. Lastly, Section [5.3](#) introduces the quality-based [RIO](#) formulation and details its implementation.

As mentioned in the previous chapter, the notation for this paper is as follows. All vectors are column vectors, denoted as bold lowercase letters. Scalars are denoted as regular text lowercase letters. Matrices are denoted as bold uppercase letters. Sets are denoted as calligraphic uppercase letters. All inequalities are row-wise.

5.1 Loss functions

The objective function, or loss function, of the **e-MIO** formulation discussed in the previous chapter, formulation (4.4), drives the desired properties of the inferred feasible region. This means the loss function directly affects the constraints being inferred. For instance, the loss function can aim to have all observations roughly the same total distance from all constraints, or it could minimize the minimum distance of each observation from all the constraints. Moreover, the complexity of the formulation can be increased by requiring additional decision variables (e.g., continuous, binary, or integer) to enforce the logic of the loss function. Some loss functions leverage a prior belief of the shape of the feasible region and set the objective function to minimize the deviation from such prior belief (Chan et al., 2023). This study does not incorporate prior belief of the shape of the feasible region, as two-dimensional constraints for RT treatment planning are neither currently considered nor available. Ghobadi and Mahmoudzadeh (2021) discussed two linear loss functions that rely on no prior belief that are relevant to this study.

- **Fairness Measure:** The fairness measure minimizes the deviation of the total distance for all observations and aims to have all observations roughly the same total distance from all constraints. Let d_{ik} be defined as the distance between observation $\mathbf{x}^k$, for $k \in \mathcal{K}$ and the hyperplane of the i^{th} constraint, for $i \in \mathcal{I}_1$. Let $d_k = \sum_{i \in \mathcal{I}_1} d_{ik}$, where d_k is the total distance for observation $k \in \mathcal{K}$. Let $|\mathcal{K}|$ be the number of observations. The fairness measure is:

$$\mathcal{F}(\mathbf{A}, \mathbf{b}; \mathcal{D}) = \sum_{k \in \mathcal{K}} (d_k - \sum_{k \in \mathcal{K}} d_k / |\mathcal{K}|) \quad (5.1)$$

- **Compactness Measure:** The compactness measure minimizes the minimum distance of each observation from all constraints. The compactness measure is:

$$\mathcal{F}(\mathbf{A}, \mathbf{b}; \mathcal{D}) = \sum_{k \in \mathcal{K}} \min_{i \in \mathcal{I}_1} d_{ik} \quad (5.2)$$

Different loss functions produce varying optimal solutions for the **e-MIO** formulation, resulting in distinct inferred feasible regions for the forward problem. In this study, we will employ both the fairness measure and compactness measure loss functions in the upcoming formulations, with their performance and outcomes thoroughly analyzed.

5.2 Proposed Robust-inverse optimization formulation

From an **IO** perspective, past decisions are considered as input data points. When evaluating these data points, it is essential to consider their accuracy. Leveraging concepts from robust optimization studies (Ben-Tal et al., 2009), we can define an uncertainty set that encapsulates all potential realizations of the input data.

Building upon the **e-MIO** formulation from the previous chapter, formulation (4.4), we define an inverse optimization formulation that is robust to the uncertainty of the observations. This model is defined as the **RIO** formulation, and it is as follows:

$$\mathbf{RIO} : \quad \underset{\mathbf{A}, \mathbf{b}}{\text{minimize}} \quad \mathcal{F}(\mathbf{A}, \mathbf{b}; \mathcal{D}) \quad (5.3a)$$

$$\text{subject to} \quad \sum_{j \in \mathcal{J}} a_{ij} x_j^k \geq b_i, \quad \forall i \in \mathcal{I}_1, k \in \mathcal{K}, x_j^k \in \mathcal{U}_j^k, \quad (5.3b)$$

$$\|\mathbf{a}'_i\| = 1, \quad \forall i \in \mathcal{I}_1, \quad (5.3c)$$

$$b_i \in \mathbb{R}, \quad \forall i \in \mathcal{I}_1, \quad (5.3d)$$

$$a_{ij} \in \mathbb{R}, \quad \forall i \in \mathcal{I}_1, j \in \mathcal{J}. \quad (5.3e)$$

Many different types of uncertainty sets may be considered. We will use a box uncertainty set to keep the model linear (depending on the complexity of constraint (5.3c)). This means that $\mathcal{U}_j^k = [\underline{x}_j^k, \overline{x}_j^k]$, for all $j \in \mathcal{J}$ and $k \in \mathcal{K}$, where $\underline{x}_j^k$ and $\overline{x}_j^k$ are known parameters. Figure 5.1 displays an example of a box uncertainty applied to a set of two-dimensional observations that we generated. The box uncertainty set for a single observation, say (x_1^1, x_2^1) , considers all realizations of the observation for $\underline{x}_1^1 \leq x_1^1 \leq \overline{x}_1^1$ and $\underline{x}_2^1 \leq x_2^1 \leq \overline{x}_2^1$. As seen in Figure 5.1, this results in a box-shaped uncertainty set around each observation.

Constraint (5.3b), $\sum_{j \in \mathcal{J}} a_{ij} x_j^k \geq b_i$, in the **RIO** formulation above may be rewritten by considering the nominal (without uncertainty) value of x_j^k and uncertain component of x_j^k separately. We rewrite the constraint as $\sum_{j \in \mathcal{J}} a_{ij} x_j^k + B_i(x_1^k, \dots, x_n^k) \geq b_i$, where the $B_i(x_1^k, \dots, x_n^k)$ term represents the uncertain component of constraint $i \in \mathcal{I}_1$. Since we require the inferred constraint parameters to be feasible for all realizations of each observation, we would like to consider the worst-case scenario for each observation. This means we would like to find the lowest possible value of the $B_i(x_1^k, \dots, x_n^k)$ term in order to ensure the constraint parameters we infer are feasible to all possible realizations of each observation. Formulation (5.3) has infinitely many constraints (in constraint (5.3b)) but

can be rewritten in the equivalent **e-RIO** formulation (Ben-Tal et al., 2009):

$$\textbf{e-RIO} : \quad \underset{\mathbf{A}, \mathbf{b}}{\text{minimize}} \quad \mathcal{F}(\mathbf{A}, \mathbf{b}; \mathcal{D}) \quad (5.4a)$$

$$\text{subject to} \quad \sum_{j \in \mathcal{J}} a_{ij} x_j^k + B_i(x_1^k, \dots, x_n^k) \geq b_i, \quad \forall i \in \mathcal{I}_1, k \in \mathcal{K}, \quad (5.4b)$$

$$(5.3c) - (5.3e). \quad (5.4c)$$

Where,

$$B_i(x_1^k, \dots, x_n^k) = \underset{\hat{\mathbf{x}}}{\text{minimize}} \quad \sum_{j \in \mathcal{J}} a_{ij} \hat{x}_j^k \quad (5.5a)$$

$$\text{subject to} \quad \hat{x}_j^k \leq \overline{x}_j^k, \quad \forall j \in \mathcal{J}, \quad (5.5b)$$

$$\hat{x}_j^k \geq \underline{x}_j^k, \quad \forall j \in \mathcal{J}, \quad (5.5c)$$

$$\hat{x}_j^k \in \mathbb{R}, \quad \forall j \in \mathcal{J}. \quad (5.5d)$$

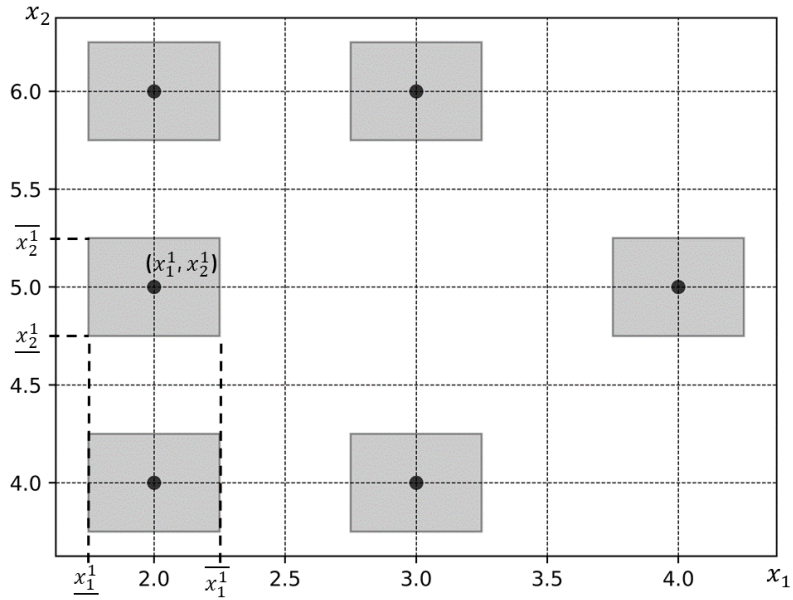


Figure 5.1: An example of a two-dimensional problem with box uncertainty

As stated previously, formulation (5.3) has an infinite number of constraints. However, by using the theory of linear programming duality, we can construct an equivalent finite constraint formulation. Now we interpret the reformulated problem (5.4) (5.5). In formulation (5.4), we try to solve for the a_{ij} and b_i constraint parameters for the forward model such that they are feasible to the worst possible outcome of the x_j^k values. Formulation (5.5) represents the additional LHS value obtained from the worst-case variation for inferred constraint $i \in \mathcal{I}_1$ with the $B_i(x_1^k, \dots, x_n^k)$ term. This additional LHS value can be negative or positive depending on the direction of the realized constraint of the forward problem in relation to the observation. By writing constraint (5.4b) in this manner and considering the results of formulation (5.5), this produces a model equivalent to formulation (5.3).

We introduce two new sets of dual variables, $\mathbf{q}$ and $\mathbf{r}$, for each constraint of the subproblem, (5.5b) and (5.5c), respectively. The j^{th} elements, for $j \in \mathcal{J}$, of the $\mathbf{q}$ and $\mathbf{r}$ dual vectors are q_j and r_j , respectively. The dual of formulation (5.5) is:

$$(Dual) \ B_i(x_1^k, \dots, x_n^k) = \underset{\mathbf{q}, \mathbf{r}}{\text{maximize}} \quad \sum_{j \in \mathcal{J}} q_j \overline{x_j^k} + \sum_{j \in \mathcal{J}} r_j \underline{x_j^k} \quad (5.6a)$$

$$\text{subject to} \quad q_j + r_j = a_{ij}, \quad \forall j \in \mathcal{J}, \quad (5.6b)$$

$$q_j \leq 0, \quad r_j \geq 0, \quad \forall j \in \mathcal{J}. \quad (5.6c)$$

Replacing this $B_i(x_1^k, \dots, x_n^k)$ into formulation (5.4), we derive the tractable robust counterpart formulation, **e-RIO-RC**, as follows:

$$\mathbf{e}\text{-RIO-RC} : \quad \underset{\mathbf{A}, \mathbf{b}, \mathbf{q}, \mathbf{r}}{\text{minimize}} \quad \mathcal{F}(\mathbf{A}, \mathbf{b}; \mathcal{D}) \quad (5.7a)$$

$$\begin{aligned} \text{subject to} \quad & \sum_{j \in \mathcal{J}} a_{ij} x_j^k + \left[\sum_{j \in \mathcal{J}} q_{ij} \overline{x_j^k} + \sum_{j \in \mathcal{J}} r_{ij} \underline{x_j^k} \right] \geq b_i, \\ & \forall i \in \mathcal{I}_1, k \in \mathcal{K}, \end{aligned} \quad (5.7b)$$

$$q_{ij} + r_{ij} = a_{ij}, \quad \forall i \in \mathcal{I}_1, j \in \mathcal{J}, \quad (5.7c)$$

$$q_{ij} \leq 0, \quad r_{ij} \geq 0, \quad \forall i \in \mathcal{I}_1, j \in \mathcal{J}, \quad (5.7d)$$

$$(5.3c) - (5.3e). \quad (5.7e)$$

Formulation (5.7) is the robust counterpart of the robust problem (5.3). We have introduced new decision variables and constraints compared to the nominal **e-MIO** formulation, formulation (4.4). We omit the proof of equivalence as it is covered elsewhere ([Ben-Tal et al., 2009](#)).

5.2.1 Robust-inverse optimization implementation

An alternative approach to formulation (5.7) for solving the **RIO** model is to add all the corner points of the polyhedral uncertainty set as observations in the **e-MIO** model. By paying close attention to formulation (5.5), it can be seen that an optimal solution for an n -dimensional problem must have at least n binding constraints (with the assumption that these constraints are linearly independent). There are n variables and $2n$ constraints in formulation (5.5). This means that for any constraint inferred for the forward problem, there must be at least one corner point of each polyhedral uncertainty set, representing the worst-case variation for each observation. Figure 5.2 visualizes this concept with an example of a two-dimensional problem with six observations that we generated. With this

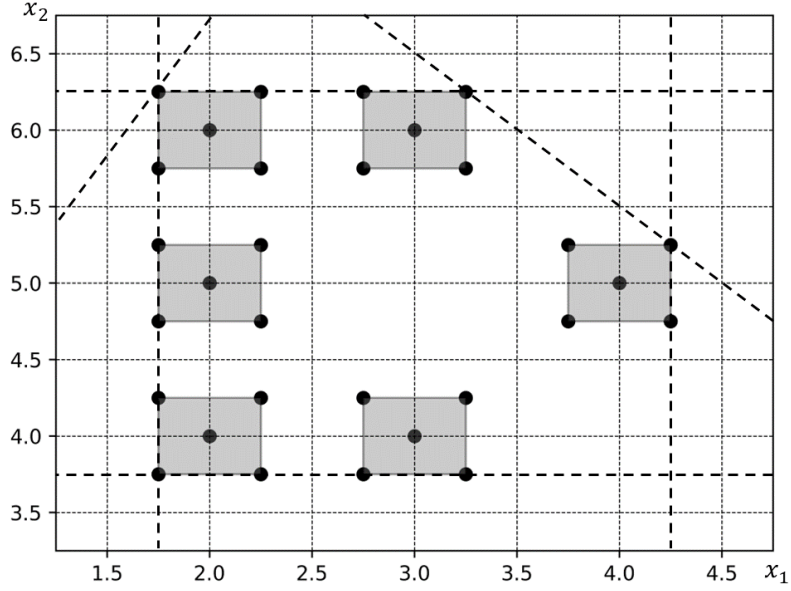


Figure 5.2: An example of a two-dimensional problem with the box uncertainty corner points as observations and potential inferred constraints

information in mind, it is possible to consider the corner points of each box uncertainty set as observations themselves to ensure the formulation is robust to the worst-case variation of each observation $\mathbf{x}^k$ for $k \in \mathcal{K}$. In other words, by considering the corner points of the box uncertainty set as observations, the convex hull of the corner points (the entire box uncertainty set) must be within the inferred feasible region of the [FO](#) problem.

This box uncertainty set corner point method ensures that every realization of each observation is within the inferred feasible region while using formulation (4.4). This results in a robust [IO](#) formulation that has $|\mathcal{I}_1| \times |\mathcal{J}|$ fewer decision variables and $|\mathcal{I}_1| \times |\mathcal{J}|$ fewer constraints than formulation (5.7) as there is no need for the q_{ij} and r_{ij} dual decision variables. Moreover, it allows for the possibility to access these corner points more easily in later formulations, as seen in Chapter 8 in the proposed iterative single-constraint inference

algorithm. One disadvantage, however, is that the number of observations increases from $|\mathcal{K}|$ to $2^n \times |\mathcal{K}|$ (or $(2^n + 1) \times |\mathcal{K}|$ depending on if you still consider the original observations) for an n -dimensional problem. Considering the corner points of the box uncertainty sets as observations also increases the number of primal feasibility constraints, (4.4b), from $|\mathcal{I}_2| \times |\mathcal{K}|$ to $2^n \times |\mathcal{I}_2| \times |\mathcal{K}|$ (or $(2^n + 1) \times |\mathcal{I}_2| \times |\mathcal{K}|$ depending on if you consider the original observations) constraints in the **e-MIO** formulation.

We explored three different methods to integrate the corner points of the uncertainty sets into the **e-MIO** formulation without considering the robust counterpart:

1. **All Corner Points Approach:** Solve the problem while considering all corner points as observations.
2. **Constraint Generation Approach:** Solve a master problem considering only the original observations. Then, use a subproblem to identify the corner points that violate the constraints inferred by the master problem, add these to the set of constraints in the master problem, and resolve the master problem, similar to [Mahmoudzadeh et al. \(2016\)](#).
3. **Limited Constraint Generation Approach:** The same process as the previous approach, but instead of adding all the corner points that violate the inferred constraints, we add a limited number of corner points at each iteration and resolve. Repeat this step until all corner points adhere to the inferred constraints.

The first approach is suitable when the number of observations and the dimension of the problem are relatively small. However, if there are many observations or the problem's dimensionality is high, the second or third approach may be preferred to reduce the number of observations that need to be considered. During the remainder of this study, the first

approach will be used. A new set of observations will be considered, $\mathcal{K}_2$, which denotes the set of all corner points of each box uncertainty set. Considering $\mathcal{K}_2$ instead of $\mathcal{K}$ in the formulations developed in the following chapters ensures that the formulations are robust to all possible realizations of each observation within their corresponding uncertainty set.

A noteworthy outcome of using the corner point approach is that it can accommodate any type of polyhedral uncertainty set, not only the box uncertainty set. By treating the corner points of any polyhedral uncertainty set as observations themselves, we ensure that the convex hull of these corner points lies within the inferred feasible region. In this study, the box uncertainty set will be utilized, as the concept of an [RT](#) treatment plan feature having an uncertainty of $\pm$ some value for each feature is more clinically interpretable. This leads to the formation of a box uncertainty set around each observation.

5.3 Quality-based robust-inverse optimization formulation

When determining the size of the uncertainty area around an observation, it may seem intuitive to apply the same level of uncertainty to all observations. However, as shown in [Figure 5.3](#), this approach in the context of the [IO](#) problem, leads to a similar feasible region shape as for observations without uncertainty, but slightly larger in all directions.

In addition to the previous argument, it is important to consider that expanding the inferred feasible region toward higher-quality plans while limiting expansion toward lower-quality plans is ideal. This can be achieved by considering a quality-based uncertainty size around the observations. In doing so, the inferred feasible region will expand in the direction of higher-quality plans and contract in the direction of lower-quality plans.

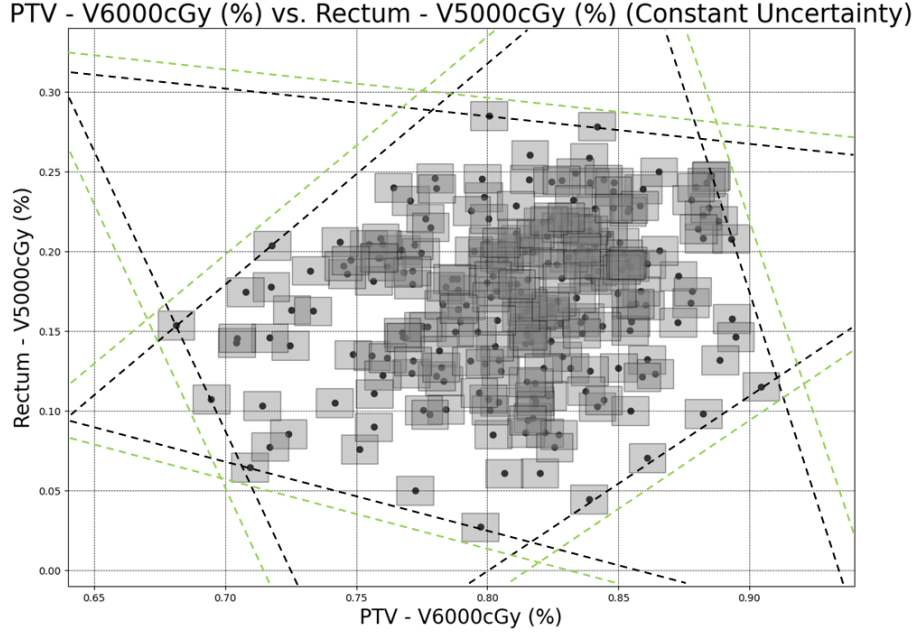


Figure 5.3: Inferred constraints without uncertainty (black) and with constant uncertainty size (green)

Moreover, based on the [IO](#) model in Section 4.1, it is assumed that the optimal solution is within the set of observations. By enlarging the uncertainty set around higher-quality plans, the likelihood of including the optimal observation within the inferred feasible region may increase.

To explain the quality-based robust-inverse solution approach, we assume that outliers have been identified and removed from the data. Additionally, we assume that corner-restricting constraints, which prevent the elimination of high-quality plans, are not enforced and that the corner points of the box uncertainty sets are not considered, as these concepts are not necessary for the explanation in this section.

To apply a quality-based uncertainty size around each observation, the quality metric of all observations can be normalized so that the best quality observation is set to 1.0 and the worst quality observation is set to 0. This normalized quality metric can then be multiplied by a specified maximum uncertainty size. This method varies the size of the uncertainty set based on the quality of the observation. The normalization function in this study can be seen in Equation (5.8) below. The maximum uncertainty size for each RT plan feature has been discussed previously and is stated in Table 3.3 of Section 3.3. It is important to note that the quality value of an observation was assumed to be better if it was lower. If this were not the case, then the normalization function would need to be adjusted accordingly.

$$\text{Normalized Quality Value } (x) = \frac{X_{max} - x}{X_{max} - X_{min}} \quad (5.8)$$

The X_{max} term seen in Equation (5.8) represents the largest quality value from the set of observations. Conversely, the X_{min} term represents the lowest quality value from the set of observations. Finally, the x variable represents the quality value of a given observation. It is evident that the best quality (lowest value = X_{min}) observation would result in a 1.0, whereas the worst quality (highest value = X_{max}) would result in a 0, and observations between these two would result in a number between 1.0 and 0 accordingly.

Figure 5.4 shows the new potential feasible region when considering a quality-based uncertainty for each observation. It's important to note that the definition of RT treatment plan quality used in this study is not the only measure of quality for an RT plan. A different measure of plan quality could be used in place of the one specified in this study.

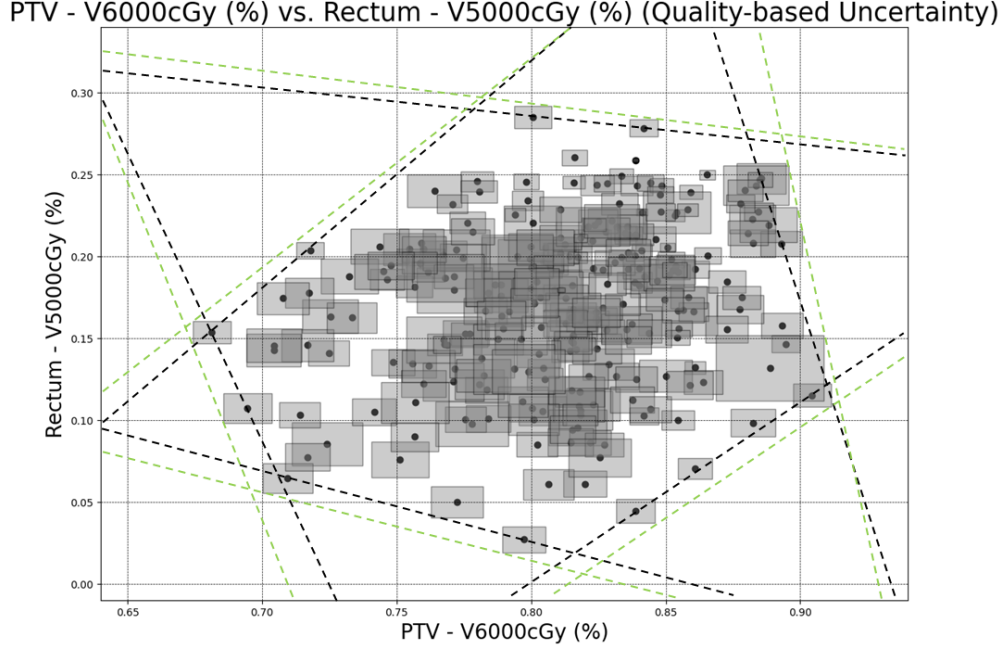


Figure 5.4: Inferred constraints without uncertainty (black) and with quality-based uncertainty size (green)

5.4 Summary

In Chapter 5, we introduced and discussed the proposed quality-based **RIO** formulation. We examined two loss functions, the fairness measure and compactness measure loss functions, which are utilized to achieve different desired properties of the inferred feasible region. The general **RIO** formulation was built from the **e-MIO** formulation detailed in the previous chapter. Due to the infinite number of constraints in the **RIO** formulation, an equivalent formulation with a finite number of constraints, known as the robust counterpart, was presented and labeled as the **e-RIO-RC** formulation. We also presented an alternative approach for solving the **RIO** formulation by incorporating all the corner points of the polyhedral uncertainty set to the set of observations using the **e-MIO** for-

mulation. It was noted that the corner point implementation would be used throughout the remainder of the study.

So far, we have established a robust foundation for implementing **IO** techniques specifically designed for **RT** treatment plans. The methodologies discussed up to this point have laid the groundwork for developing robust **IO** models that can effectively handle data uncertainty. In the following chapters, we will expand on these concepts by applying the fairness measure and compactness measure loss functions to the **e-MIO** formulation while considering the set of observations $\mathcal{K}_2$, which includes the corner points of the polyhedral uncertainty sets.

Chapter 6

Fairness Measure Formulation

In this chapter, we present the fairness measure formulation, which leverages the fairness measure loss function specified in Section 5.1. The fairness measure loss function minimizes the deviation of the total distance for all observations and aims to have all observations roughly the same total distance from all constraints. In Section 6.1, we present the model and discuss the constraints in detail. In Section 6.2, we present the results of the fairness measure formulation and comment on the quality of constraints inferred. Finally, in Section 6.3, we discuss the advantages, disadvantages, and overall performance of the fairness measure formulation.

6.1 Formulation

As discussed in Section 5.2.1, one approach to incorporating robustness into an IO problem is to consider the corner points of the box uncertainty sets as observations. The set $\mathcal{K}_2$ consists of the corner points of all the box uncertainty sets. The size of the box uncertainty

set for each observation is based on the quality of the plan from which the observation feature value originates, as explained in Section 5.3. Ensuring all the observations in the $\mathcal{K}_2$ set are feasible observations to the inferred constraints makes the below formulation a quality-based **RIO** formulation.

Similar to the fairness measure description in the previous chapter, a new decision variable, d_{ik} , will be introduced. Let d_{ik} be defined as the distance between observation $\mathbf{x}^k$, for $k \in \mathcal{K}_2$, and the hyperplane of the i^{th} constraint, for $i \in \mathcal{I}_1$. Various distance metrics can be considered, ranging in complexity, including Euclidean distance and slack distance. In this study, the distance metric used for the d_{ik} decision variable will be the linear slack distance metric. Similarly to Ghobadi and Mahmoudzadeh (2021), the slack distance is defined as $d_{ik} = \sum_{j \in \mathcal{J}} a_{ij}x_j^k - b_i$, for all $i \in \mathcal{I}_1$ and $k \in \mathcal{K}_2$. Let the matrix $\mathbf{D}$ represent all the d_{ik} values in matrix form. An additional decision variable, d_k , is introduced where $d_k = \sum_{i \in \mathcal{I}_1} d_{ik}$, for all $k \in \mathcal{K}_2$. The decision variable d_k is the total distance for observation $k \in \mathcal{K}_2$. Let $\mathbf{d}$ be the vector of all d_k decision variables and $|\mathcal{K}_2|$ be the total number of observations. The fairness measure formulation is labeled as the **F-e-RIO** formulation and is defined as follows:

$$\mathbf{F}\text{-e-RIO} : \quad \underset{\mathbf{A}, \mathbf{D}, \mathbf{b}, \mathbf{d}}{\text{minimize}} \quad \sum_{k \in \mathcal{K}_2} (d_k - \sum_{k \in \mathcal{K}_2} d_k / |\mathcal{K}_2|) \quad (6.1a)$$

$$\text{subject to} \quad \sum_{j \in \mathcal{J}} a_{ij} x_j^k \geq b_i, \quad \forall i \in \mathcal{I}_1, k \in \mathcal{K}_2, \quad (6.1b)$$

$$d_{ik} = \sum_{j \in \mathcal{J}} a_{ij} x_j^k - b_i, \quad \forall i \in \mathcal{I}_1, k \in \mathcal{K}_2, \quad (6.1c)$$

$$d_k = \sum_{i \in \mathcal{I}_1} d_{ik}, \quad \forall k \in \mathcal{K}_2, \quad (6.1d)$$

$$\|\mathbf{a}'_i\| = 1, \quad \forall i \in \mathcal{I}_1, \quad (6.1e)$$

$$d_{ik} \geq 0, \quad \forall i \in \mathcal{I}_1, k \in \mathcal{K}_2, \quad (6.1f)$$

$$d_k \geq 0, \quad \forall k \in \mathcal{K}_2, \quad (6.1g)$$

$$b_i \in \mathbb{R}, \quad \forall i \in \mathcal{I}_1, \quad (6.1h)$$

$$a_{ij} \in \mathbb{R}, \quad \forall i \in \mathcal{I}_1, j \in \mathcal{J}. \quad (6.1i)$$

The objective function, (6.1a), minimizes the deviation of the total distance for all observations. Constraint (6.1c) defines the distance decision variable, d_{ik} , as the slack distance between observation $\mathbf{x}^k$, for $k \in \mathcal{K}_2$, and the hyperplane of the i^{th} constraint, for $i \in \mathcal{I}_1$. Constraint (6.1d) defines the total distance decision variable, d_k , as the sum of all d_{ik} decision variables over $k \in \mathcal{K}_2$. Constraints (6.1f) and (6.1g) are non-negativity constraints for d_{ik} and d_k respectively. The remaining constraints have been described in previous sections.

An interesting observation about the d_{ik} decision variable is that it can be specified to be non-negative (as well as the d_k decision variable) as constraint (6.1b) ensures that the slack distance between inferred constraint $i \in \mathcal{I}_1$ and observation $k \in \mathcal{K}_2$ is positive. By making the d_{ik} variables non-negative, constraint (6.1b) is then redundant and, therefore,

can be removed. For the purpose of clarity and interpretability of the above formulation, constraint (6.1b) will not be removed.

Mathematically, the slack distance d_{ik} represents the difference between the LHS and RHS of constraint (6.1b). Clinically, the slack distance represents the gap for an inferred guideline with respect to an RT treatment plan feature or combination of two plan features (in the case of a trade-off guideline). The fairness measure loss function minimizes the overall gap for all patients, aiming to ensure that all patients have an equal total gap from all inferred guidelines. However, this loss function may cause bias by forcing the guidelines to have a smaller gap for patients with similar RT treatment plan feature values or, in some cases, by making the inferred guideline less strict (resulting in no gap for only a single patient). From a mathematical perspective, the fairness measure loss function is fair with respect to minimizing the slack distance equally for all observations. However, from a clinical perspective, it may not be considered fair. These potential drawbacks are discussed in greater detail later in this chapter.

Three potential norm evaluations were considered for the normalization constraint (6.1e). The $\mathbf{L}_2$ norm ($\sqrt{\sum_{j \in \mathcal{J}} a_{ij}^2} = 1$ or equivalently $\sum_{j \in \mathcal{J}} a_{ij}^2 = 1$), the $\mathbf{L}_1$ norm ($\sum_{j \in \mathcal{J}} |a_{ij}| = 1$) or as seen in Ghobadi and Mahmoudzadeh (2021), a proxy for the $\mathbf{L}_1$ norm ($|\sum_{j \in \mathcal{J}} a_{ij}| = 1$). The proxy may be used to reduce the number of auxiliary binary variables to only $2n$ (as opposed to $2n(m_1 + m_2)$) when reformulating formulation **F-e-RIO** as a linear mixed-integer model. A negative aspect of this proxy is that it is impossible to infer a two-dimensional constraint such as $0.5x_1 - 0.5x_2 \geq b_i$ because $|0.5 - 0.5| = 0$. It was observed that by using this proxy and attempting to infer a constraint of this shape, the model would infer a constraint that was very close to $0.5x_1 - 0.5x_2 \geq b_i$; however, it would not infer this exact constraint. As discussed in Section 6.2 of this chapter, the **F-e-RIO** formulation was solved using Python and Gurobi Optimizer software. By set-

ting the NonConvex parameter of the Gurobi solver to a value of 2.0, Gurobi can solve non-convex quadratic constraints by translating them into a bilinear form and applying spatial branching (Gurobi, 2024). Due to Gurobi's ability to recognize and solve quadratic constraints, the $\mathbf{L}_2$ norm was used for constraint (6.1e). The normalization constraint, (6.1e), can be re-written as $\sum_{j \in \mathcal{J}} a_{ij}^2 = 1$, for all $i \in \mathcal{I}_1$.

An intriguing benefit of using the $\mathbf{L}_2$ norm for constraint (6.1e) is that the d_{ik} decision variable simultaneously represents both the slack distance and the Euclidean distance. Assume that ϵ_{ik} is used to denote the Euclidean distance between observation $k \in \mathcal{K}_2$ and the hyperplane of the i^{th} inferred constraint, for $i \in \mathcal{I}_1$. Then, the constraint for defining the ϵ_{ik} decision variable would be as follows:

$$\epsilon_{ik} = \frac{|\sum_{j \in \mathcal{J}} a_{ij} x_j^k - b_i|}{\|\mathbf{a}'_i\|_2}, \quad \forall i \in \mathcal{I}_1, k \in \mathcal{K}_2. \quad (6.2)$$

Note that if constraint (6.1e) uses the $\mathbf{L}_2$ norm, then it specifies that the $\mathbf{L}_2$ norm of each row $\mathbf{a}_i$ is equal to one ($\|\mathbf{a}'_i\|_2 = 1$). This means the calculation for the Euclidean distance can be reformulated to the following constraint:

$$\epsilon_{ik} = |\sum_{j \in \mathcal{J}} a_{ij} x_j^k - b_i|, \quad \forall i \in \mathcal{I}_1, k \in \mathcal{K}_2. \quad (6.3)$$

Moreover, constraint (6.1b) forces the $\sum_{j \in \mathcal{J}} a_{ij} x_j^k$ term to be strictly larger or equal to the b_i variable for all $i \in \mathcal{I}_1$ and all $k \in \mathcal{K}_2$. Therefore, the constraint for the Euclidean distance can be simplified further to be:

$$\epsilon_{ik} = \sum_{j \in \mathcal{J}} a_{ij} x_j^k - b_i, \quad \forall i \in \mathcal{I}_1, k \in \mathcal{K}_2. \quad (6.4)$$

This constraint directly defines the value of the d_{ik} decision variable, and therefore, the d_{ik} decision variable is both the slack distance and the Euclidean distance.

6.2 Results

The data used for the experiments conducted in this chapter was described in Chapter 3. The **F-e-RIO** formulation was solved using Python and Gurobi Optimizer software. The computer used to execute these experiments had a 12th Gen Intel Core i7-12700 2.10 GHz processor with 16.0 GB of RAM. The NonConvex parameter of the Gurobi solver was set to a value of 2.0 to solve the non-convex quadratic $\mathbf{L}_2$ normalization constraint (6.1e).

To test the performance of the fairness measure formulation, eight constraints were inferred for 71 different two-dimensional problem sets (the 71 feature pairs discussed in Section 4.3). The number of unknown constraints to infer was specified to be eight because that is the maximum number of unique constraints required to cover the problem-specific corner of each of the 71 feature pairs. This information is later discussed in Chapter 8. The outlier detection and additional corner restricting constraints discussed in Sections 4.4 and 4.5, respectively, were also applied to each of the 71 two-dimensional problem sets. Table 6.1 shows the performance of the **F-e-RIO** formulation, and Figure 6.1 displays an example of the solution for a single pair of features provided by the **F-e-RIO** formulation.

Table 6.1: Computational performance of the **F-e-RIO** solution approach

(a) Time results		(b) Constraint results	
Statistic	Results (s)	Statistic	Results (constraints)
Cum.	163.46	Cum.	568
Avg.	2.30	Avg.	8
Std. Dev.	1.22	Std. Dev.	0
Max.	4.48	Max.	8
Min.	0.17	Min.	8

Table 6.1a displays the computational times for solving the **F-e-RIO** formulation. The total solution time for all 71 pairs of features was 163.46 seconds. The average time to infer eight constraints for each of the 71 different two-dimensional problem sets was 2.30 seconds. The standard deviation of the solution times was 1.22, with a maximum time of 4.48 seconds and a minimum of 0.17 seconds. Table 6.1b presents the information for the constraints inferred. This table shows that exactly eight constraints were inferred for each of the 71 problem sets.

Although the **F-e-RIO** formulation has a fast computational time, it has some drawbacks. As illustrated in Figure 6.1, which presents an example of the inferred constraints for the PTV - V 6000 cGy vs. Rectum - V 5000 cGy feature pair, only two inferred constraints are visible. This is because, out of the eight inferred constraints solved, four were the vertical constraints, and the remaining four were the horizontal constraints. This issue of inferring identical constraints is not limited to this feature pair; out of the 568 constraints inferred, only 145 were unique.

An interesting note about Figure 6.1, is that the best quality observation, highlighted with a blue star, can be seen roughly at $\text{PTV - V 6000 cGy (\%)} = 0.88$ and $\text{Rectum - V 5000 cGy (\%)} = 0.13$ as this observation has the largest uncertainty set. It demonstrates how the maximum size of uncertainty (the values presented in Table 3.3) relates to the quality of the plan from which the data is from (the maximum uncertainty for both features was specified to be $\pm 2.00\%$). Note also that the best quality observation is not on the boundary of the feasible region. This is because the quality metric used in this study, specified in Section 3.2, is not a function of the two axes in Figure 6.1.

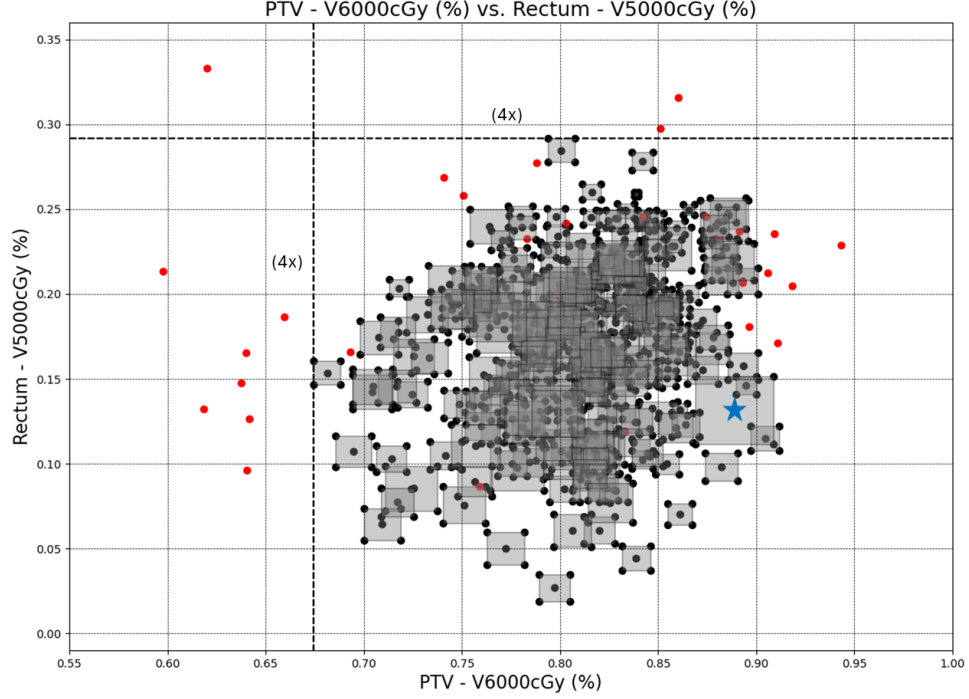


Figure 6.1: An example of the **F-e-RIO** formulation result on a single pair of features with the highest quality observation highlighted with a blue star

In addition to the repetitive constraints, some inferred constraints have unusual angles, such as the one shown in Figure 6.2. In this instance, the inferred constraint is angled and passes through only a single observation rather than two as expected, which would typically create a tighter feasible region. This occurrence may be due to the loss function used in the **F-e-RIO** formulation. The fairness measure loss function attempts to minimize the deviation of the total distance for all observations and aims to have all observations roughly the same total distance from all constraints. As a result, it can cause an inferred constraint to pass through just one observation, depending on the shape of the data, as illustrated in Figure 6.2.

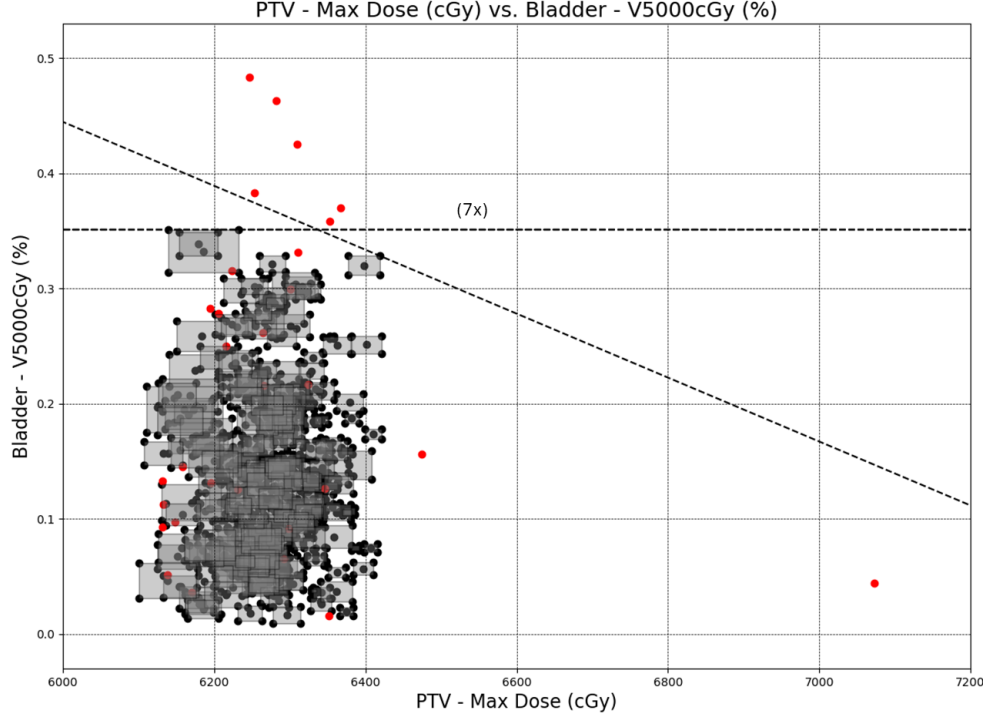


Figure 6.2: An example of an odd inferred constraint produced by the **F-e-RIO** formulation

6.3 Discussion

Overall, the **F-e-RIO** formulation benefits from a fast computational time due to its linearity; however, the quality of the inferred constraints is low. Many constraints are repetitively inferred, and some have unusual angles due to the behavior of the loss function.

One potential solution to increase the number of unique constraints inferred is to raise the specified number of constraints to impute. However, this approach does not guarantee that more unique constraints will be identified, does not prevent the formulation from inferring unusually angled constraints, and would also increase the computation time.

Another potential solution is to iteratively solve and increase the number of constraints

to infer until a specific criterion is met (e.g., a certain number of unique constraints is achieved or a time limit is reached). However, this method also does not guarantee that a certain number of unique constraints will be identified. Additionally, the unique constraints that are found may still exhibit unusual angles and may not be as tight as expected.

In the next chapter, we utilize the compactness measure loss function, which allows us to mitigate the disadvantages of the **F-e-RIO** formulation at the price of having a more complex formulation that is not as fast to solve.

Chapter 7

Compactness Measure Formulation

In this chapter, we present the compactness measure formulation, which leverages the compactness measure loss function specified in Section 5.1. The compactness measure minimizes the minimum distance of each observation from all constraints. In Section 7.1, we present the model and discuss the constraints in detail. In Section 7.2, we present the results of the compactness measure formulation and comment on the quality of constraints inferred. Finally, in Section 7.3, we discuss the advantages, disadvantages, and overall performance of the compactness measure formulation.

7.1 Formulation

The decision variable m_k is defined as the minimum distance for observation $k \in \mathcal{K}_2$ to all the inferred constraints. Let $\mathbf{m}$ be the vector of all the m_k decision variables. A binary decision variable γ_{ik} is introduced to aid in the logic of defining the m_k decision variable in the model below. Let $\mathbf{\Gamma}$ be the matrix of all γ_{ik} decision variables. The compactness

measure formulation, denoted as **C-e-RIO**, is as follows:

$$\text{C-e-RIO : } \underset{\mathbf{A}, \mathbf{D}, \mathbf{\Gamma}, \mathbf{b}, \mathbf{m}}{\text{minimize}} \quad \sum_{k \in \mathcal{K}_2} m_k \quad (7.1a)$$

$$\text{subject to} \quad \sum_{j \in \mathcal{J}} a_{ij} x_j^k \geq b_i, \quad \forall i \in \mathcal{I}_1, k \in \mathcal{K}_2, \quad (7.1b)$$

$$d_{ik} = \sum_{j \in \mathcal{J}} a_{ij} x_j^k - b_i, \quad \forall i \in \mathcal{I}_1, k \in \mathcal{K}_2, \quad (7.1c)$$

$$m_k \geq d_{ik} - M\gamma_{ik}, \quad \forall i \in \mathcal{I}_1, k \in \mathcal{K}_2, \quad (7.1d)$$

$$\sum_{i \in \mathcal{I}_1} \gamma_{ik} = |\mathcal{I}_1| - 1, \quad \forall k \in \mathcal{K}_2, \quad (7.1e)$$

$$\|\mathbf{a}'_i\| = 1, \quad \forall i \in \mathcal{I}_1, \quad (7.1f)$$

$$\gamma_{ik} \in \{0, 1\}, \quad \forall i \in \mathcal{I}_1, k \in \mathcal{K}_2, \quad (7.1g)$$

$$m_k \geq 0, \quad \forall k \in \mathcal{K}_2, \quad (7.1h)$$

$$d_{ik} \geq 0, \quad \forall i \in \mathcal{I}_1, k \in \mathcal{K}_2, \quad (7.1i)$$

$$b_i \in \mathbb{R}, \quad \forall i \in \mathcal{I}_1, \quad (7.1j)$$

$$a_{ij} \in \mathbb{R}, \quad \forall i \in \mathcal{I}_1, j \in \mathcal{J}. \quad (7.1k)$$

The objective function, (7.1a), minimizes the minimum distance for each observation. Constraint (7.1e) ensures that all γ_{ik} decision variables are set to one except for a single γ_{ik} decision variable, which is set to zero, for each observation $k \in \mathcal{K}_2$. Constraint (7.1d) in conjunction with constraint (7.1e), defines m_k as the minimum distance for observation $k \in \mathcal{K}_2$. Constraints (7.1g) and (7.1h) are domain constraints for the decision variables γ_{ik} and m_k , respectively. The remaining constraints have been described in previous sections. Due to the binary variable, γ_{ik} , being introduced, the **C-e-RIO** formulation is an **MIP** problem, which is more difficult to solve when compared to the **F-e-RIO** formulation that has no binary decision variables. This added complexity is evident in Section 7.2.

Recall that, mathematically, the slack distance d_{ik} represents the difference between the **LHS** and **RHS** for constraint (7.1b) and that clinically, the slack distance represents the gap for an inferred guideline with respect to an **RT** treatment plan feature or combination of two features (in the case of a trade-off guideline). The compactness measure minimizes the minimum gap of each patient from all inferred guidelines. This loss function ensures that as many patients as possible have no gap with respect to the inferred guidelines. These results are discussed in greater detail later in this chapter.

7.2 Results

Similar to the previous chapter, the data used for the experiments conducted in this chapter was described in Chapter 3. The **C-e-RIO** formulation was solved using Python and Gurobi Optimizer software. The computer used to execute these experiments had a 12th Gen Intel Core i7-12700 2.10 GHz processor with 16.0 GB of RAM. Similar to the previous chapters' experiments, the NonConvex parameter of the Gurobi solver was set to a value of 2.0, and the L_2 norm was used for constraint (7.1f). The same 71 different two-dimensional problem sets (71 feature pairs discussed in Section 4.3) were used to test the performance of the **C-e-RIO** formulation. The difference in these experiments was the number of unknown constraints to infer, which was set to one and two. The results of **C-e-RIO** formulation performance with the number of constraints to infer being set to one are shown in Table 7.1.

Table 7.1a displays the computational time results of the **C-e-RIO** formulation while inferring a single constraint. The total solution time for all 71 pairs of features was 2.18 seconds, which is fast. The average time to infer a single constraint for each of the 71 different two-dimensional problem sets was 0.031 seconds. The standard deviation of the

Table 7.1: Computational performance of the **C-e-RIO** solution approach (inferring one constraint)

(a) Time results		(b) Constraint results	
Statistic	Results (s)	Statistic	Results (constraints)
Cum.	2.18	Cum.	71
Avg.	0.031	Avg.	1
Std. Dev.	0.013	Std. Dev.	0
Max.	0.058	Max.	1
Min.	0.0097	Min.	1

solution times was 0.013, with a maximum time of 0.058 seconds and a minimum of 0.0097 seconds. Table 7.1b presents the information for the constraints inferred. This table shows that exactly one constraint was inferred for each of the 71 problem sets. Figure 7.1 presents an example of the **C-e-RIO** formulation applied to the feature pair PTV - V 6000 cGy and Bladder - V 5000 cGy when inferring for a single constraint.

Given the nature of the compactness measure loss function, the constraints inferred using the **C-e-RIO** formulation always pass through at least two observations. This behavior is expected as the compactness measure aims to minimize the minimum distance of each observation from all constraints. Unlike the **F-e-RIO** formulation, the **C-e-RIO** formulation avoids creating unusually angled constraints and ensures that the inferred constraints are as tight as possible.

When attempting to infer two constraints for each of the 71 problem sets using the **C-e-RIO** formulation, a time limit of 3600 seconds was set for each feature pair. Unfortunately, the model scales so poorly that only a single feature pair could be solved in less

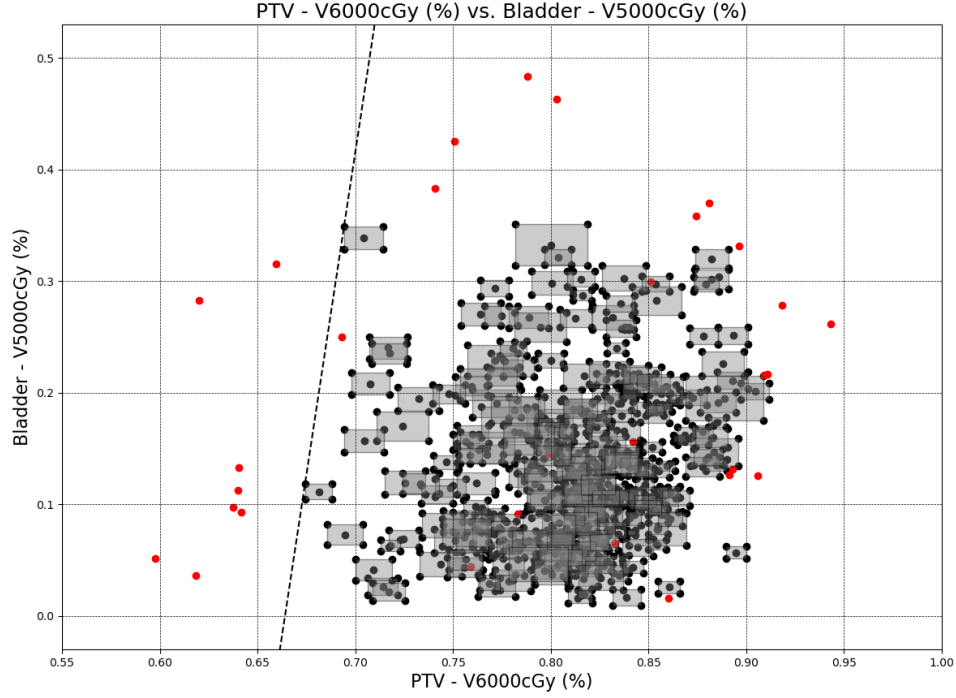


Figure 7.1: An example of the **C-e-RIO** formulation result when inferring a single constraint

than 3600 seconds. The only feature pair that could be solved in less than 3600 seconds was the PTV - D99% (%) vs. Rectum - V 5000 cGy (%) feature pair. The constraints solved for this pair of features can be observed in Figure 7.2. When the **C-e-RIO** formulation was applied to this pair of features and specified to infer two constraints, the computational time was 244.0 seconds. One explanation for this feature pair requiring less time to solve compared to the other feature pairs may be due to the shape of the data. This was the only difference compared to the other feature pairs, as the same formulation and number of observations were used for all other problems. An interesting direction for future research may be to experiment with different shapes of data in order to see how it affects the computational time for the **C-e-RIO** formulation.

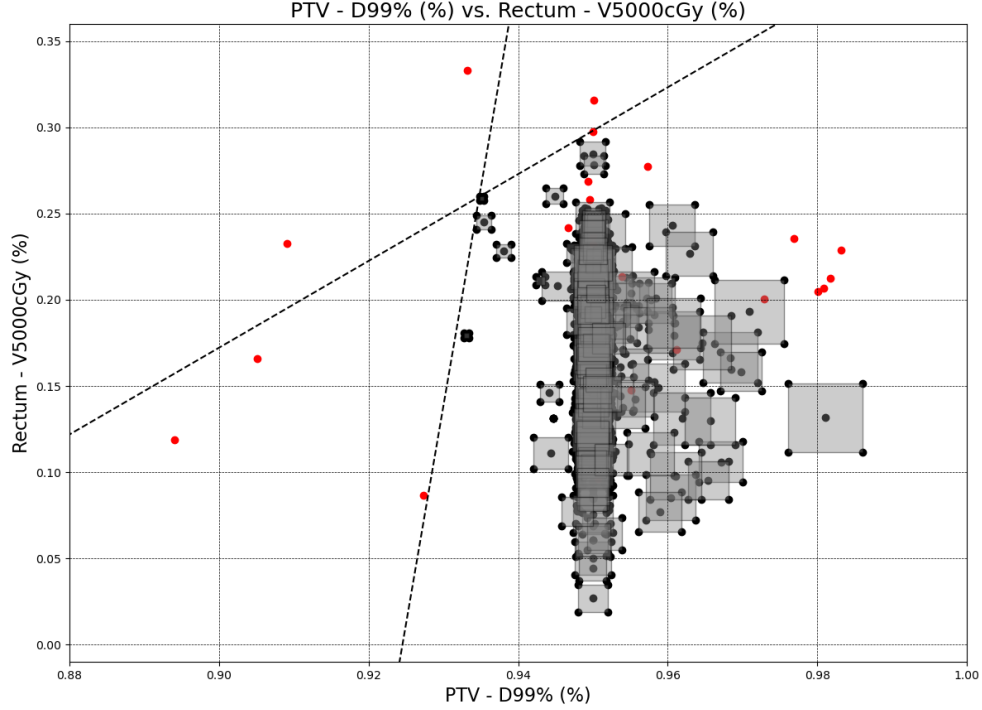


Figure 7.2: An example of the **C-e-RIO** formulation result when inferring two constraints

7.3 Discussion

The **C-e-RIO** formulation offers two notable advantages. First, the constraints inferred always pass through at least two observations, preventing the creation of unusually angled constraints, as seen with the **F-e-RIO** formulation. This ensures that the inferred constraints are as close as possible to the observations. Second, the computational time is a significant benefit. As shown in Table 7.1, inferring a single constraint using the **C-e-RIO** formulation is remarkably fast. However, this formulation struggles with inferring two or more constraints; when set to infer two constraints, nearly every feature pair requires over 3600 seconds to solve. This indicates that the compactness measure formulation does not scale well and is unsuitable for inferring two or more constraints. Given the application

needs, inferring just one constraint per feature pair is insufficient.

In the next chapter, we present an iterative single-constraint inference algorithm that takes advantage of the computational speed of the single constraint inferring **C-e-RIO** formulation. We iteratively solve for new constraints using a simplified and slightly altered **C-e-RIO** formulation in a branch-and-bound type algorithm. The performance of the iterative algorithm is then compared against the **F-e-RIO** and **C-e-RIO** formulations.

Chapter 8

Proposed Iterative Single-constraint Inference Algorithm

In this chapter, we propose an iterative single-constraint inference algorithm that leverages the single constraint inferring **C-e-RIO** formulation. This iterative algorithm infers all relevant constraints for each pair of features. The results show that the algorithm can infer multiple constraints more efficiently than the **C-e-RIO** formulation. In Section 8.1, we develop the simplified and slightly altered **C-e-RIO** formulation used in the iterative algorithm and then present the proposed algorithm. In Section 8.2, we present the results of the proposed algorithm and comment on the quality of constraints inferred. Finally, in Section 8.3, we discuss the advantages, disadvantages, and overall performance of the proposed algorithm.

8.1 Formulation

The proposed iterative single-constraint inference algorithm leverages the fact that the **C-e-RIO** formulation can infer a single constraint very quickly while also ensuring the inferred constraint binds at least two observations, inferring tight constraints. The description of the algorithm will begin with formulation (7.1). If we are only considering inferring a single constraint, we can re-write the formulation as follows:

$$\underset{\mathbf{a}, \mathbf{d}, \mathbf{m}, b}{\text{minimize}} \quad \sum_{k \in \mathcal{K}_2} m_k \quad (8.1a)$$

$$\text{subject to} \quad \sum_{j \in \mathcal{J}} a_j x_j^k \geq b, \quad \forall k \in \mathcal{K}_2, \quad (8.1b)$$

$$d_k = \sum_{j \in \mathcal{J}} a_j x_j^k - b, \quad \forall k \in \mathcal{K}_2, \quad (8.1c)$$

$$m_k \geq d_k, \quad k \in \mathcal{K}_2, \quad (8.1d)$$

$$\|\mathbf{a}'\| = 1, \quad (8.1e)$$

$$m_k \geq 0, \quad \forall k \in \mathcal{K}_2, \quad (8.1f)$$

$$d_k \geq 0, \quad \forall k \in \mathcal{K}_2, \quad (8.1g)$$

$$b \in \mathbb{R}, \quad (8.1j)$$

$$a_j \in \mathbb{R}, \quad \forall j \in \mathcal{J}. \quad (8.1h)$$

Formulation (8.1) is a simplified version of the **C-e-RIO** formulation. The i index and $\mathcal{I}_1$ set are dropped as only a single constraint is being considered. Constraint (7.1e) from the **C-e-RIO** formulation specified that the number of γ_{ik} binary decision variables that are set to one is equal to the number of constraints inferred minus one for each $k \in \mathcal{K}_2$. When inferring for a single constraint, this means all γ_{ik} decision variables are

zero, and so they can be removed from the formulation. This results in constraints (7.1e) and (7.1g) being removed and constraint (7.1d) being adjusted accordingly. Formulation (8.1) can be simplified even further as m_k is equal to d_k for all $k \in \mathcal{K}_2$ in the optimal solution. This is because m_k is minimized and constraint (8.1d) states that $m_k \geq d_k$ for all $k \in \mathcal{K}_2$. Therefore, the m_k decision variables can be replaced by d_k . This means the objective function (8.1a) is updated to have d_k instead of m_k , and constraint (8.1d) may be removed. These changes are reflected in formulation (8.2).

$$\underset{\mathbf{a}, \mathbf{d}, b}{\text{minimize}} \quad \sum_{k \in \mathcal{K}_2} d_k \quad (8.2a)$$

$$\text{subject to} \quad \sum_{j \in \mathcal{J}} a_j x_j^k \geq b, \quad \forall k \in \mathcal{K}_2, \quad (8.2b)$$

$$d_k = \sum_{j \in \mathcal{J}} a_j x_j^k - b, \quad \forall k \in \mathcal{K}_2, \quad (8.2c)$$

$$\|\mathbf{a}'\| = 1, \quad (8.2d)$$

$$d_k \geq 0, \quad \forall k \in \mathcal{K}_2, \quad (8.2e)$$

$$b \in \mathbb{R}, \quad (8.2f)$$

$$a_j \in \mathbb{R}, \quad \forall j \in \mathcal{J}. \quad (8.2g)$$

Now that formulation (8.2) is simplified further, a new binary decision variable, n_k , can be added to signal if the inferred constraint passes through an observation. If n_k is equal to zero, this means the inferred constraint passes through observation $k \in \mathcal{K}_2$. If n_k is equal to one, this means the inferred constraint does not necessarily pass through observation $k \in \mathcal{K}_2$. In addition to introducing a new decision variable, a new constraint will be added to ensure that the inferred constraint passes through at least two observations. This change is reflected in constraints (8.3d), (8.3e) and (8.3g) of formulation (8.3).

$$\mathbf{C-e-RIO-sc} : \quad \underset{\mathbf{a}, \mathbf{d}, \mathbf{n}, b}{\text{minimize}} \quad \sum_{k \in \mathcal{K}_2} d_k \quad (8.3a)$$

$$\text{subject to} \quad \sum_{j \in \mathcal{J}} a_j x_j^k \geq b, \quad \forall k \in \mathcal{K}_2, \quad (8.3b)$$

$$d_k = \sum_{j \in \mathcal{J}} a_j x_j^k - b, \quad \forall k \in \mathcal{K}_2, \quad (8.3c)$$

$$d_k \leq M n_k, \quad \forall k \in \mathcal{K}_2, \quad (8.3d)$$

$$\sum_{k \in \mathcal{K}_2} n_k = |\mathcal{K}_2| - 2, \quad (8.3e)$$

$$\|\mathbf{a}'\| = 1, \quad (8.3f)$$

$$n_k \in \{0, 1\}, \quad \forall k \in \mathcal{K}_2, \quad (8.3g)$$

$$d_k \geq 0, \quad \forall k \in \mathcal{K}_2, \quad (8.3h)$$

$$b \in \mathbb{R}, \quad (8.3i)$$

$$a_j \in \mathbb{R}, \quad \forall j \in \mathcal{J}. \quad (8.3j)$$

Formulation (8.3), denoted as **C-e-RIO-sc**, is the final formulation and will be used in the proposed iterative single-constraint inference algorithm. Constraint (8.3e) specifies that exactly two n_k binary decision variables are set to zero, while the remaining are set to one. Constraints (8.3e) and (8.3d) enforce that the distance between the inferred constraint and at least two observations is zero. The phrase “at least two observations” is used as it is possible that more than two observations may pass through the inferred constraint. This can occur because constraint (8.3d) requires d_k to be less than or equal to some large constant value, M , multiplied by n_k . Even when n_k is one, it is still possible for d_k to be zero. Lastly, constraint (8.3g) specifies the domain for the n_k binary decision variables. The remaining constraints have been described in previous sections.

Assuming a solution is found, formulation (8.3) guarantees to find a constraint that passes through at least two observations. These two observations may be identified by finding the n_k decision variables that are equal to zero. The solution may then be labeled using these two observations. Using this information, a branch-and-bound type algorithm can be leveraged to explore the boundary of the feasible region until all the constraints are inferred.

The branch-and-bound technique has been applied with some success to a variety of [Operations Research \(OR\)](#) problems ([Hillier and Lieberman, 2015](#)). The basic concept of the branch-and-bound technique is to divide and conquer. Since the original problem of inferring multiple constraints using the **C-e-RIO** formulation is too difficult to solve directly, it is divided into smaller subproblems that can be solved more easily by leveraging the fact that the **C-e-RIO** formulation can infer one constraint very quickly. The branching in the proposed iterative algorithm is done by fixing certain n_k binary decision variables while inferring a single constraint. This branching technique aims for formulation (8.3) to infer different constraints for each subproblem. The method of branching in the proposed algorithm differs from the traditional branch-and-bound technique as some added constraints from the parent to the child node are not passed down after solving the child node (assuming the child node is not fathomed). The proposed algorithm also differs from the traditional branch-and-bound technique, as a node cannot be fathomed due to bounding. In the traditional branch-and-bound approach, fathoming is partially done by bounding how good the best solution in a subset can be and then discarding the subset if its bound indicates that it cannot possibly contain an optimal solution for the original problem ([Hillier and Lieberman, 2015](#)). In the case of the proposed algorithm, a node is only fathomed if it is found to be infeasible. The pseudocode for the proposed iterative single-constraint inference algorithm is provided in [Algorithm 1](#).

Algorithm 1: Pseudocode of the iterative single-constraint inference algorithm

```
1 Initialize: let  $F_n$  be the set of forbidden observations for node  $n$ , let  $R_n$  be the set
   of required observations for node  $n$ , let  $l_n = 1$  if node  $n$  is a left child and 0
   otherwise, let  $S$  be the set of unsolved nodes,  $n = 0$ ,  $F_0 = R_0 = \emptyset$ ,  $S = \{n\}$ 
2 while  $S \neq \emptyset$  do
3   Let  $n \in S$  be a node,  $S = S/\{n\}$ ;
4   if  $n = 0$  then
5      $F_n = \emptyset$ ,  $R_n = \emptyset$ ;
6   else
7     Let  $i$  be the parent node of node  $n$ , and  $(p_1, p_2)$  be the solution from node  $i$ ;
8     if  $l_n = 1$  then
9        $F_n = F_i \cup \{p_2\}$ ,  $R_n = \{p_1\}$ ;
10    else
11       $F_n = F_i \cup \{p_1\}$ ,  $R_n = \{p_2\}$ ;
12    end
13  end
14  solve node  $n$  using formulation (8.3) while considering  $F_n$  and  $R_n$ ;
15  if an optimal solution is found then
16    record the observations the inferred constraint passes through  $(p_1, p_2)$ ,
       $n = n + 1$ , create a left child node,  $n$ ,  $l_n = 1$ ,  $S = S \cup \{n\}$ ,  $n = n + 1$ ,
      create a right child node,  $n$ ,  $l_n = 0$ ,  $S = S \cup \{n\}$ ;
17  else
18    There is no solution; fathomed this node;
19  end
20 end
```

A two-dimensional example problem will be used to aid in describing the proposed algorithm. In Figure 8.1, there are eight observations, with a specific focus on five of the eight observations (p_1 , p_2 , p_3 , p_4 , and p_6). We assume that both features considered in the example problem are to be minimized. We also assume that any outliers have been removed and the corner restricting constraints have been applied. This means that only constraints that restrict the top right of the feasible region should be inferred. The results of the proposed algorithm applied to this example problem are illustrated in the branch-and-bound type tree in Figure 8.2.

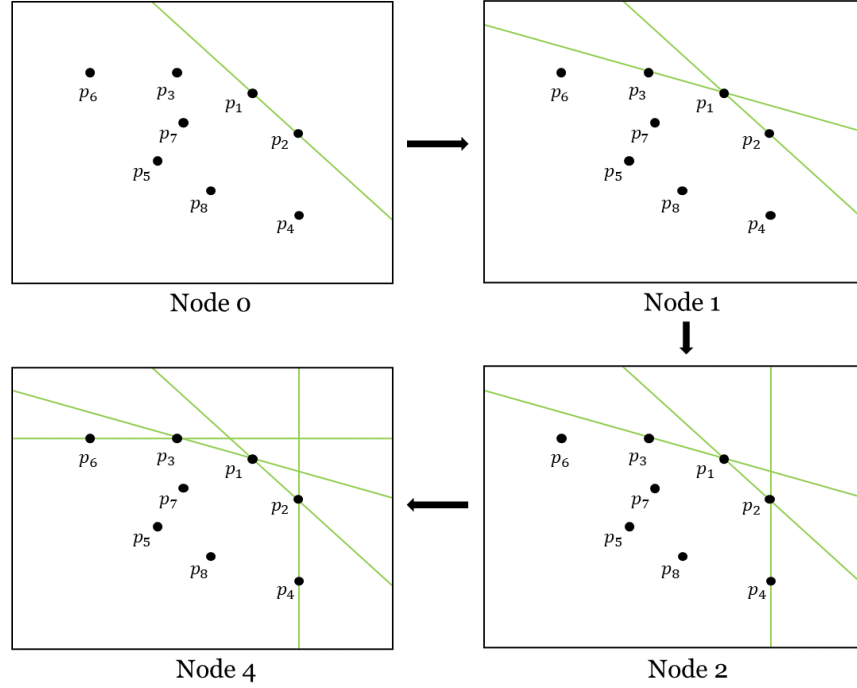


Figure 8.1: An example problem for the iterative single-constraint inference algorithm

The proposed algorithm begins by using formulation (8.3), with no added constraints, to infer a single constraint. In the example problem, the constraint inferred at node 0 passes through observations p_1 and p_2 as seen in Figures 8.1 and 8.2. These two observations

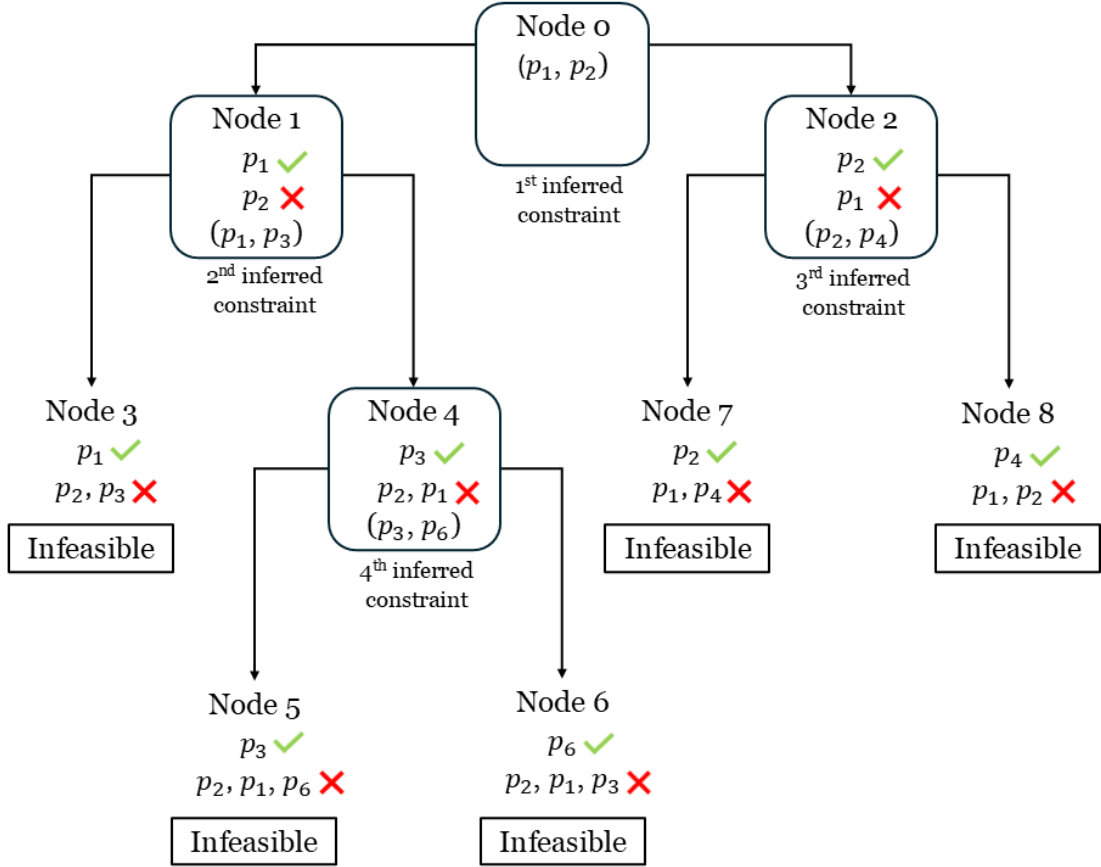


Figure 8.2: A visualization of the iterative single-constraint inference algorithm on the example problem

(identified by the n_k decision variables) and the inferred constraint are then recorded, and this solution is considered the root node. The nodes in this example will be explored using a depth-first search technique. Once the root node is solved, the problem is branched into two nodes, node 1 and node 2. As seen in Figure 8.2, node 1 is specified to infer a constraint that must pass through p_1 but not p_2 , whereas node 2 is specified to infer a constraint that must pass through p_2 but not p_1 . The n_k decision variables are fixed for these observations to enforce these additional constraints. For example, the inferred constraint at node 1

must pass through p_1 , so n_1 is set equal to zero. Conversely, the inferred constraint at node 1 must not pass through p_2 , so n_2 is set equal to one. Since formulation (8.3) requires exactly two n_k decision variables to be zero, the inferred constraint at node 1 must pass through at least one other observation that is not p_2 .

Looking at Figure 8.1, the solution for node 1 will be a constraint that passes through p_1 and p_3 . Since a solution for node 1 was found, the problem is branched into two more nodes, node 3 and node 4. A similar process that was applied to the root node occurs; however, when creating the forbidden list of observations for the children nodes, the parent nodes' forbidden list will be added as well. For instance, the inferred constraint for node 3 must pass through p_1 because it is the first observation in the solution label of its parent node, node 1. Additionally, the inferred constraint for node 3 must not pass through p_3 as it is the second observation in the solution label of its parent node. Moreover, the inferred constraint for node 3 must avoid passing through p_2 , as this observation is in the forbidden set of node 3's parent node. If the parent node of node 3 had multiple observations in its forbidden set, all of these observations would be included in the forbidden set of node 3. An example of this scenario, where multiple observations are passed down in the forbidden list, can be seen in the relationship between node 4 and node 5.

It can be seen in Figure 8.1 that it is not possible to infer a constraint that passes through p_1 and not p_2 and p_3 . Therefore, node 3 is infeasible and fathomed. A new node is then chosen to evaluate, and this process repeats until all nodes are fathomed and all the constraints possible to infer are found.

Potential issues in the proposed algorithm may arise if there are multiple observations at the same location. This could lead to the algorithm unnecessarily discovering the same inferred constraint multiple times. To mitigate this, one could process the data to ensure the set of observations for each pair of features is unique. Another issue might occur if a

constraint is inferred that passes through three or more observations. This could, again, lead to the algorithm unnecessarily discovering the same inferred constraint multiple times. To prevent this, one could check if the inferred constraint at each node has already been identified and fathom that node if it has.

8.2 Results

Similar to the previous two chapters, the data used for the experiments conducted in this chapter was described in Chapter 3. The computer used to execute these experiments had a 12th Gen Intel Core i7-12700 2.10 GHz processor with 16.0 GB of RAM. The NonConvex parameter of the Gurobi solver was, again, set to a value of 2.0, and the $\mathbf{L}_2$ norm was used for the normalization constraint (8.3f). The same 71 different two-dimensional problem sets (71 feature pairs discussed in Section 4.3) were used to test the performance of the proposed iterative algorithm. The proposed algorithm was solved using Python and Gurobi Optimizer software. While using the Gurobi Optimizer software, the integrality focus parameter was set to a value of 1.0. Setting this value to 1.0 requests that the solver works harder to try to avoid solutions that exploit integrality tolerances (Gurobi, 2024). More specifically, given the constraint $d_k \leq Mn_k$, where d_k is a non-negative continuous decision variable, n_k is a binary decision variable, and M is some large constant value, the constraint is intended to enforce the relationship that d_k must be zero if n_k is zero. With the default integer feasibility tolerance in Gurobi, the binary variable is allowed to take a value as large as 1×10^{-5} while still being considered as taking a value of zero. If the M value is large enough, the Mn_k upper bound can be substantial. Therefore, by setting the integrality focus parameter to one, this forced the logic of formulation (8.3) to hold true.

Once the root node is solved, the following nodes may be explored using techniques such

as breadth-first search, depth-first search, or at random. For these particular experiments, the nodes were explored using a depth-first search technique. Interestingly, the proposed iterative algorithm does not require specifying the number of constraints to infer; it will continue inferring constraints until no further constraints can be identified. The results of the proposed algorithm performance can be seen in Table 8.1.

Table 8.1: Computational performance of the iterative single-constraint inference algorithm

(a) Time results		(b) Constraint results	
Statistic	Results (s)	Statistic	Results (constraints)
Cum.	820.74	Cum.	325
Avg.	11.56	Avg.	4.58
Std. Dev.	12.71	Std. Dev.	1.35
Max.	58.64	Max.	8
Min.	2.61	Min.	2

The solution time for all 71 pairs of features was 820.74 seconds (13.68 minutes), which is much faster than the **C-e-RIO** formulation, considering most feature pairs took over 3600 seconds (1 hour) to infer just two constraints. The standard deviation of the solution times was 12.71, with a maximum time of 58.64 seconds and a minimum of 2.61 seconds. Table 8.1b presents the information for the constraints inferred. This table shows that a total of 325 constraints were inferred, with an average of 4.58 constraints per feature pair. The standard deviation of the constraints inferred was 1.35, with a maximum of 8 constraints and a minimum of 2 constraints.

When comparing the solution times between the proposed algorithm and the **F-e-RIO** formulation, the average solution time for the **F-e-RIO** formulation is better (2.30 vs.

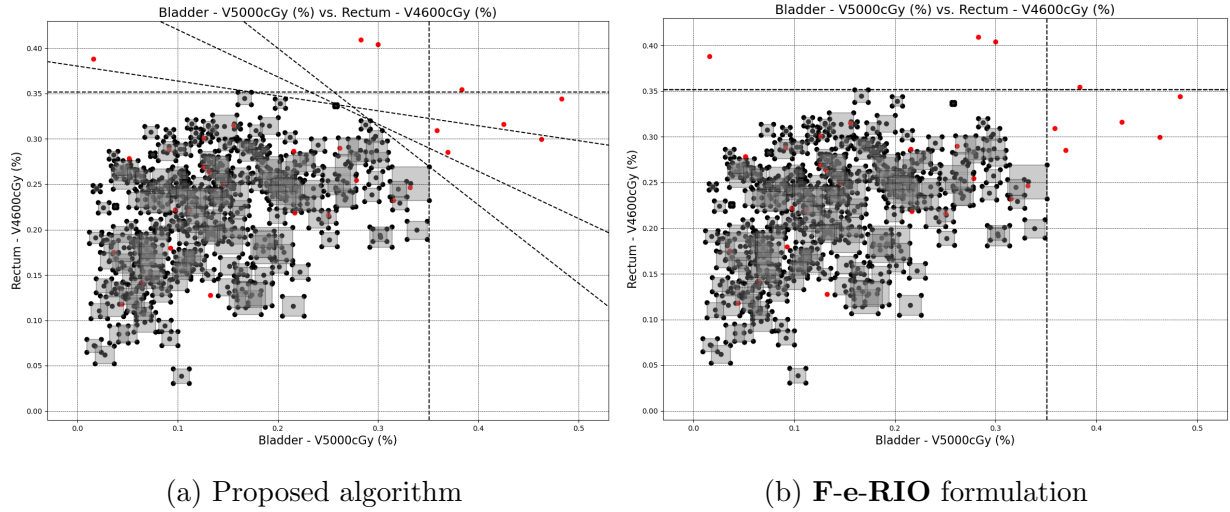


Figure 8.3: A comparison of the solutions from the proposed algorithm and the **F-e-RIO** formulation

11.56 seconds). However, in terms of the quality of the constraints inferred, the proposed algorithm outperforms the **F-e-RIO** formulation. Figure 8.3 below shows a comparison of the constraints inferred for the feature pair Bladder - V 5000 cGy (%) vs. Rectum - V 4600 cGy (%) using the proposed algorithm and the **F-e-RIO** formulation. As seen in the comparison, the quality of the solution is better in the proposed algorithm as there are no repetitive constraints inferred, and all the possible trade-off constraints to infer are found. The proposed algorithm also does not suffer from inferring loose constraints (passing through only one observation) as the **F-e-RIO** formulation did, due to the n_k constraints forcing each inferred constraint to pass through at least two observations. Another great example of the quality of the proposed algorithm includes the solution for the feature pair PTV - V 6000 cGy (%) vs. Rectum - V 4600 cGy (%) in Figure 9.1 in Chapter 9. This figure compares the inferred constraints with the current guidelines. The complete set of inferred constraints using the proposed algorithm can be seen in Appendix A.

8.3 Discussion

Regarding the quality of constraints inferred, the iterative single-constraint inference algorithm consistently delivers the best results. When compared to the **F-e-RIO** formulation, the proposed algorithm does not suffer from inferring repeated constraints. Another advantage displayed by the proposed algorithm when compared to the **F-e-RIO** formulation is that the constraints are tighter (not at unusual angles). Every inferred constraint passes through at least two observations, whereas the **F-e-RIO** formulation would sometimes infer constraints that pass through only a single observation. The proposed algorithm leverages the **C-e-RIO** formulation (with some adjustments), so the quality of the constraints inferred by both these solution approaches is comparable. The main benefit of the proposed algorithm over the **C-e-RIO** formulation is the solution time.

The average solution time of the **F-e-RIO** formulation (2.30 seconds) was approximately five times faster than that of the proposed algorithm (11.56 seconds). However, as previously mentioned, the proposed algorithm offers significant advantages in the quality of the inferred constraints compared to the **F-e-RIO** formulation. Additionally, solving for each possible inferred constraint in an average time of 11.56 seconds per feature pair remains quite reasonable for the application at hand, which is constraint inference for [RT](#) treatment plans.

The largest advantage of the proposed algorithm over the two previous formulations is that it does not require specifying the number of constraints to infer. This advantage saves time by eliminating the need to guess the number of constraints, solve for them, verify the inferred constraints, adjust the number of constraints to infer, and then resolve. Instead, the proposed algorithm automatically infers all possible constraints.

8.4 Summary

In Chapter 8, we designed the iterative single-constraint inference algorithm to leverage the computational speed of the **C-e-RIO** formulation, which can infer a single constraint very quickly. The proposed algorithm iteratively applies a simplified and slightly altered version of the **C-e-RIO** formulation to infer all relevant constraints for each pair of features, ensuring that the constraints remain tight and computational times remain manageable. The formulation simplifies the problem by using a branch-and-bound type algorithm and focusing on subproblems of single constraint inference, resulting in a more efficient and scalable solution approach. This method addresses the scalability issues identified with the **C-e-RIO** formulation when inferring multiple constraints, thus providing a practical and effective solution for inferring guidelines and trade-off constraints for RT treatment plans.

So far, we have demonstrated the development of the proposed algorithm to overcome the computational challenges associated with the **C-e-RIO** formulation. Using these ideas, we have shown how to infer constraints efficiently while maintaining high-quality solutions. The next chapters will delve into the discussion and conclusions, summarizing the findings of the entire study and highlighting potential future research directions. This will provide a comprehensive overview of how the proposed algorithm can contribute to the field of RT treatment planning and the broader implications for optimization in healthcare.

Chapter 9

Discussion

A discussion of each of the three [IO](#) approaches has been made in their respective chapters. Table [9.1](#) summarizes the advantages and disadvantages of each of the solution approaches. The average solution times for the **F-e-RIO** formulation were the fastest out of the three solution approaches; however, due to the higher quality constraints inferred and the lack of requirement of specifying the number of constraints to infer, the iterative single-constraint inference algorithm is determined to be the better solution approach. An area of interest that has not been explored is if the solution of the proposed algorithm is equivalent to the solution of **C-e-RIO** formulation. This is left as a future area of research.

As mentioned in Chapter [4](#), once the two-dimensional constraints are inferred, they can be superimposed back onto the n -dimensional space from which they came. The complete set of constraints inferred using the proposed algorithm is presented in Appendix [A](#).

Table 9.1: Performance summary of the solution approaches

Solution Approach	Advantages	Disadvantages
F-e-RIO Model	• Fast solution times for inferring 8 constraints (avg 2.30s)	• Infers repeated constraints • Infers loose constraints • Requires input for the number of constraints to infer
C-e-RIO Model	• Infers unique tight constraints • Very fast solutions times for inferring 1 constraint (avg. 0.031s)	• Scales poorly, solution times exceed 3600s for 2+ constraints • Requires input for the number of constraints to infer
Proposed Algorithm	• Infers unique tight constraints • Does not require input for the number of constraints to infer	• Solution times are slower than the F-e-RIO model (avg. 11.56s)

The results of the proposed algorithm are intended to be compared with the original guidelines used by the [GRRCC](#) to see if there is a need to update specific guidelines or if there is a possibility to consider an implicit trade-off constraint between a pair of [RT](#) plan features. Figure [9.1](#) compares the inferred constraints discovered using the proposed algorithm with the current guidelines for the PTV - V 6000 cGy (%) and Rectum - V

4600 cGy(%) RT plan features. The guidelines for the treatment plan features are stated in Chapter 3. As seen in the figure, based on the RT plans that have been accepted, the outliers having been removed, and a quality-based uncertainty having been applied, the current guideline for the Rectum - V 4600 cGy (%) feature is very close to the inferred guideline. This indicates that the GRRCC should not adjust this current dose limitation (or if they do, it should be a minor adjustment). The other feature dose goal, PTV - V 6000 cGy (%) $\geq 75.0\%$, on the other hand, appears to be too strict. Based on the proposed algorithm, the dose goal for the PTV - V 6000 cGy (%) should be relaxed to PTV - V 6000 cGy (%) $\geq 67.43\%$.

In terms of potential implicit trade-off constraints, Figure 9.1 shows that there is one very significant trade-off constraint that may be considered. This constraint is as follows:

$$0.7153[\text{PTV - V6000 cGy (\%)}] - 0.6988[\text{Rectum - V4600 cGy (\%)}] \geq 0.322 \quad (9.1)$$

It is understandable that if only one-dimensional constraints were used, and the triangular area of the feasible region created by the three inferred constraints was to be avoided, the current guidelines might be used; however, there are visible observations that violate the current guidelines, and are still considered acceptable. Introducing a second dimension to the constraints of the feasible region allows for the ability to infer constraints that more accurately represent the RT plans that are being accepted. Ideally, the goal is to provide these inferred constraints to the RT treatment planners and oncologists to aid in making “acceptable” RT treatment that would require fewer iterations.

As mentioned above, adding a second dimension to the constraints of the feasible region allows for the ability to infer constraints that more accurately represent the RT plans that are being accepted. This is true for higher-dimensional constraints as well. As stated in previous chapters, higher-dimensional constraints were not considered because the primary

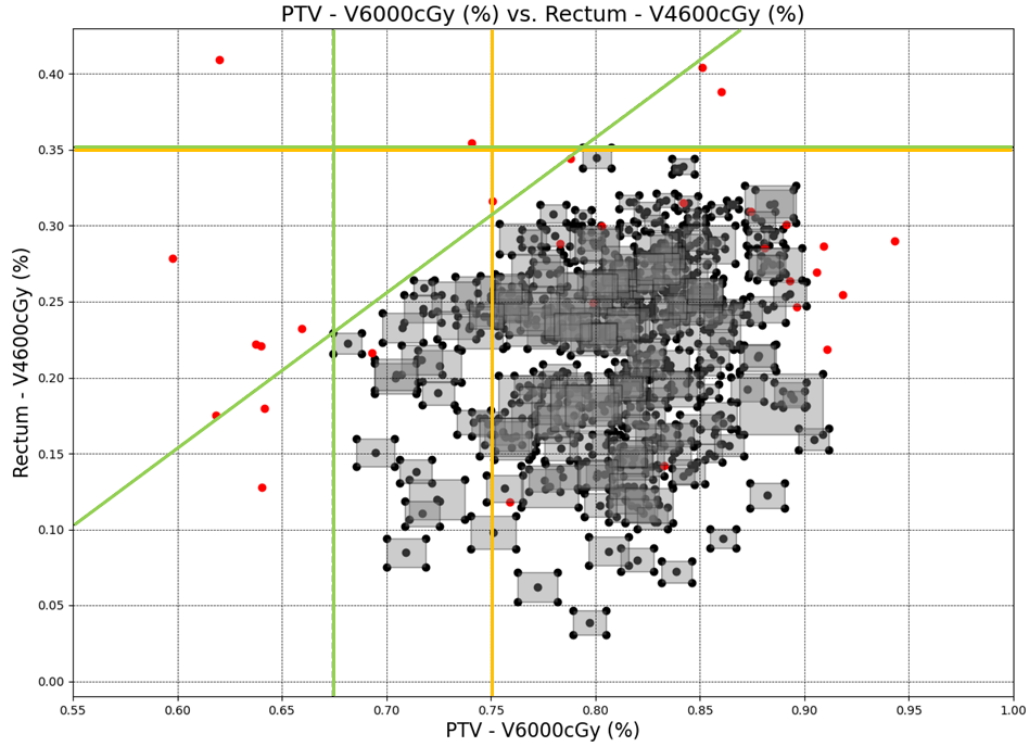


Figure 9.1: PTV - V 6000 cGy (%) vs. Rectum - V 4600 cGy (%) with inferred constraints (green) and current guidelines (orange)

users of the generated constraints are expected to be treatment planners or oncologists. If the inferred constraints were to be used in an optimization model, then higher-dimensional constraints could be considered as optimization models do not struggle to interpret constraints of three dimensions or higher. A future area of research could involve developing the proposed algorithm further to infer higher-dimension constraints for [RT](#) planning.

Another future area of research could involve finding a subset of the most useful inferred constraints. The proposed algorithm inferred 325 constraints, which are too many constraints for a treatment planner or oncologist to consider while designing treatment

plans. To determine how good each trade-off constraint is, one could find the percent area it removes from the one-dimensional feasible area for that pair of features. A subset of the best trade-off constraint may then be found using this approach.

The proposed algorithm could be particularly useful for inferring different levels of constraints in RT planning. For instance, red-level inferred constraints might cover the top 95% of RT plans, yellow-level the top 85%, and green-level the top 75%. This could be achieved by modifying the outlier detection algorithm to a top X% algorithm, as discussed in Chapter 4, and adjusting the X value. This method could greatly benefit treatment planners and oncologists by providing three sets of constraints. These constraints could be used to follow a treatment plan strategy that, for example, allows a maximum of five green-level violations and three yellow-level violations while ideally avoiding any red-level violations.

In addition to the abovementioned point, instead of considering different percentages of patient outliers, we could introduce the concept of inferring chance constraints. This would mean we are inferring for a constraint, $\sum_{j \in \mathcal{J}} a_j x_j^k \geq b$, which is true for 95% of the observations, thereby inferring constraints that are outlier-aware.

Other interesting uses for the proposed algorithm could involve separating the data by the year to see if the RT treatment planners and oncologists produced significantly different plans throughout different years. The data could also be separated into different prostate size groups to see how different prostate sizes affect the RT treatment plans.

It is worth noting that ideally, we would not need to remove outliers from the data since all observations come from a set of approved RT treatment plans for patients with various guideline requirements. This raises the idea of examining different patient subsets based on health conditions, diseases, age, etc. A more homogeneous patient group could allow

for more accurate constraint inference for [RT](#) treatment plans without needing to discard outliers. This concept also opens the door to using [Machine Learning \(ML\)](#) techniques, like clustering, to partition the set of patients. However, patient-specific information, such as the factors mentioned, was not collected because these factors for each [RT](#) plan were unavailable to us.

Another important thing to note is that using this [RIO](#) method to constantly update the [RT](#) treatment plan goals would not be wise. If the proposed algorithm were to be used periodically and the inferred constraints were implemented, the feasible region might increase with each iteration due to the uncertainty applied to the observations. It would be more appropriate to use the proposed algorithm as a tool to inform treatment planners and oncologists that a current dose goal is too loose or strict or to inform the cancer center of potential trade-off constraints that may be leveraged in the planning process.

It must also be mentioned that many parameters of the proposed algorithms can be fine-tuned or adjusted, such as the outlier detection algorithm, the uncertainty shapes and sizes, the pair of features considered, and the [RT](#) treatment plan quality function used.

Chapter 10

Conclusions

In this study, we considered the problem of inferring the feasible region for [RT](#) treatment plans designed for patients with prostate cancer. We defined a quality-based [Robust-inverse Optimization \(RIO\)](#) formulation which infers for both the [LHS](#) and [RHS](#) values of the constraint parameters for a linear [FO](#) problem while considering uncertainty in the observations. The data provided by the [GRRCC](#) was discussed and analyzed in-depth by comparing their current guidelines with the RTOG and Quantec guidelines, defining a quality metric for an [RT](#) treatment plans for prostate cancer, and discussing and presenting data uncertainty for the [RT](#) plan features considered in current guidelines used by the [GRRCC](#).

We introduced application-based considerations when developing an [IO](#) formulation to infer for [RT](#) treatment plan guidelines. The n -dimensional problem space was divided into smaller two-dimensional problem spaces to infer a maximum of two-dimensional constraints to ensure interpretability. The specific features used to impute the feasible region for [RT](#) treatment plans were determined to be a subset of the features leveraged in the current

guidelines used by the [GRRCC](#). A subset of the two-dimensional problem spaces was used to infer the constraints to avoid inferring less meaningful guidelines, such as a trade-off constraint between two features from the same [OAR](#) or target volume. An outlier detection algorithm was discussed and applied to the data to ensure outlier observations are not used to infer the feasible region. Constraint parameter restrictions to avoid inferring constraints that would exclude better [RT](#) treatment plans in the future were derived and presented.

The quality-based [RIO](#) formulation was presented by first considering the linear [FO](#) problem. We leveraged previous research, which presented a tractable reformulation of the [IO](#) problem while considering multiple observations. We proposed an addition to this formulation by considering uncertainty in the observations using robust optimization techniques. We showed that by considering the corner points of a polyhedral uncertainty set as additional observations, the [IO](#) formulation is robust to all realizations of each observation. We addressed the need for incorporating quality-based uncertainty when inferring the feasible region for [RT](#) treatment plans and introduced a method that utilized both [RT](#) plan quality and the previously discussed data uncertainty.

We evaluated the quality of constraints inferred and the computational performance of the [RIO](#) formulation by using the fairness measure loss function, which resulted in the **F-e-RIO** formulation, and the compactness measure loss function, which resulted in the **C-e-RIO** formulation. The computational time of the **F-e-RIO** formulation was fast, but the quality of the constraints inferred was poor due to inferring loose and repetitive constraints. The quality of the **C-e-RIO** formulation was better than the **F-e-RIO** formulation, and the computational time of inferring a single constraint was fast. Still, the formulation scaled poorly, as inferring two or more constraints required more than 3600 seconds on average.

We proposed an iterative single-constraint inference algorithm, which leveraged the single constraint inferring **C-e-RIO** formulation. Due to the reduced requirement of inferring a single constraint, the **C-e-RIO** formulation was able to be simplified to formulation (8.2). An additional decision variable and constraint were added to the simplified formulation to identify the observations that the inferred constraint passes through, resulting in formulation (8.3). This formulation was then used in a branch-and-bound type algorithm to infer all possible constraints for a two-dimensional problem. The proposed algorithm was shown to infer high-quality constraints within a reasonable time frame with the added benefit of not having to specify how many constraints to impute.

The constraints inferred by the proposed algorithm were compared with the existing guidelines used by the **GRRCC**. This comparison allows the **GRRCC** to evaluate whether updating their current guidelines or incorporating trade-off constraints could better align with the implicit logic of the **RT** plans accepted by oncologists. The ultimate goal is to leverage these inferred guidelines to streamline the development of **RT** treatment plans, reducing the number of iterations needed and thereby saving resources such as time and money.

It is important to note that our solution approach was trained on only a subset of the **RT** treatment plans provided by the **GRRCC**. To infer constraints or guidelines that are closer to universal acceptance, a larger dataset from multiple institutions would be needed to train and evaluate the performance of our solution approach.

As discussed throughout this thesis, several areas for future research can be explored. These include proving that the solution from the proposed algorithm is equivalent to the solution of the compactness measure formulation, identifying a subset of the most clinically relevant inferred constraints, and advancing the proposed algorithm to infer higher-dimensional constraints. A minor area for future research, discussed in Chapter 7, involves

investigating why some feature pairs are solved more quickly than others using the compactness measure formulation for inferring two constraints. Another promising direction is to integrate [ML](#) techniques into the pre-processing stage of the solution approaches discussed. For instance, using [Principle Component Analysis \(PCA\)](#) (based on dose constraint violations or other metrics) for dimensionality reduction could be an alternative to the feature selection approach presented.

References

- Farzin Ahmadi, Fardin Ganjkanloo, and Kimia Ghobadi. Inverse learning: Solving partially known models using inverse optimization, 2023. URL <https://arxiv.org/abs/2011.03038>.
- Ravindra K Ahuja and James B Orlin. Inverse optimization. *Operations Research*, 49(5):771–783, 2001.
- Temitayo Ajayi, Taewoo Lee, and Andrew J Schaefer. Objective selection for cancer treatment: An inverse optimization approach. *Operations Research*, 70(3):1717–1738, 2022.
- Anil Aswani, Zuo-Jun Shen, and Auyon Siddiq. Inverse optimization with noisy data. *Operations Research*, 66(3):870–892, 2018.
- Aaron Babier, Justin J Boutilier, Michael B Sharpe, Andrea L McNiven, and Timothy CY Chan. Inverse optimization of objective function weights for treatment planning using clinical dose-volume histograms. *Physics in Medicine & Biology*, 63(10):105004, 2018.
- Aaron Babier, Timothy CY Chan, Taewoo Lee, Rafid Mahmood, and Daria Terekhov. An ensemble learning framework for model fitting and evaluation in inverse linear optimization. *Informa Journal on Optimization*, 3(2):119–138, 2021.

- Gillian C Barnett, Catherine ML West, Alison M Dunning, Rebecca M Elliott, Charlotte E Coles, Paul DP Pharoah, and Neil G Burnet. Normal tissue reactions to radiotherapy: towards tailoring treatment dose by genotype. *Nature Reviews Cancer*, 9(2):134–142, 2009.
- Rajamanickam Baskar, Kuo Ann Lee, Richard Yeo, and Kheng-Wei Yeoh. Cancer and radiation therapy: current advances and future directions. *International Journal of Medical Sciences*, 9(3):193–199, 2012.
- Adrian C Begg, Fiona A Stewart, and Conchita Vens. Strategies to improve radiotherapy with targeted drugs. *Nature Reviews Cancer*, 11(4):239–253, 2011.
- Aharon Ben-Tal and Arkadi Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming*, 88:411–424, 2000.
- Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*, volume 28. Princeton university press, 2009.
- Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.
- Dimitris Bertsimas, Omid Nohadani, and Kwong Meng Teo. Nonconvex robust optimization for problems with constraints. *INFORMS Journal on Computing*, 22(1):44–58, 2010.
- Dimitris Bertsimas, David B Brown, and Constantine Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, 2011.
- Dimitris Bertsimas, Vishal Gupta, and Ioannis Ch Paschalidis. Data-driven estimation in equilibrium using inverse optimization. *Mathematical Programming*, 153:595–633, 2015.

- Hans-Georg Beyer and Bernhard Sendhoff. Robust optimization—a comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, 196(33-34):3190–3218, 2007.
- Merve Bodur, Timothy CY Chan, and Ian Yihang Zhu. Inverse mixed integer optimization: Polyhedral insights and trust region methods. *INFORMS Journal on Computing*, 34(3):1471–1488, 2022.
- 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.
- Justin J Boutilier, Taewoo Lee, Tim Craig, Michael B Sharpe, and Timothy CY Chan. Models for predicting objective function weights in prostate cancer imrt. *Medical Physics*, 42(4):1586–1595, 2015.
- Freddie Bray, Mathieu Laversanne, Hyuna Sung, Jacques Ferlay, Rebecca L Siegel, Isabelle Soerjomataram, and Ahmedin Jemal. Global cancer statistics 2022: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: A Cancer Journal for Clinicians*, 74(3):229–263, 2024.
- Aykut Bulut and Ted K Ralphs. On the complexity of inverse mixed integer linear optimization. *SIAM Journal on Optimization*, 31(4):3014–3043, 2021.
- Michal Černý and Milan Hladík. Inverse optimization: towards the optimal parameter set of inverse lp with interval coefficients. *Central European Journal of Operations Research*, 24:747–762, 2016.
- Timothy CY Chan and Neal Kaw. Inverse optimization for the recovery of constraint parameters. *European Journal of Operational Research*, 282(2):415–427, 2020.

- Timothy CY Chan and Taewoo Lee. Trade-off preservation in inverse multi-objective convex optimization. *European Journal of Operational Research*, 270(1):25–39, 2018.
- Timothy CY Chan and Velibor V Mišić. Adaptive and robust radiation therapy optimization for lung cancer. *European Journal of Operational Research*, 231(3):745–756, 2013.
- 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, 2014.
- 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.
- Timothy CY Chan, Rafid Mahmood, and Ian Yihang Zhu. Inverse optimization: Theory and applications. *Operations Research*, 0(0), 2023.
- Joseph YJ Chow and Will W Recker. Inverse optimization with endogenous arrival time constraints to calibrate the household activity pattern problem. *Transportation Research Part B: Methodological*, 46(3):463–479, 2012.
- Christoffer Cromvik and Michael Patriksson. On the robustness of global optima and stationary solutions to stochastic mathematical programs with equilibrium constraints, part 2: Applications. *Journal of Optimization Theory and Applications*, 144:479–500, 2010.

- Johnson Darko, Ernest Osei, Andre Fleck, and Ramana Rachakonda. Retrospective dosimetric evaluation of VMAT plans for prostate cancer treatment. *Journal of Radiotherapy in Practice*, 18(2):155–164, 2019.
- Geoff Delaney, Susannah Jacob, Carolyn Featherstone, and Michael Barton. The role of radiotherapy in cancer treatment: estimating optimal utilization from a review of evidence-based clinical guidelines. *Cancer: Interdisciplinary International Journal of the American Cancer Society*, 104(6):1129–1137, 2005.
- Stephan Dempe and Sebastian Lohse. Inverse linear programming. In *Recent Advances in Optimization*, pages 19–28. Springer, 2006.
- Laurent El Ghaoui and Hervé Lebre. Robust solutions to least-squares problems with uncertain data. *SIAM Journal on Matrix Analysis and Applications*, 18(4):1035–1064, 1997.
- Virginie Gabrel, Cécile Murat, and Aurélie Thiele. Recent advances in robust optimization: An overview. *European Journal of Operational Research*, 235(3):471–483, 2014.
- Bennet Gebken and Sebastian Peitz. Inverse multiobjective optimization: Inferring decision criteria from data. *Journal of Global Optimization*, 80(1):3–29, 2021.
- Archis Ghate. Imputing radiobiological parameters of the linear-quadratic dose-response model from a radiotherapy fractionation plan. *Physics in Medicine & Biology*, 65(22):225009, 2020a.
- Archis Ghate. Inverse optimization in semi-infinite linear programs. *Operations Research Letters*, 48(3):278–285, 2020b.

- Zahra Ghatrani and Archis Ghate. Inverse optimization in semi-definite programs to impute unknown constraint matrices. *Computers & Operations Research*, 168:106681, 2024.
- Kimia Ghobadi and Houra Mahmoudzadeh. Inferring linear feasible regions using inverse optimization. *European Journal of Operational Research*, 290(3):829–843, 2021.
- Kimia Ghobadi, Taewoo Lee, Houra Mahmoudzadeh, and Daria Terekhov. Robust inverse optimization. *Operations Research Letters*, 46(3):339–344, 2018.
- Ali Goli, Justin J Boutilier, Tim Craig, Michael B Sharpe, and Timothy CY Chan. A small number of objective function weight vectors is sufficient for automated treatment planning in prostate cancer. *Physics in Medicine & Biology*, 63(19):195004, 2018.
- Çiğdem Güler and Horst W Hamacher. Capacity inverse minimum cost flow problem. *Journal of Combinatorial Optimization*, 19(1):43–59, 2010.
- Gurobi. Gurobi optimization. <https://www.gurobi.com/>, 2024. Online; accessed 15 July 2024.
- Frederick S Hillier and Gerald J Lieberman. *Introduction to operations research*. McGraw-Hill, 2015.
- Garud Iyengar and Wanmo Kang. Inverse conic programming with applications. *Operations Research Letters*, 33(3):319–330, 2005.
- Arezou Keshavarz, Yang Wang, and Stephen Boyd. Imputing a convex objective function. In *2011 IEEE International Symposium on Intelligent Control*, pages 613–619, 2011. doi: 10.1109/ISIC.2011.6045410.

- Jourdain B Lamperski and Andrew J Schaefer. A polyhedral characterization of the inverse-feasible region of a mixed-integer program. *Operations Research Letters*, 43(6):575–578, 2015.
- Houra Mahmoudzadeh and Kimia Ghobadi. Learning acceptability criteria from good and bad decisions: An inverse optimization approach, 2024. URL <https://arxiv.org/abs/2207.02894>.
- 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.
- Lawrence B Marks, Ellen D Yorke, Andrew Jackson, Randall K Ten Haken, Louis S Constine, Avraham Eisbruch, Søren M Bentzen, Jiho Nam, and Joseph O Deasy. Use of normal tissue complication probability models in the clinic. *International Journal of Radiation Oncology* Biology* Physics*, 76(3):S10–S19, 2010.
- Mahsa Moghaddass and Daria Terekhov. Inverse integer optimization with multiple observations. *Optimization Letters*, 15(4):1061–1079, 2021.
- Peyman Mohajerin Esfahani, Soroosh Shafieezadeh-Abadeh, Grani A Hanasusanto, and Daniel Kuhn. Data-driven inverse optimization with imperfect information. *Mathematical Programming*, 167:191–234, 2018.
- Mostafa Naghavi, Ali Asghar Foroughi, and Masoud Zarepisheh. Inverse optimization for multi-objective linear programming. *Optimization Letters*, 13(2):281–294, 2019.
- Julien Roland, Yves De Smet, and José Rui Figueira. Inverse multi-objective combinatorial optimization. *Discrete Applied Mathematics*, 161(16-17):2764–2771, 2013.

- RTOG. RTOG Foundation Inc. <https://www.rtog.org/>, 2024. Online; accessed 09 July 2024.
- Javier Saez-Gallego and Juan M Morales. Short-term forecasting of price-responsive loads using inverse optimization. *IEEE Transactions on Smart Grid*, 9(5):4805–4814, 2017.
- GA Sayre and D Ruan. Automatic treatment planning with convex imputing. In *Journal of Physics: Conference Series*, volume 489, page 012058. IOP Publishing, 2014.
- Andrew J Schaefer. Inverse integer programming. *Optimization Letters*, 3:483–489, 2009.
- Zahed Shahmoradi and Taewoo Lee. Quantile inverse optimization: Improving stability in inverse linear programming. *Operations Research*, 70(4):2538–2562, 2022.
- Allen L Soyster. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21(5):1154–1157, 1973.
- Onur Tavashioğlu, Taewoo Lee, Silviya Valeva, and Andrew J Schaefer. On the structure of the inverse-feasible region of a linear program. *Operations Research Letters*, 46(1): 147–152, 2018.
- Lizhi Wang. Cutting plane algorithms for the inverse mixed integer linear programming problem. *Operations Research Letters*, 37(2):114–116, 2009.
- Jianzhong Zhang and Zhenhong Liu. Calculating some inverse linear programming problems. *Journal of Computational and Applied Mathematics*, 72(2):261–273, 1996.
- Jianzhong Zhang and Zhenhong Liu. A further study on inverse linear programming problems. *Journal of Computational and Applied Mathematics*, 106(2):345–359, 1999.

- Jianzhong Zhang and Liwei Zhang. An augmented lagrangian method for a class of inverse quadratic programming problems. *Applied Mathematics and Optimization*, 61(1):57, 2010.
- Jianzhong Zhang, Zhongfan Ma, and Chao Yang. A column generation method for inverse shortest path problems. *Zeitschrift für Operations Research*, 41:347–358, 1995.
- Jianzhong Zhang, Liwei Zhang, and Xiantao Xiao. A perturbation approach for an inverse quadratic programming problem. *Mathematical Methods of Operations Research*, 72: 379–404, 2010.
- Yi Zhang, Liwei Zhang, Jia Wu, and Jianzhong Zhang. A perturbation approach for an inverse quadratic programming problem over second-order cones. *Mathematics of Computation*, 84(291):209–236, 2015.

APPENDICES

Appendix A

The Complete Set of Constraints Inferred

The following figures display the complete set of constraints inferred using the iterative single-constraint inference algorithm from Chapter 8. The solution approach leverages accepted [RT](#) treatment plans from the [GRRCC](#), with outliers removed (the red observations), quality-based uncertainty applied, and constraint parameter restrictions applied to avoid inferring constraints which would exclude better [RT](#) treatment plans.

Note that many parameters of the proposed algorithm can be fine-tuned or adjusted, such as the outlier detection algorithm, the uncertainty shapes and sizes, the pair of features considered, and the [RT](#) treatment plan quality function used.

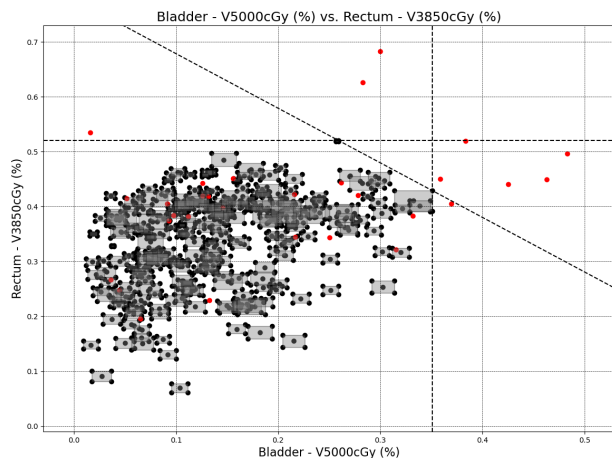


Figure A.1: Bladder - V 5000 cGy (%)
vs. Rectum - V 3850 cGy (%)

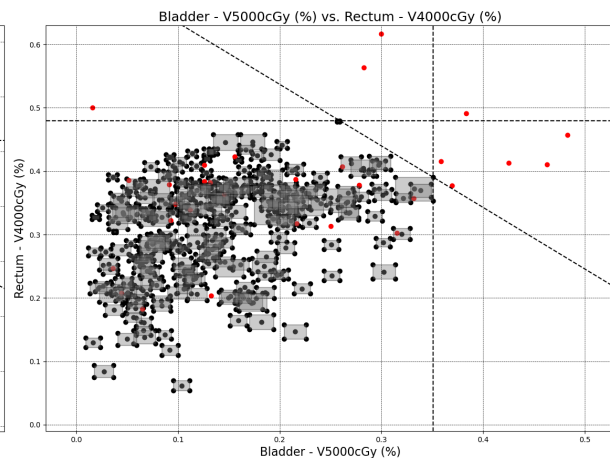


Figure A.2: Bladder - V 5000 cGy (%)
vs. Rectum - V 4000 cGy (%)

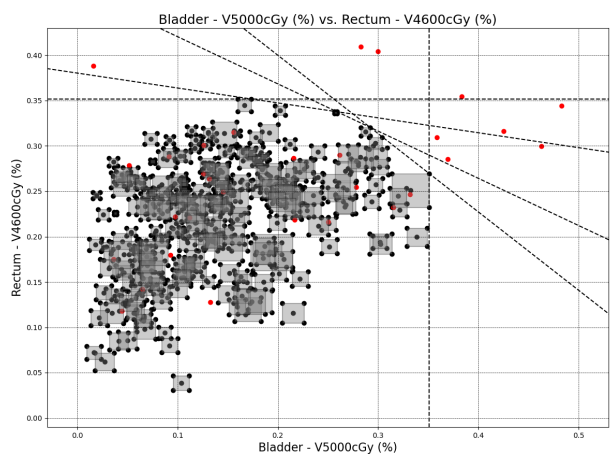


Figure A.3: Bladder - V 5000 cGy (%)
vs. Rectum - V 4600 cGy (%)

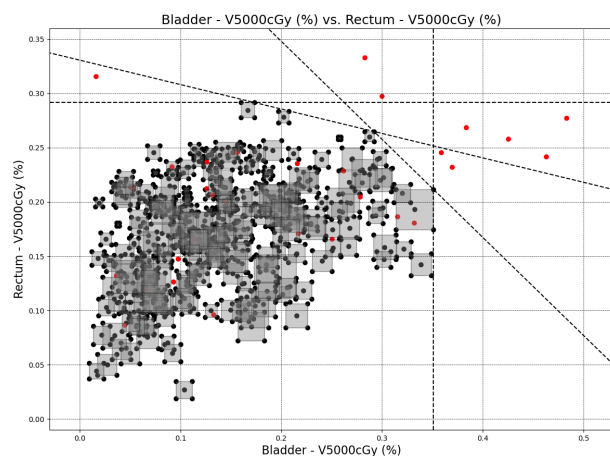


Figure A.4: Bladder - V 5000 cGy (%)
vs. Rectum - V 5000 cGy (%)

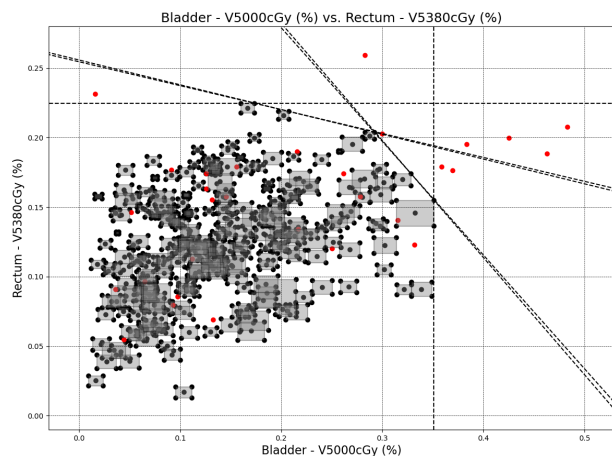


Figure A.5: Bladder - V 5000 cGy (%)
vs. Rectum - V 5380 cGy (%)

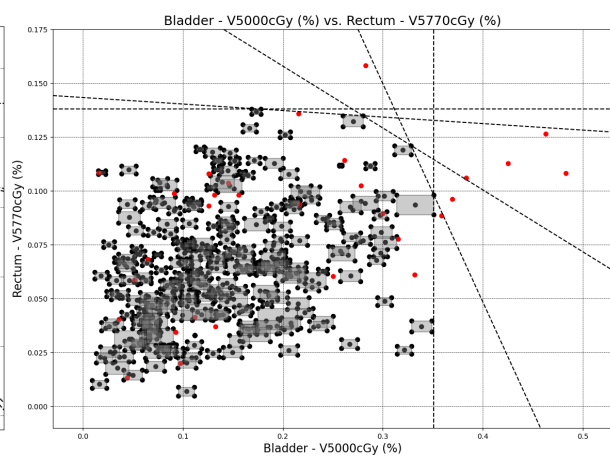


Figure A.6: Bladder - V 5000 cGy (%)
vs. Rectum - V 5770 cGy (%)

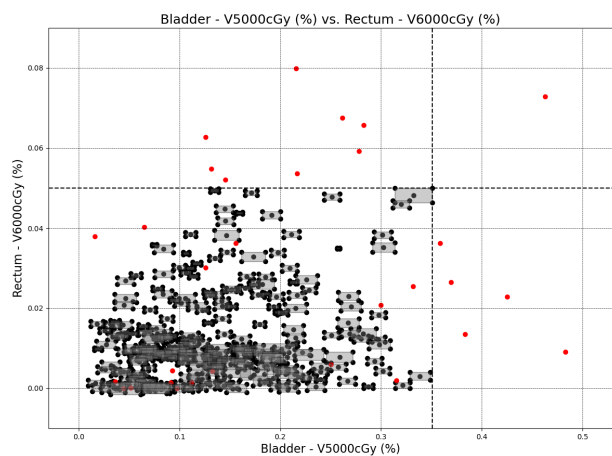


Figure A.7: Bladder - V 5000 cGy (%)
vs. Rectum - V 6000 cGy (%)

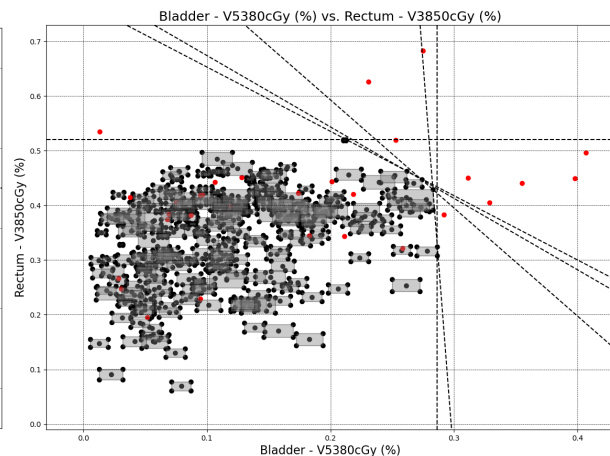


Figure A.8: Bladder - V 5380 cGy (%)
vs. Rectum - V 3850 cGy (%)

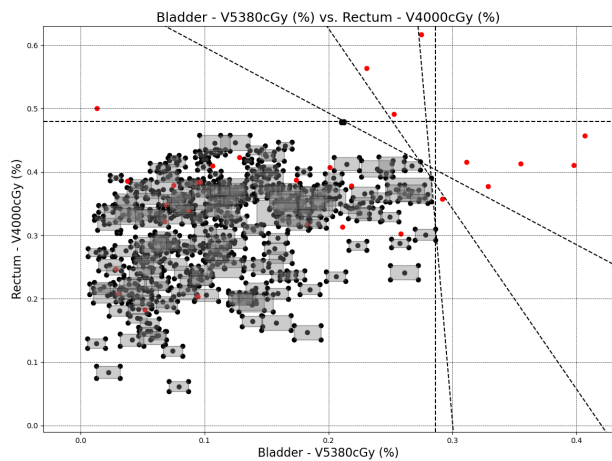


Figure A.9: Bladder - V 5380 cGy (%)
vs. Rectum - V 4000 cGy (%)

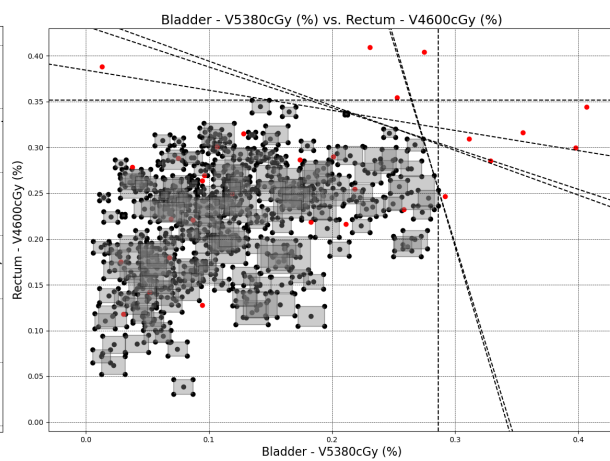


Figure A.10: Bladder - V 5380 cGy (%)
vs. Rectum - V 4600 cGy (%)

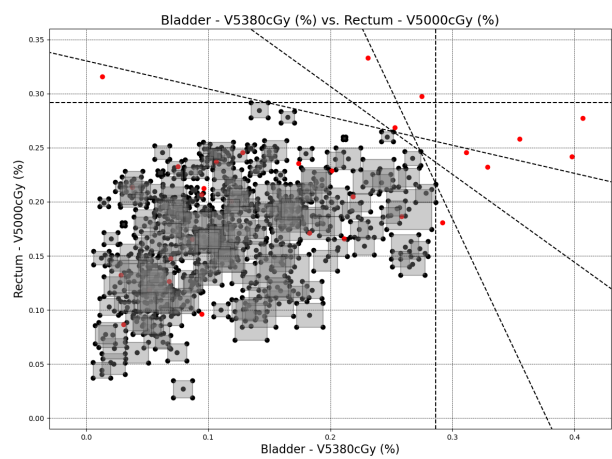


Figure A.11: Bladder - V 5380 cGy (%)
vs. Rectum - V 5000 cGy (%)

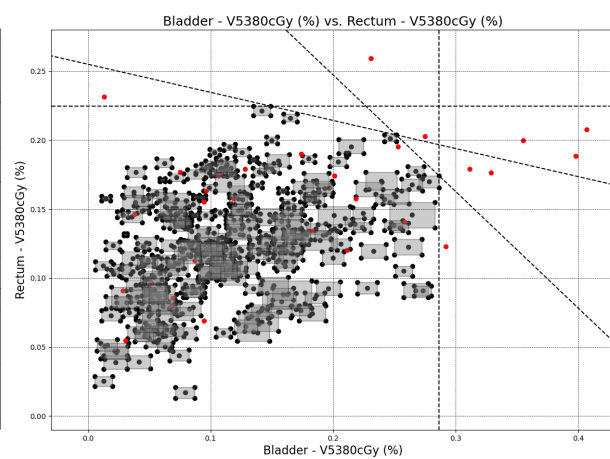


Figure A.12: Bladder - V 5380 cGy (%)
vs. Rectum - V 5380 cGy (%)

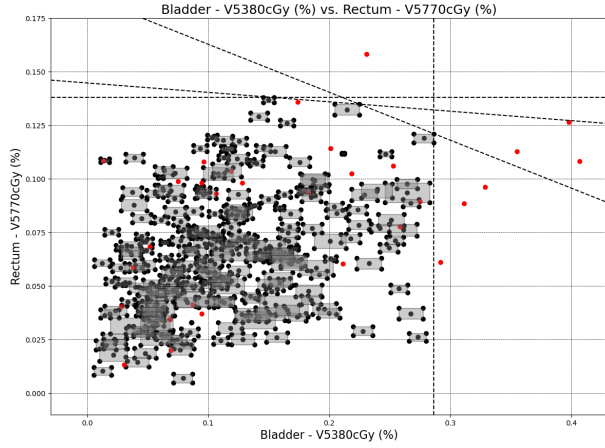


Figure A.13: Bladder - V 5380 cGy (%)
vs. Rectum - V 5770 cGy (%)

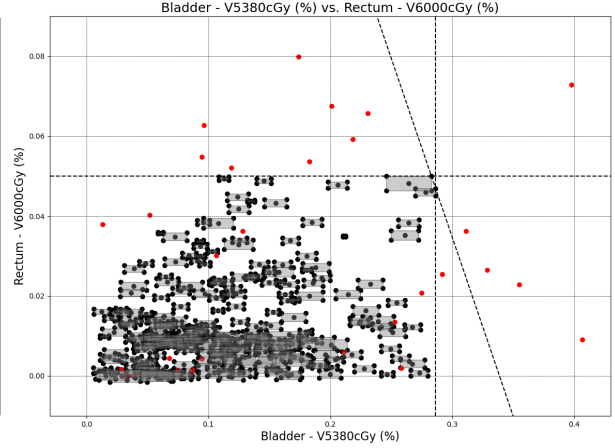


Figure A.14: Bladder - V 5380 cGy (%)
vs. Rectum - V 6000 cGy (%)

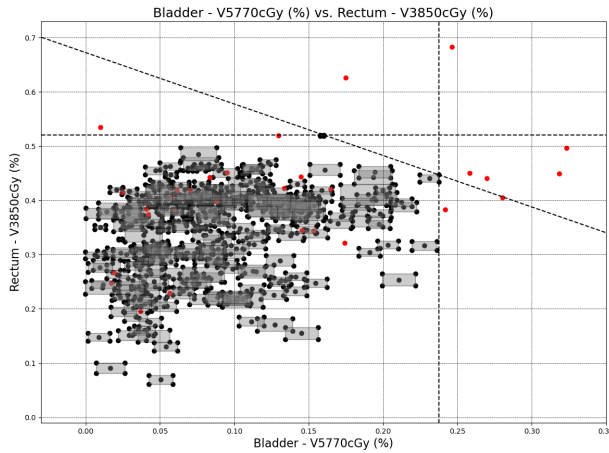


Figure A.15: Bladder - V 5770 cGy (%)
vs. Rectum - V 3850 cGy (%)

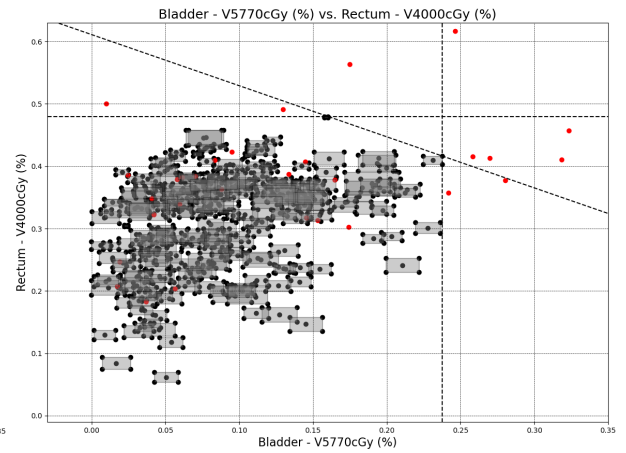


Figure A.16: Bladder - V 5770 cGy (%)
vs. Rectum - V 4000 cGy (%)

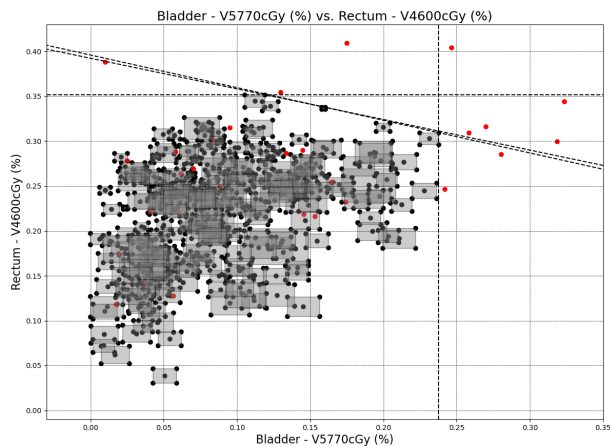


Figure A.17: Bladder - V 5770 cGy (%)
vs. Rectum - V 4600 cGy (%)

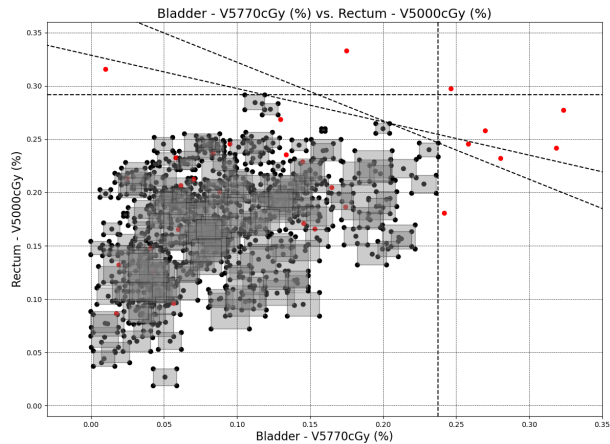


Figure A.18: Bladder - V 5770 cGy (%)
vs. Rectum - V 5000 cGy (%)

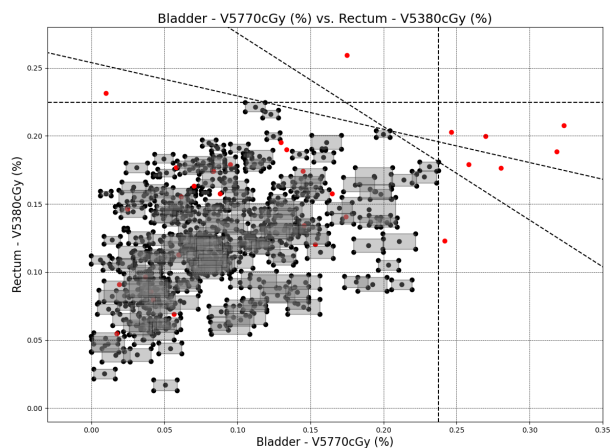


Figure A.19: Bladder - V 5770 cGy (%)
vs. Rectum - V 5380 cGy (%)

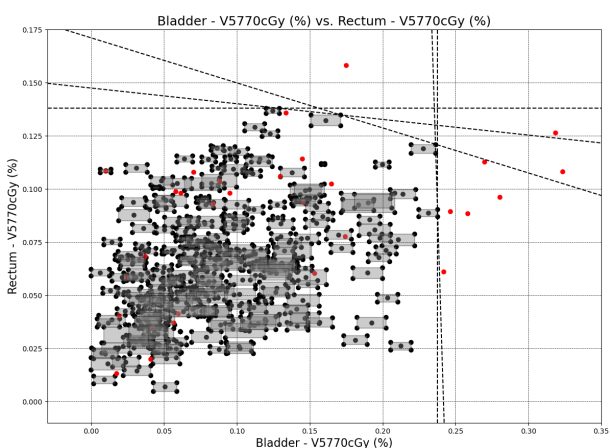


Figure A.20: Bladder - V 5770 cGy (%)
vs. Rectum - V 5770 cGy (%)

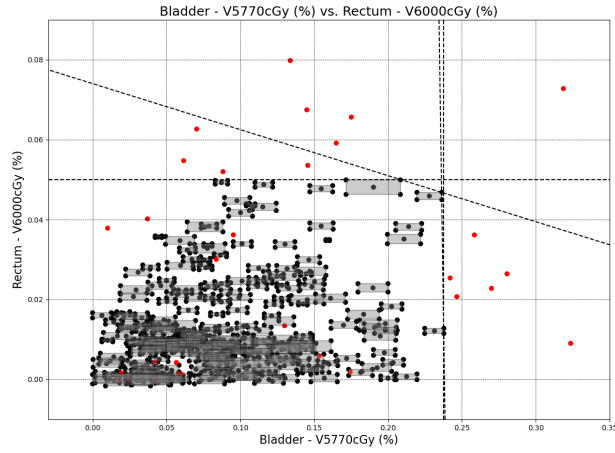


Figure A.21: Bladder - V 5770 cGy (%)
vs. Rectum - V 6000 cGy (%)

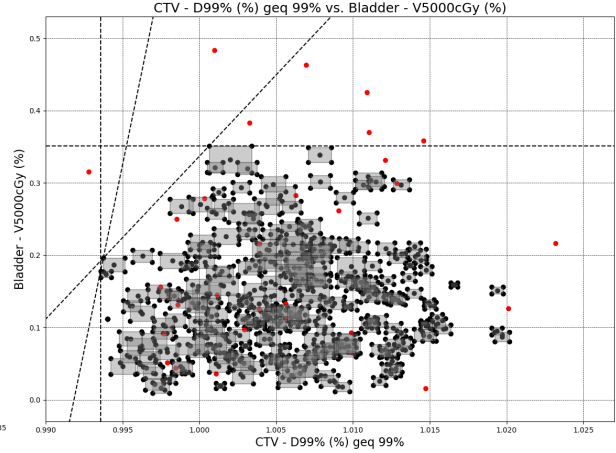


Figure A.22: CTV - D99% (%) $\geq$ 99%
vs. Bladder - V 5000 cGy (%)

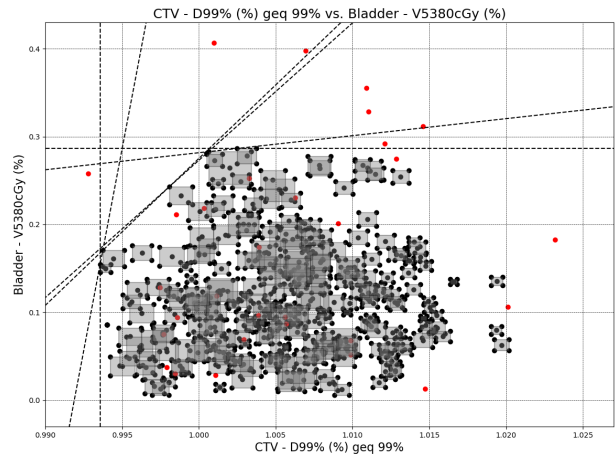


Figure A.23: CTV - D99% (%) $\geq$ 99%
vs. Bladder - V 5380 cGy (%)

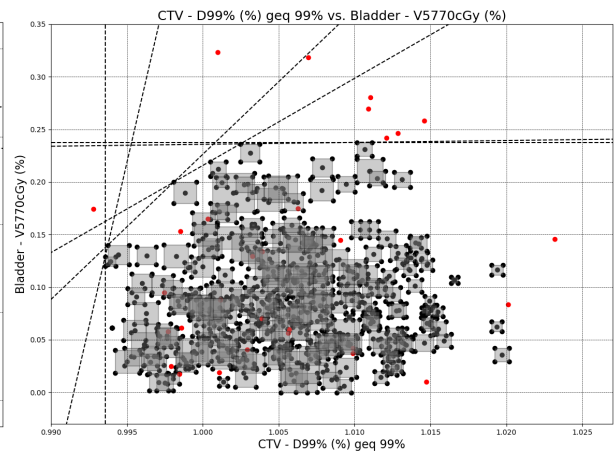


Figure A.24: CTV - D99% (%) $\geq$ 99%
vs. Bladder - V 5770 cGy (%)

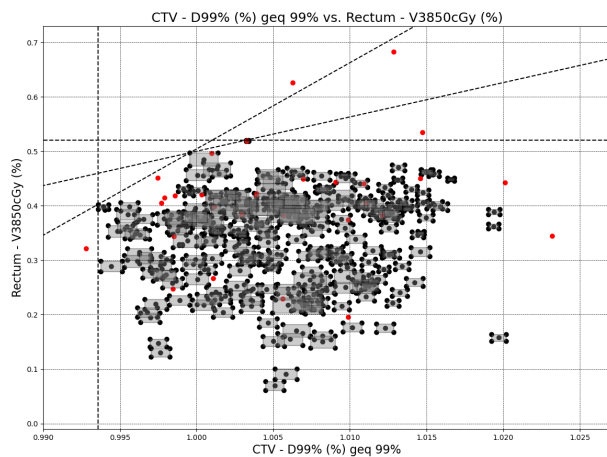


Figure A.25: CTV - D99% (%) $\geq$ 99%
vs. Rectum - V 3850 cGy (%)

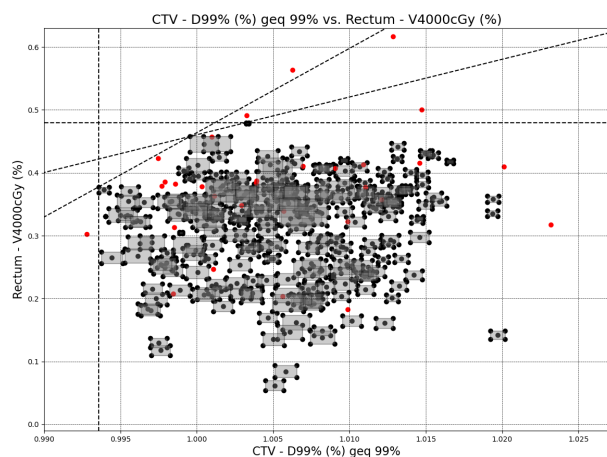


Figure A.26: CTV - D99% (%) $\geq$ 99%
vs. Rectum - V 4000 cGy (%)

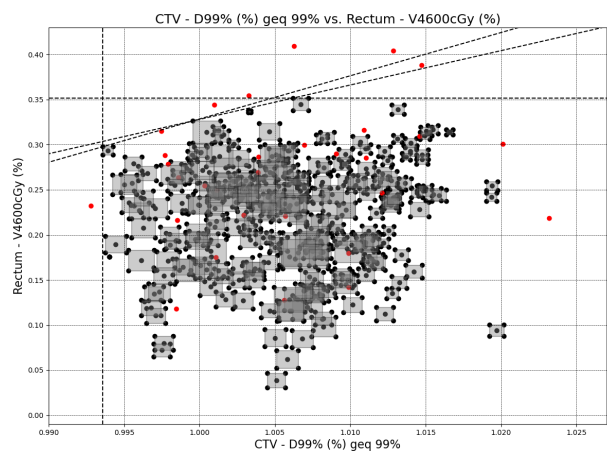


Figure A.27: CTV - D99% (%) $\geq$ 99%
vs. Rectum - V 4600 cGy (%)

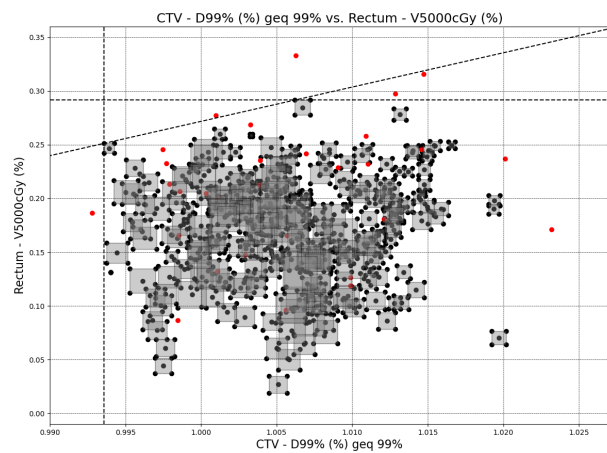


Figure A.28: CTV - D99% (%) $\geq$ 99%
vs. Rectum - V 5000 cGy (%)

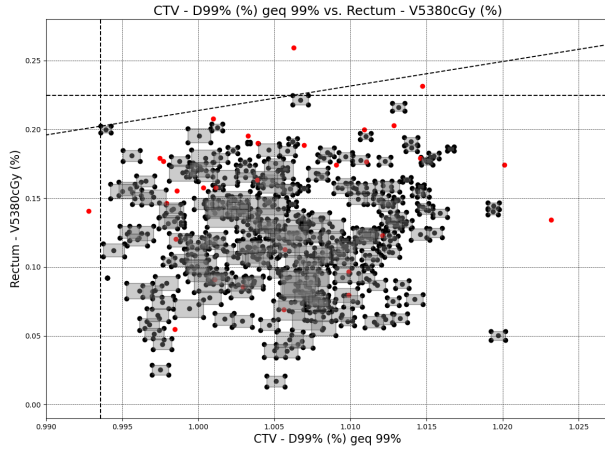


Figure A.29: $\text{CTV} - \text{D99\%} (\%) \geq 99\%$
vs. Rectum - V 5380 cGy (%)

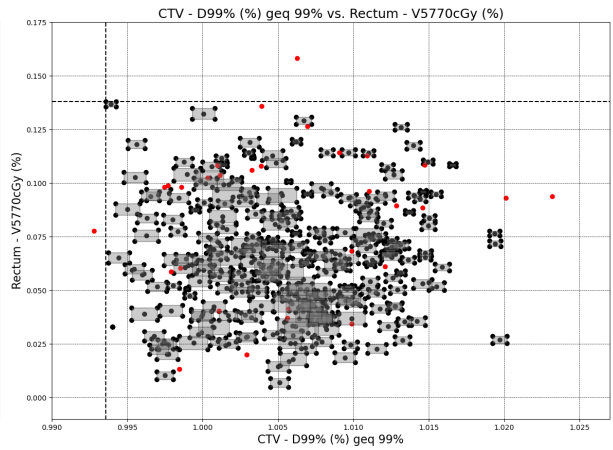


Figure A.30: $\text{CTV} - \text{D99\%} (\%) \geq 99\%$
vs. Rectum - V 5770 cGy (%)

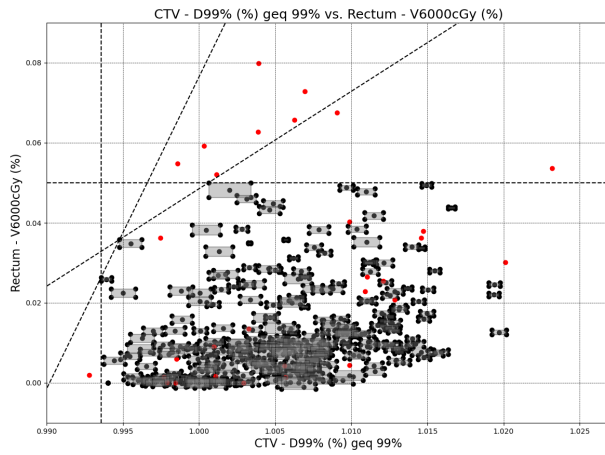


Figure A.31: $\text{CTV} - \text{D99\%} (\%) \geq 99\%$
vs. Rectum - V 6000 cGy (%)

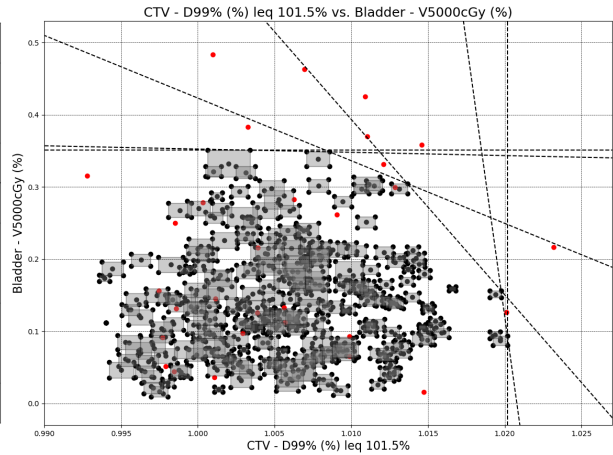


Figure A.32: $\text{CTV} - \text{D99\%} (\%) \leq 101.5\%$
vs. Bladder - V 5000 cGy (%)

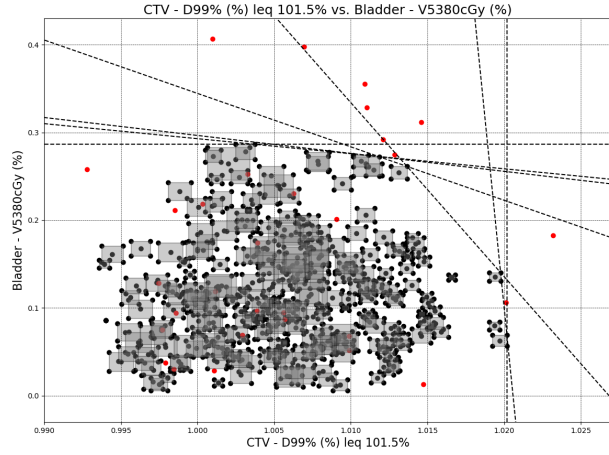


Figure A.33: $\text{CTV} - \text{D99\%} (\%) \leq 101.5\%$
vs. Bladder - V 5380 cGy (%)

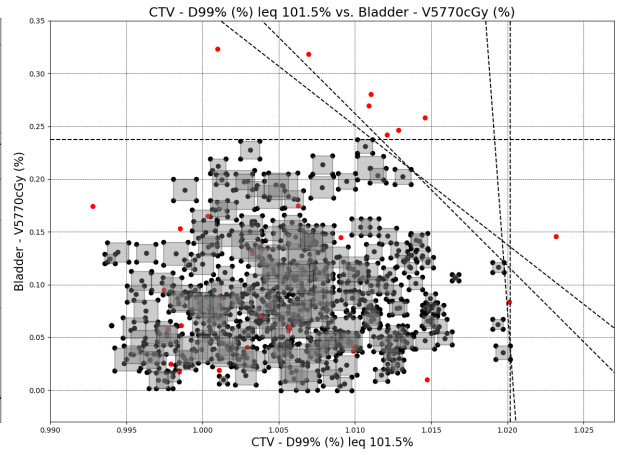


Figure A.34: $\text{CTV} - \text{D99\%} (\%) \leq 101.5\%$
vs. Bladder - V 5770 cGy (%)

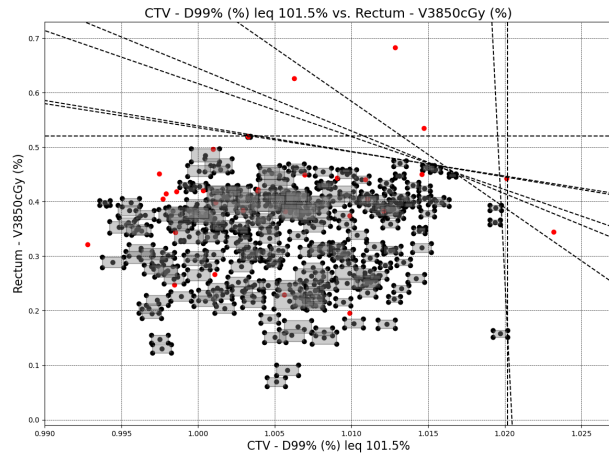


Figure A.35: $\text{CTV} - \text{D99\%} (\%) \leq 101.5\%$
vs. Rectum - V 3850 cGy (%)

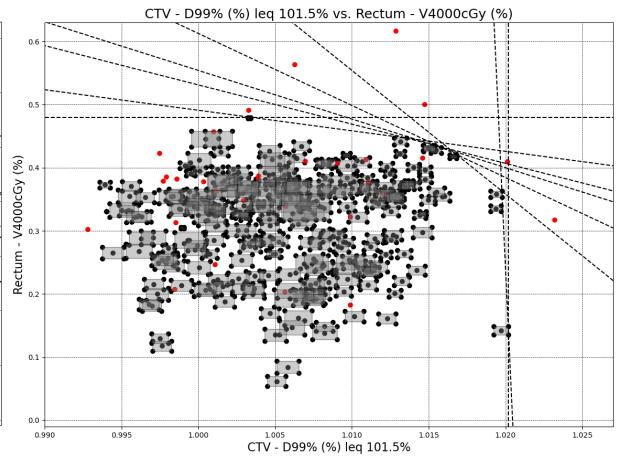


Figure A.36: $\text{CTV} - \text{D99\%} (\%) \leq 101.5\%$
vs. Rectum - V 4000 cGy (%)

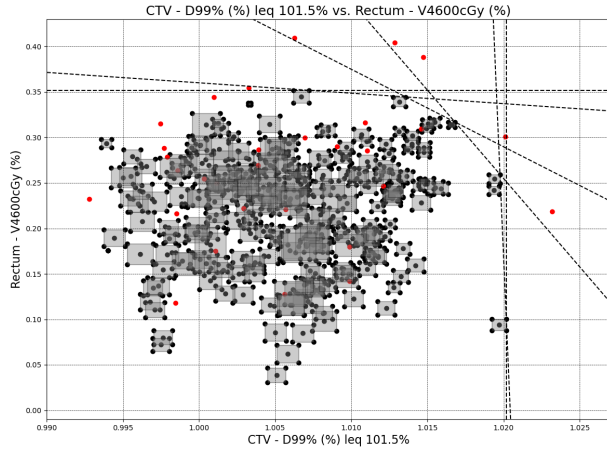


Figure A.37: $\text{CTV} - \text{D99\%} (\%) \leq 101.5\%$
vs. Rectum - V 4600 cGy (%)

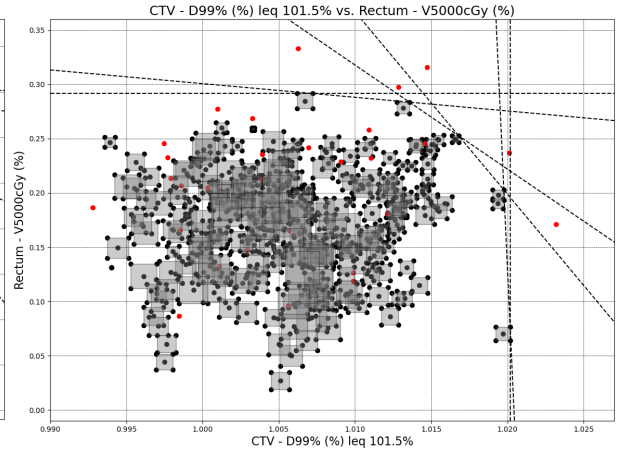


Figure A.38: $\text{CTV} - \text{D99\%} (\%) \leq 101.5\%$
vs. Rectum - V 5000 cGy (%)

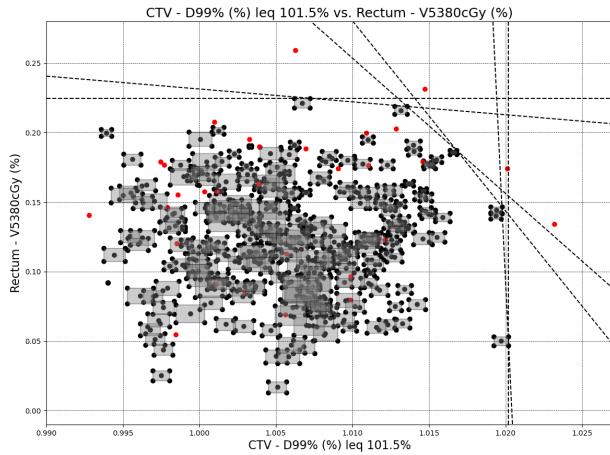


Figure A.39: $\text{CTV} - \text{D99\%} (\%) \leq 101.5\%$
vs. Rectum - V 5380 cGy (%)

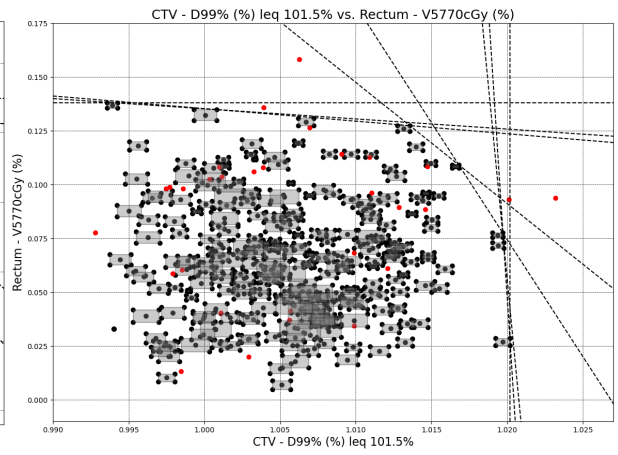


Figure A.40: $\text{CTV} - \text{D99\%} (\%) \leq 101.5\%$
vs. Rectum - V 5770 cGy (%)

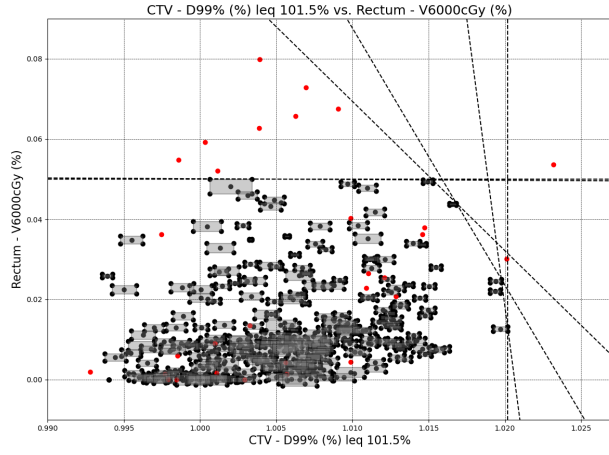


Figure A.41: CTV - D99% (%) $\leq 101.5\%$
vs. Rectum - V 6000 cGy (%)

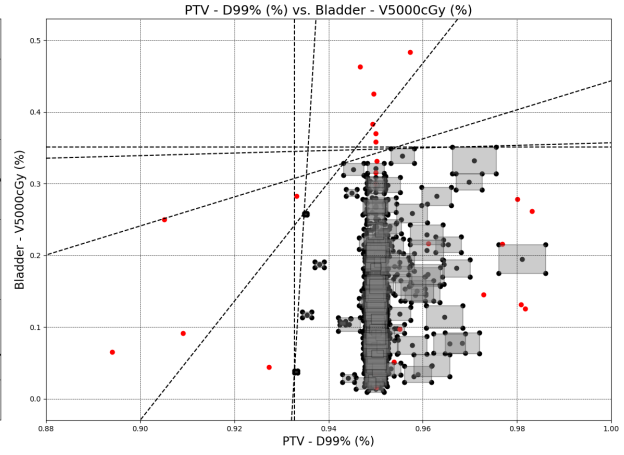


Figure A.42: PTV - D99% (%)
vs. Bladder - V 5000 cGy (%)

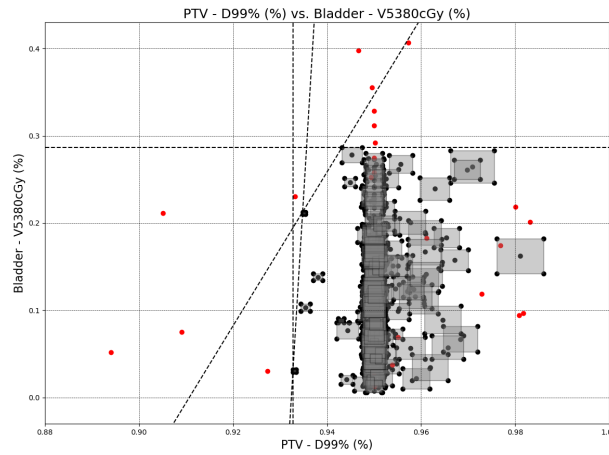


Figure A.43: PTV - D99% (%)
vs. Bladder - V 5380 cGy (%)

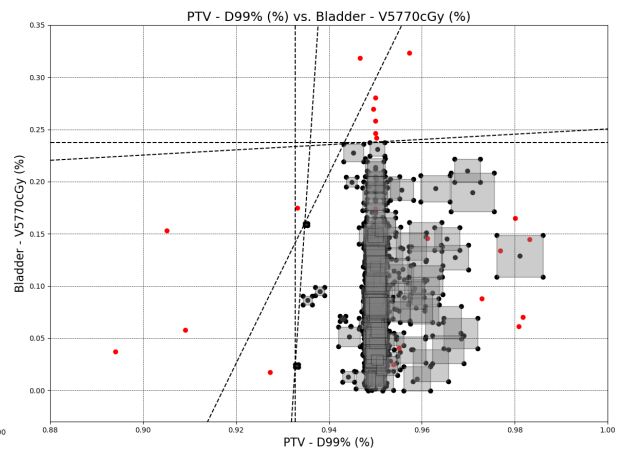


Figure A.44: PTV - D99% (%)
vs. Bladder - V 5770 cGy (%)

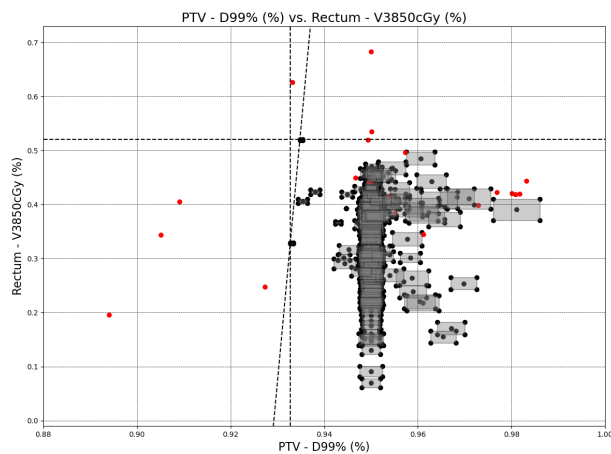


Figure A.45: PTV - D99% (%)
vs. Rectum - V 3850 cGy (%)

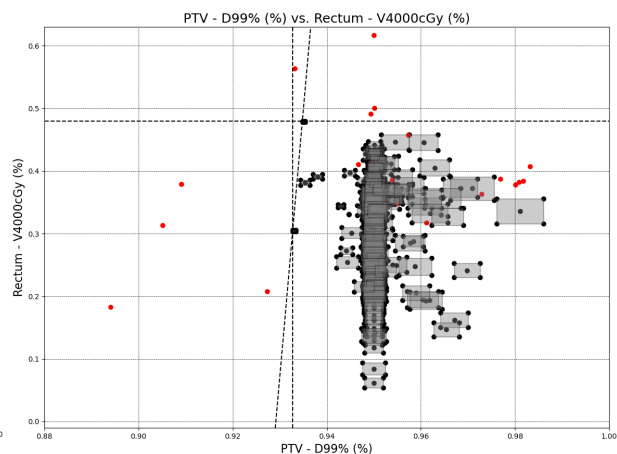


Figure A.46: PTV - D99% (%)
vs. Rectum - V 4000 cGy (%)

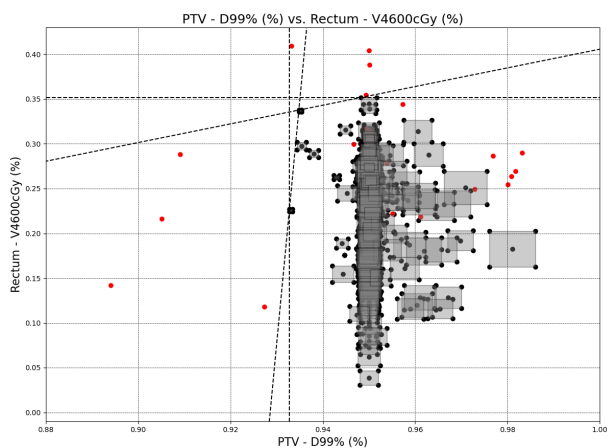


Figure A.47: PTV - D99% (%)
vs. Rectum - V 4600 cGy (%)

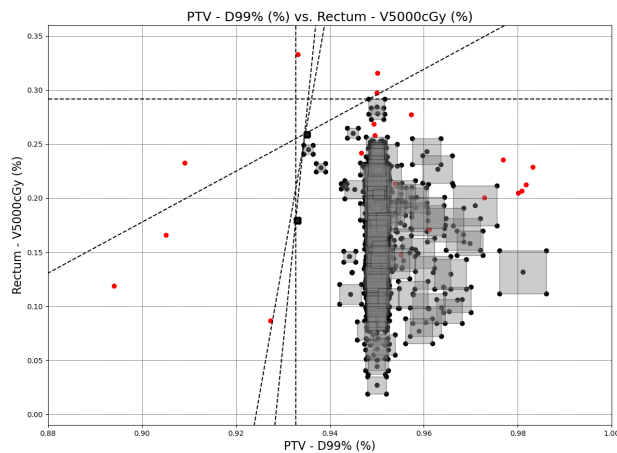


Figure A.48: PTV - D99% (%)
vs. Rectum - V 5000 cGy (%)

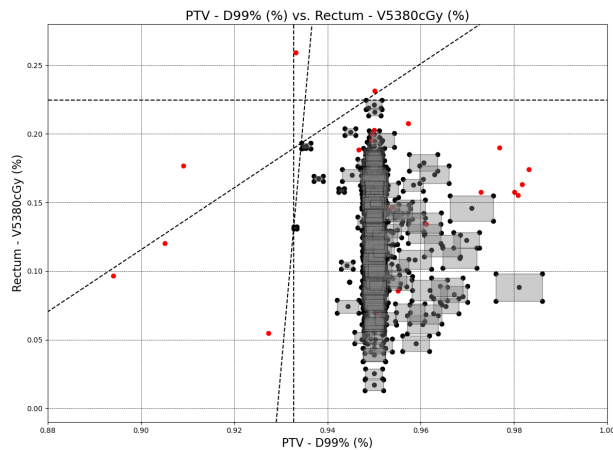


Figure A.49: PTV - D99% (%)
vs. Rectum - V 5380 cGy (%)

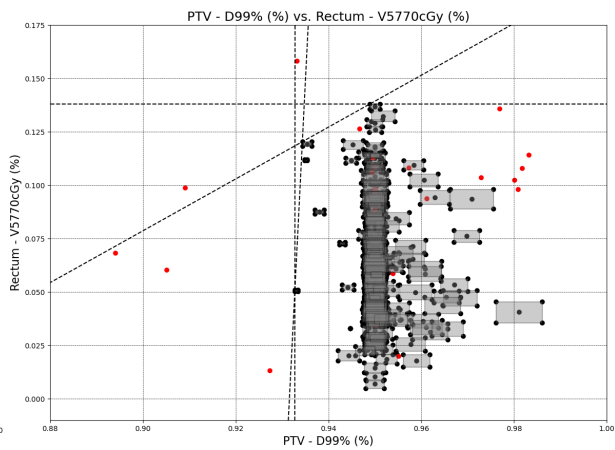


Figure A.50: PTV - D99% (%)
vs. Rectum - V 5770 cGy (%)

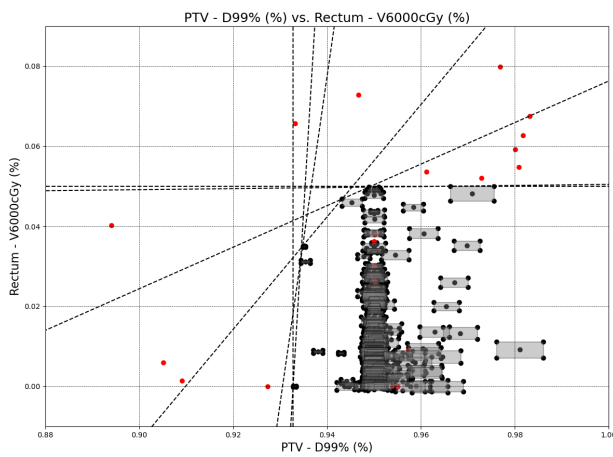


Figure A.51: PTV - D99% (%)
vs. Rectum - V 6000 cGy (%)

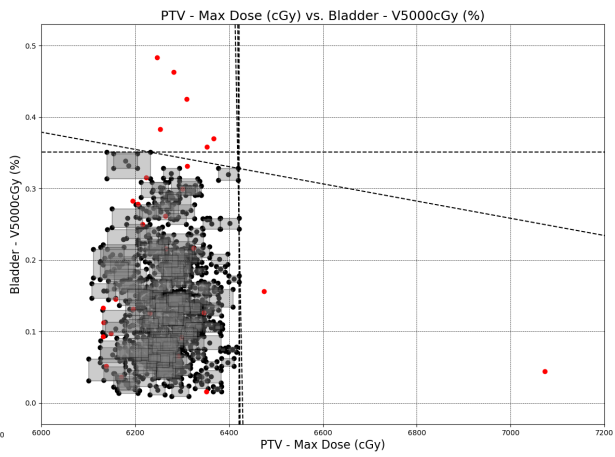


Figure A.52: PTV - Max Dose (cGy)
vs. Bladder - V 5000 cGy (%)

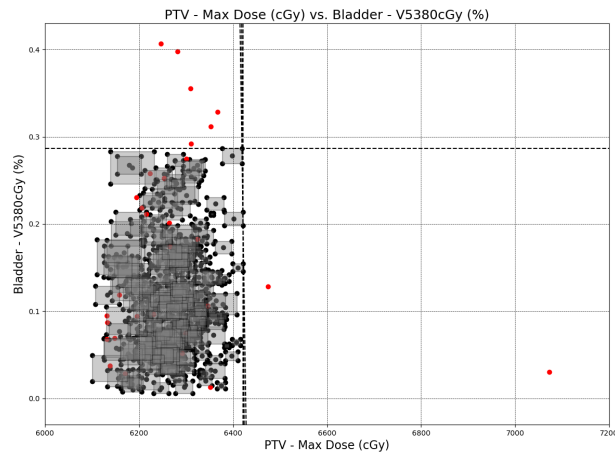


Figure A.53: PTV - Max Dose (cGy)
vs. Bladder - V 5380 cGy (%)

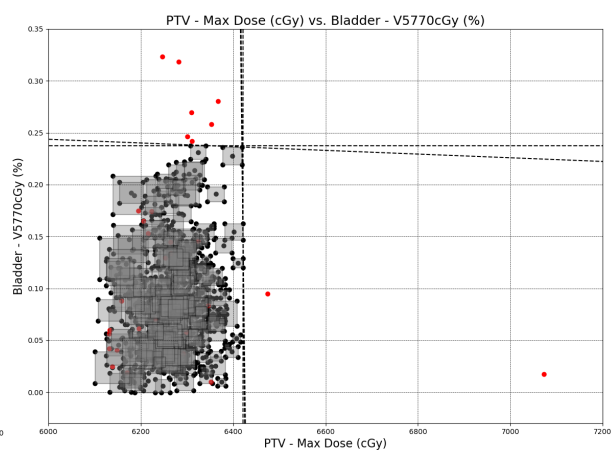


Figure A.54: PTV - Max Dose (cGy)
vs. Bladder - V 5770 cGy (%)

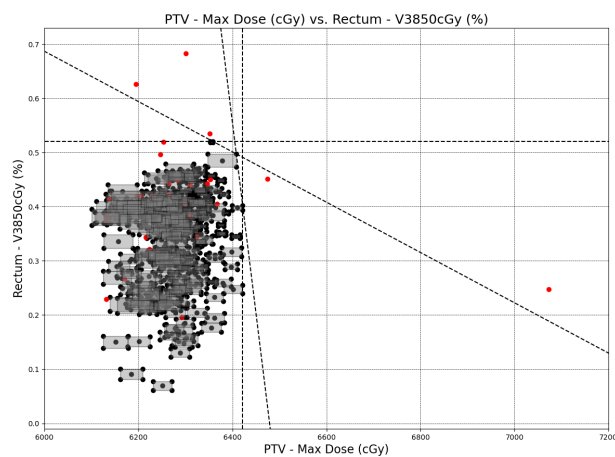


Figure A.55: PTV - Max Dose (cGy)
vs. Rectum - V 3850 cGy (%)

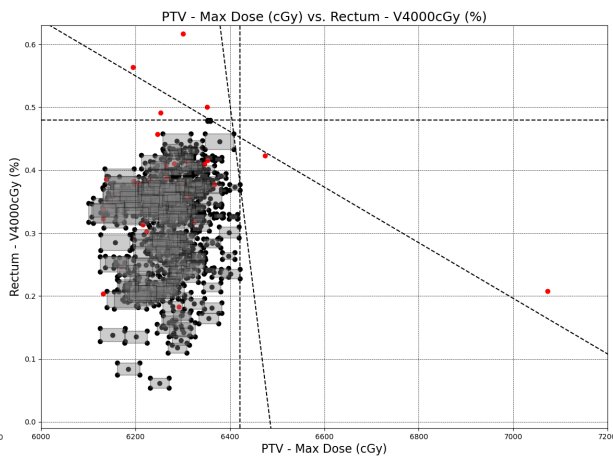


Figure A.56: PTV - Max Dose (cGy)
vs. Rectum - V 4000 cGy (%)

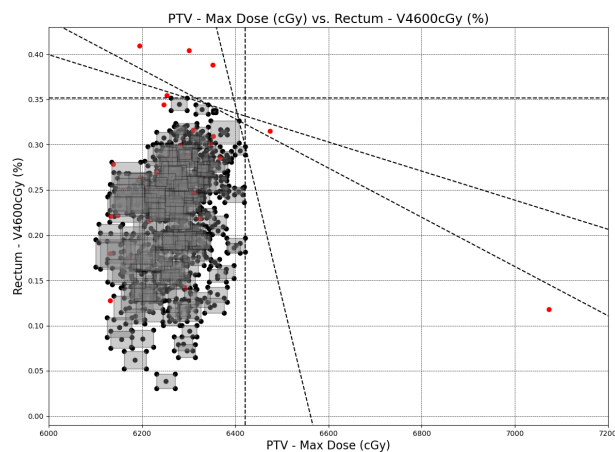


Figure A.57: PTV - Max Dose (cGy)
vs. Rectum - V 4600 cGy (%)

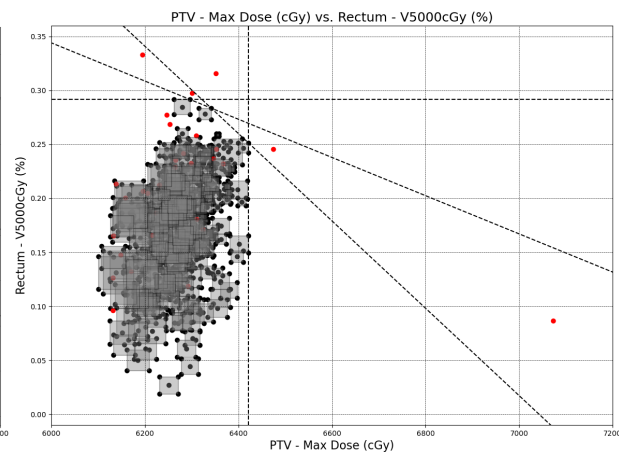


Figure A.58: PTV - Max Dose (cGy)
vs. Rectum - V 5000 cGy (%)

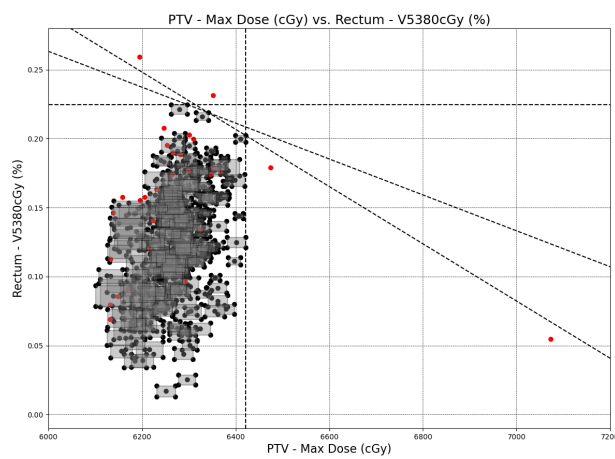


Figure A.59: PTV - Max Dose (cGy)
vs. Rectum - V 5380 cGy (%)

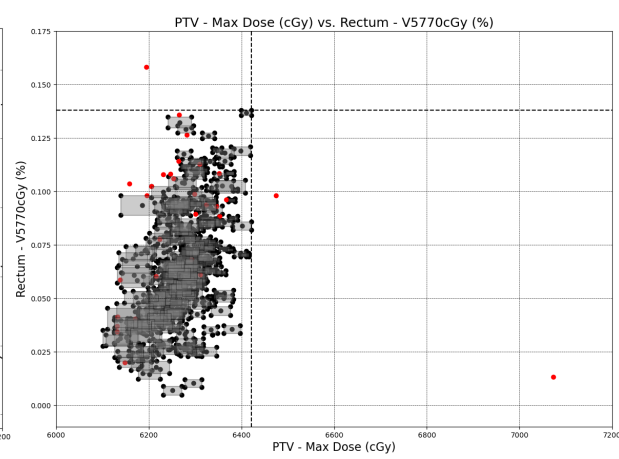


Figure A.60: PTV - Max Dose (cGy)
vs. Rectum - V 5770 cGy (%)

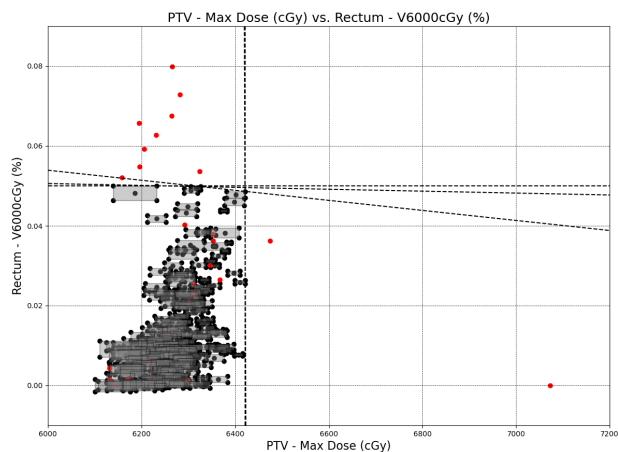


Figure A.61: PTV - Max Dose (cGy)
vs. Rectum - V 6000 cGy (%)

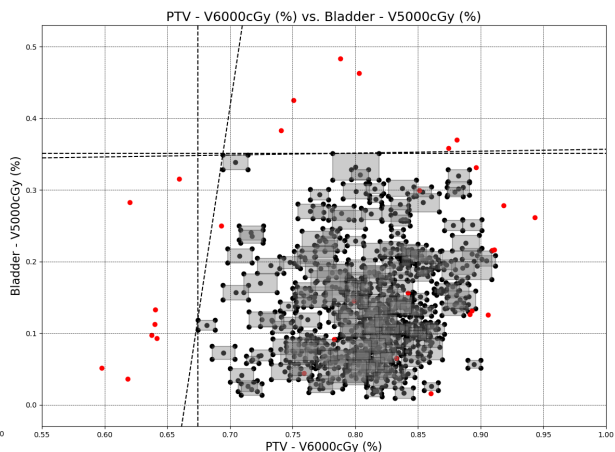


Figure A.62: PTV - V 6000 cGy (%)
vs. Bladder - V 5000 cGy (%)

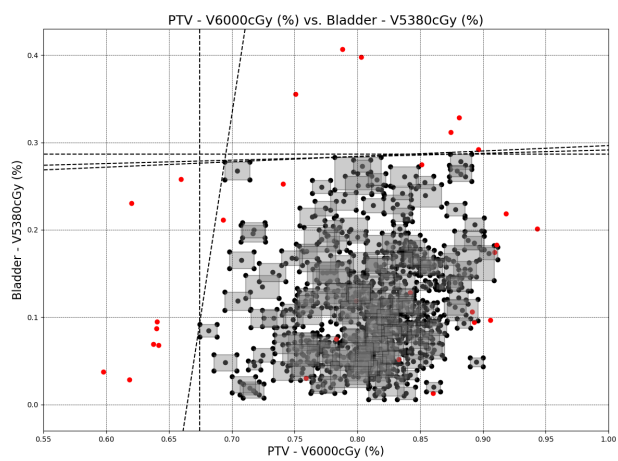


Figure A.63: PTV - V 6000 cGy (%)
vs. Bladder - V 5380 cGy (%)

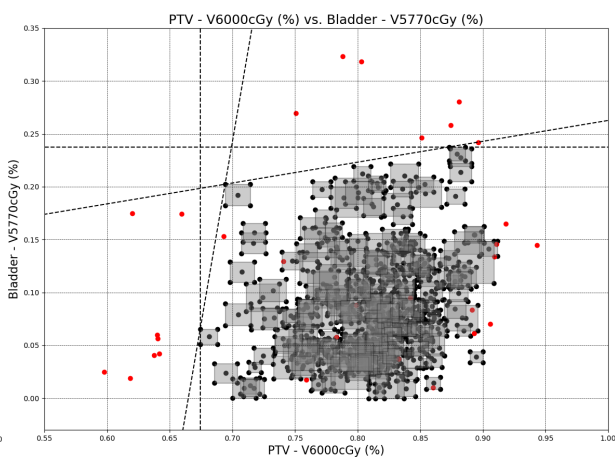


Figure A.64: PTV - V 6000 cGy (%)
vs. Bladder - V 5770 cGy (%)

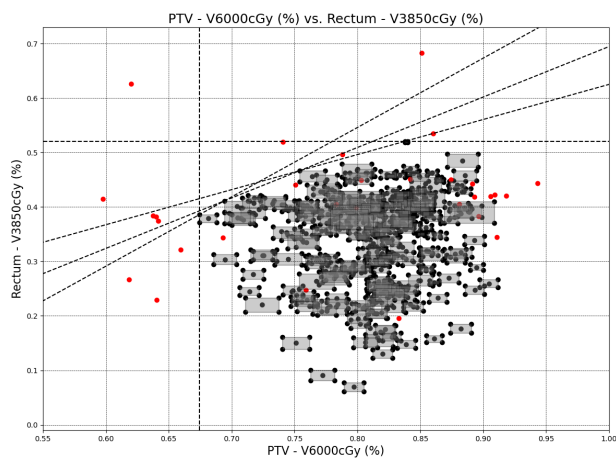


Figure A.65: PTV - V 6000 cGy (%)
vs. Rectum - V 3850 cGy (%)

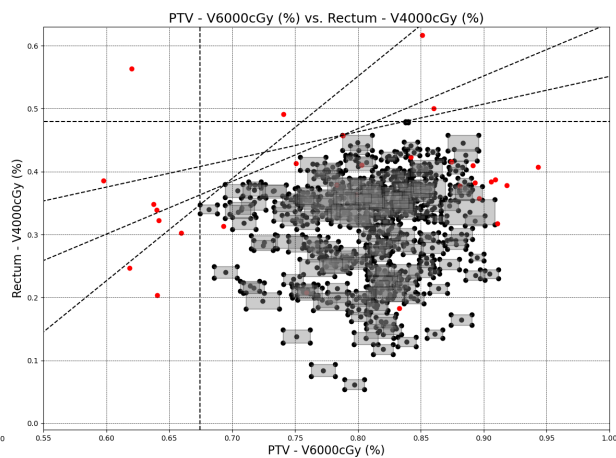


Figure A.66: PTV - V 6000 cGy (%)
vs. Rectum - V 4000 cGy (%)

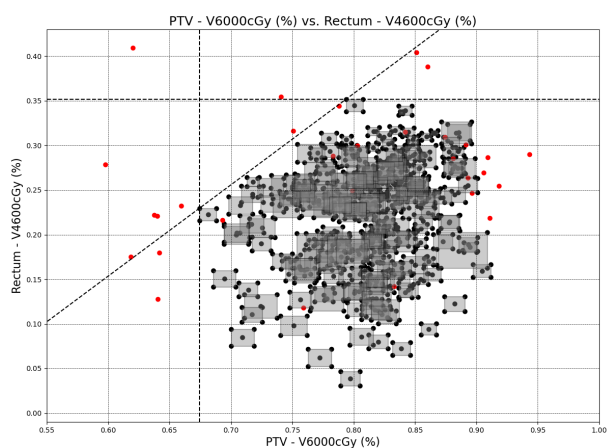


Figure A.67: PTV - V 6000 cGy (%)
vs. Rectum - V 4600 cGy (%)

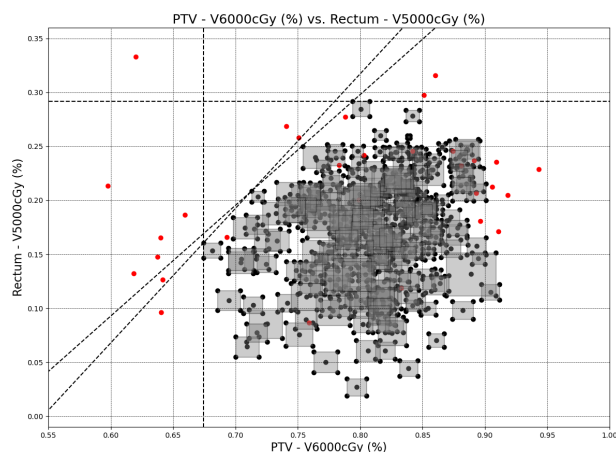


Figure A.68: PTV - V 6000 cGy (%)
vs. Rectum - V 5000 cGy (%)

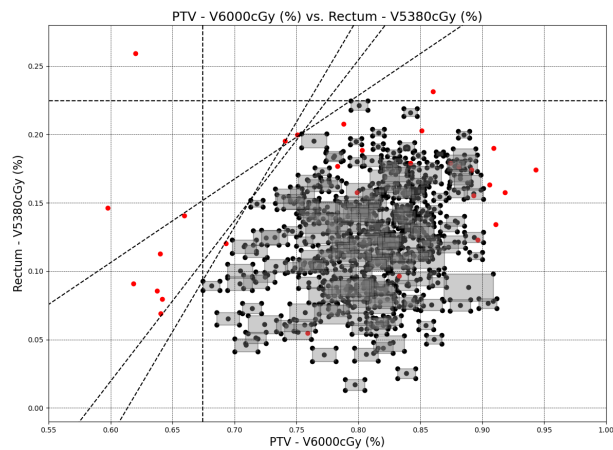


Figure A.69: PTV - V 6000 cGy (%)
vs. Rectum - V 5380 cGy (%)

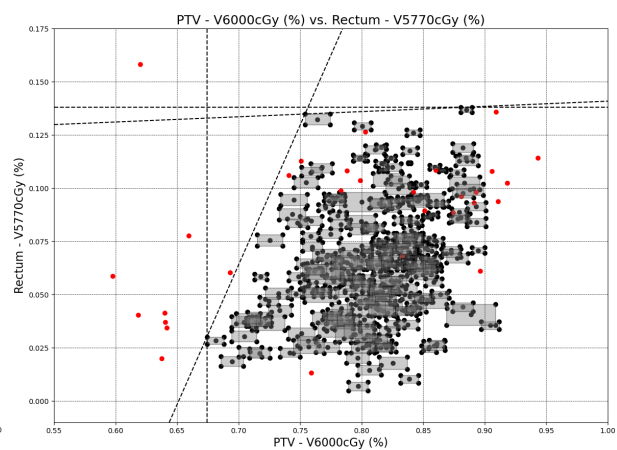


Figure A.70: PTV - V 6000 cGy (%)
vs. Rectum - V 5770 cGy (%)

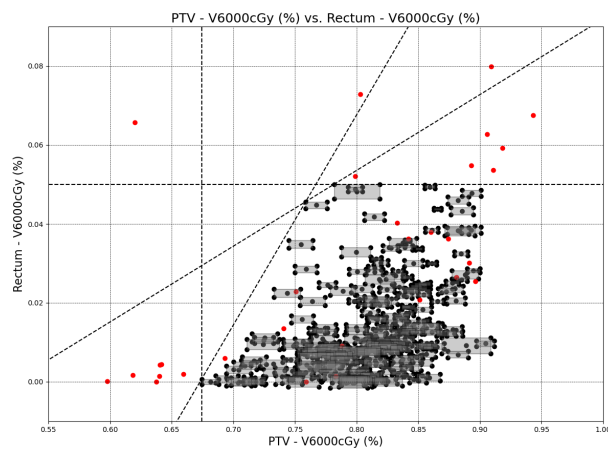


Figure A.71: PTV - V 6000 cGy (%) vs. Rectum - V 6000 cGy (%)