OPTIMIZATION IN RADIATION TREATMENT PLANNING

By

Jinho Lim

A dissertation submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

(Industrial Engineering)

at the

UNIVERSITY OF WISCONSIN – MADISON

2002

Abstract

We present a collection of optimization frameworks for radiation treatment planning problems in this thesis. Firstly, an automated treatment planning framework is developed for the Gamma Knife machine, a specialized unit for the radiation treatment of brain tumors. Nonlinear programs and a mixed integer program are presented to obtain treatment plans. Since nonconvex nonlinear programs do not guarantee global optimality, two techniques are developed to enhance the performance of the optimization models by generating initial starting points. The first technique is a heuristic-based approach to find an initial starting point for the discrete variables of the nonlinear program (the isocenters of radiation doses and the collimator sizes.) This technique uses a variant of a sphere packing approach combined with a Medial Axis Transformation, often used in computer graphics. A linear program is then solved to find the initial radiation exposure time by fixing the values of the discrete variables generated by the above heuristic.

Since the amount of data used in the optimization is very large, an iterative solution scheme for the nonlinear program is presented to speed up the solution process. The optimization problem is first solved using uniformly sampled data points. The resulting solution becomes a starting point for the next optimization process that includes data points previously ignored. The entire treatment planning optimization process for Gamma Knife is fully automated by the combination of these solution processes. This tool is currently in use at the Radiation Oncology Department at the University of Maryland School of Medicine.

Secondly, we present an optimization framework for three-dimensional conformal radiation treatment planning problems that are commonly used to treat patients with cancer of the prostate, lung, and pancreas. Various optimization models are developed for radiation treatment planning. We formulate an optimization problem that simultaneously optimizes beam configurations and beam weights as a mixed integer program. Another optimization model includes wedge filters, which are often placed in front of the beam to produce a gradient in the beam intensity across the aperture. We present several techniques to significantly improve solution time of the model without degrading the solution quality. We also demonstrate that the quality of the dose distribution can be significantly improved by incorporating wedge filters into the optimization. Using our algorithms, both the use (or non-use) of a wedge and the wedge orientation are optimized. We present methods to control the dose volume histogram on organs implicitly using hot and cold spot control parameters in the optimization model.

Finally, MATLAB routines are developed to aid the design of treatment plans. These include: (1) a MATLAB routine to generate appropriate dose matrices based on the beam's-eye-view approach, (2) a variety of GAMS optimization models to solve problems by selecting beam angles, determining wedge orientations, and the beam intensities, and (3) a MATLAB routine to examine the quality of treatment plans. We also provide a MATLAB program that enables the user to create simulated organ structures.

Acknowledgements

I would like to thank all who encouraged me in many different ways during the course of graduate study. Very special thanks go to my advisor Michael Ferris. I am indebted to him for helping me enter the world of mathematical programming. Without his guidance and numerous discussions to improve the material, this dissertation would not be possible.

I am grateful to David Shepard for sharing his practical knowledge on the radiation treatment planning problems. His help was crucial to improve my thesis work. I would like to give many thanks to Stephen Wright for his willingness for the discussions. Robert Meyer and Andrew Miller helped me by sharing their insights on how to improve the thesis material.

I would like to thank Olvi Mangasarian for his help with linear and nonlinear programming concepts. I fully enjoyed his lectures. I cannot thank Michael Smith enough for his support and much-needed help during my graduate program.

There are many others who deserve recognition for their discussions: Stephen Dirkse, Bill Donaldson, Glenn Fung, Birol Kilic, Todd Munson, Leah Newman, Addi Olafsson, Meta Voelker, Chuan Wu, and Vic Zandy.

This thesis is the fruit of my mother who respected, encouraged, and supported her son more than he deserved. I thank all my family, especially my wife Heather, my brother, and my sisters for their endless support for my graduate study.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

The National Cancer Institute estimates that approximately 8.9 million Americans with a history of cancer were alive in 1997. Some of these individuals were considered cured, while others still had evidence of cancer and many have been undergoing treatment. About 1,284,900 new cancer cases are expected to be diagnosed in 2002. Since 1990, about 16 million new cancer cases have been diagnosed. This year about 555,500 Americans are expected to die of cancer, more than 1,500 people a day. Cancer is the second leading cause of death in the US, exceeded only by heart disease. In the US, one of every four deaths is from cancer.

Treatment options are determined by the type and stage of the cancer and include surgery, *radiation therapy*, chemotherapy, immunotherapy, etc. Often a combination of those treatments is used to obtain the best result.

Radiation is a special kind of energy carried by waves or a stream of particles. High dose radiation can be used to treat cancer and other illnesses. It can be delivered from outside of the patient using special machines (teletherapy) or deposited from radioactive substances within the patient (brachytherapy). The use of high-energy rays or particles to treat disease is called radiation therapy. Related terms are radiotherapy, x-ray therapy, or irradiation. In teletherapy, special equipment is used to aim the radiation at tumors or areas of the body where there is disease.

The radiation damages the DNA of the cells in the area being treated, interfering with their ability to divide and grow. Cancerous cells are unable to repair this damage as quickly, so their growth is curtailed and the tumor shrinks. Although some normal cells are affected by radiation, most normal cells appear to recover more fully from the effects of radiation than do cancer cells. Radiation therapy is used to treat solid tumors, such as cancers of the skin, brain, breast, prostate, etc. It can attack cancer cells both on the surface of the body or deep within. It can be used as the sole form of treatment, or in conjunction with surgery (to shrink the tumor before surgery, or to kill remaining cancer cells after surgery) or chemotherapy.

This thesis will consider techniques to improve the delivery of radiation to patients using various forms of technology. The work described will concentrate on using optimization approaches to improve the treatment planning process. Treatment plans are developed to limit the intensity and the area being treated so that the cancer will be affected more than normal tissue. The objective of treatment planning problems is to control the local tumor (target) volume by delivering a uniform (homogeneous) dose of radiation while sparing the surrounding normal and healthy tissue. A major challenge in treatment planning is the presence of organs-at-risk (OARs). An OAR is a critical structure located very close to the target for which the dose of radiation must be severely constrained. This is because overdosing with radiation within the critical structure may lead to medical complications. OAR is also termed as “sensitive structure” or “critical structure” in the literature.

There are two types of radiation treatment planning process: forward planning and inverse planning. In forward planning, treatment plans are typically generated by a trial and error approach. An improved treatment plan is produced by a sequence of experiments with different radiation beam configurations in teletherapy. Due to the complexity of the treatment planning problem, this process, in general, is very tedious and time-consuming, and does not necessarily produce "high-quality" treatment plans. Better strategies for obtaining treatment plans are therefore desired. Due to significant advances in modern technologies such as imaging technologies and computer control to aid the delivery of radiation, there has been a significant move toward inverse treatment planning (it is also called computer based treatment planning). In inverse treatment planning, an objective function is defined to measure the goodness (quality) of a treatment plan. Two types of objective functions are often used: dose-based models and radiobiological models. The biological model argues that optimization should be based on the biological effects resulting from the underlying dose distributions. The treatment objective is usually to maximize the tumor control probability (TCP) while maintaining the normal tissue complication probability (NTCP) to within acceptable levels. Unfortunately, this type of objective function is not rigorously described in the literature and hence it is currently not well suited to optimization approaches. The type of objective function we use throughout the thesis is based solely on dose, in which achieving accurate dose distributions are the main concern. The biological aspect is implicitly given in the physician's prescription.

The inverse treatment planning procedures extend the scope of complexity allowed in treatment planning problems from brachytherapy to external beam therapy. Examples of these more complex plans include conformal radiotherapy, intensity modulated radiotherapy, and tomotherapy. Although many techniques are available to produce treatment plans for each type of radiation therapy, it is important to note that all these problems share some commonalities. These commonalities lead to a notion of a "unified and automated treatment planning process". Potential benefits of the automated treatment planning process can include the reduction in planning time and improved uniformity of dose distributions of treatment plans. Another aspect is that, unlike the conventional trial and error approach, the treatment quality obtained based on the automated treatment planning procedure should depend less on the experience of the treatment planner. However, it should be noted that the treatment goals may vary from one planner to another, from one patient to the next. Therefore an automated treatment planning procedure must be able to self-adjust to these changes and accommodate different treatment goals.

Typical requirements (goals) of radiation treatment planning include *homogeneity, conformity, avoidance,* and *simplicity.* A homogeneity requirement is to irradiate tumor volume within the specified dose levels. It is important for a treatment plan to have uniform dose distributions on the target so that *cold spots* can be minimized. A cold spot is a portion of an organ that receives under its required dose level. On the other hand, the term *hot spot* is used to denote a portion of an organ that receives more than the desired dose level. This requirement can be enforced using lower and upper bounds on the dose, or approximated using penalization. A conformity requirement is used to achieve the target dose control while minimizing the damage to OARs or healthy normal structure. It can be stated as total radiation dose deposited on the target must be at least a specified fraction of the overall dose used for the treatment. As we mentioned earlier, a great difficulty of producing radiation treatment plans is the proximity between the target and OARs. Often acceptable dose levels of these requirements are established by various professional and advisory groups. An avoidance requirement can be used to limit the dose delivered to OARs. Finally, simplicity requirements state that a treatment plan should be as simple as possible. Simple treatment plans typically reduce the treatment time as well as implementation error.

Optimization techniques have become popular in designing these treatment plans automatically. Various treatment goals can be formulated in optimization models. Useful optimization methods are linear programming [78, 55], nonlinear programming [50], mixed integer programming [57], and dynamic programming [3].

In optimization, the three-dimensional volume is represented by a grid of voxels. There are several inputs required in optimization approaches in radiation treatment planning. The first input describes the machine that delivers radiation. The second and troublesome input is the dose distribution of a particular treatment problem. A dose distribution consists of dose contribution to each voxel of the region of interest from a radiation source. It can be expressed as a functional form or a set of data. However, difficulties of using such distributions include high nonlinearity of the functional form or the large amount of data that specifies the dose distribution This problem needs to be overcome in a desirable automated treatment planning tool. The third common input is the set of organ geometries that are of interest to the physician. Further common inputs are the desired dose levels for each organ of interest. These are typically provided by physicians. Other types of inputs can also be specified depending on the treatment planning problems. However, a desirable automated treatment planning tool should be able to generate high quality treatment plans with minimum additional inputs and human guidance.

Therefore, the goal of this thesis is to provide practitioners with a unified and fully automated radiation treatment planning framework in the context of optimization. In this framework, we provide robust optimization models for various treatment planning problems that can be easily incorporated into automated treatment planning tools. The second component of the framework is the ability to quickly produce high quality treatment plans from the resulting large-scale optimization problems. The third component is its reliability in obtaining clinically acceptable treatment plans for various goals and types of treatment problems. Finally, we develop software that can be used to experiement with various models and algorithms for radiation treatment planning.

However, optimization in radiation therapy is too wide to address completely. Hence, we have selected several problems that have practical relevance and for which a worthwhile contribution to the existing literature seems possible. Most of the work presented in the next chapters is based on working papers by the author and collaborators that have appeared or will appear in the literature [27, 28, 48]. We consider two types of radiation delivery mechanism in this thesis. The first system is Gamma Knife machine that is designed for treating brain tumors. We give a general problem description of the Gamma Knife radiosurgery planning problem in Section 1.1. The second radiation delivery system is an X-ray therapy machine that is designed for treating tumors located anywhere in the body. An introduction to conformal radiation treatment planning problem is discussed in

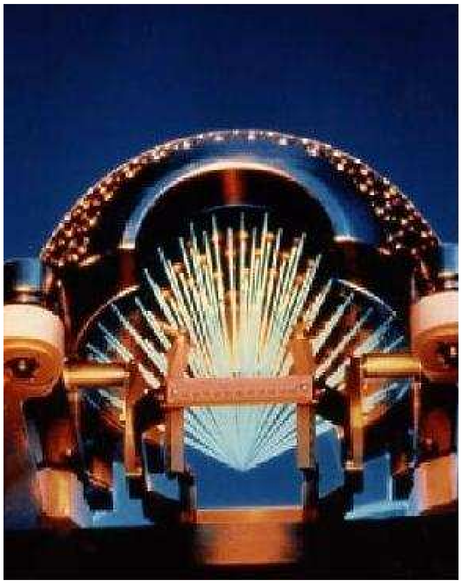
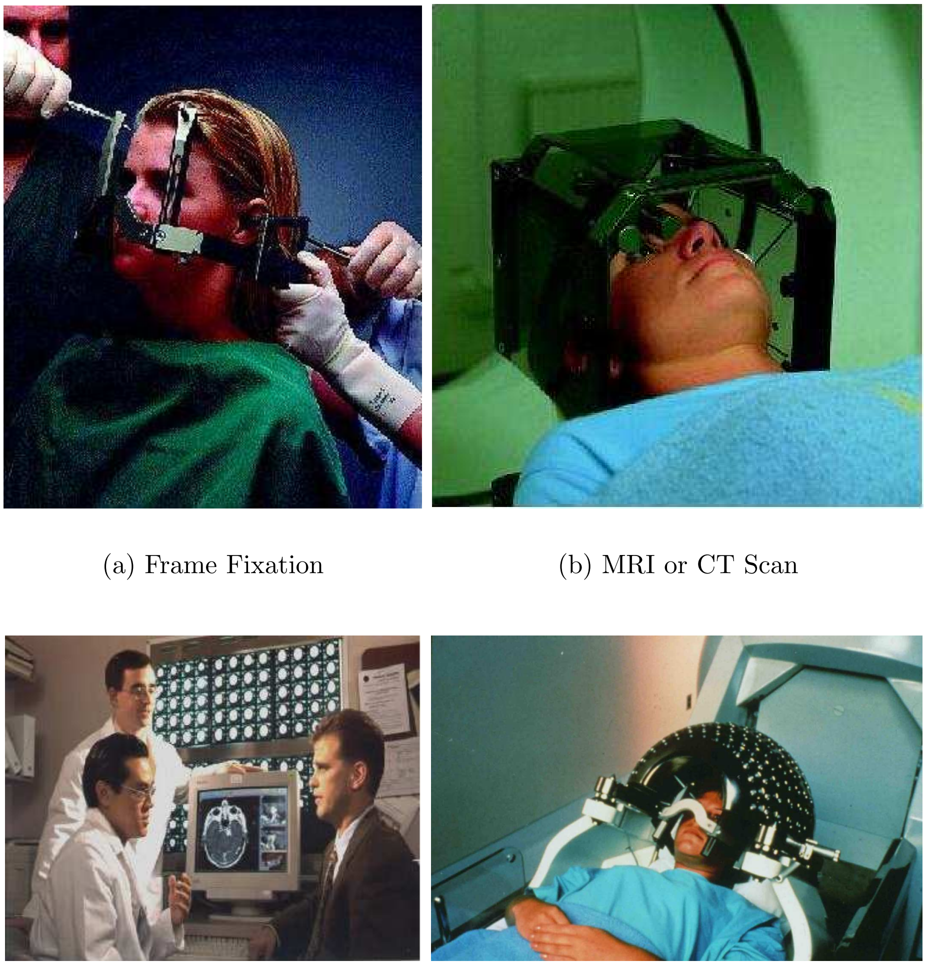


Figure 1: A collimator: A shot of radiation is formed at the intersection of 201 beams

Section 1.2.

1.1 Gamma Knife Radiosurgery Treatment Planning Problem

The Gamma Knife is a highly specialized treatment unit that provides an advanced stereotactic approach to the treatment of tumor and vascular malformations within the head [32]. The Gamma Knife delivers a single, high dose of gamma ray emanating from 201 Cobalt-60 unit sources (Figure 1). Inside a shielded treatment unit, beams from 201 cobalt-60 radioactive sources are focused so that they intersect at a certain point in space, producing a ellipsoidal region of high radiation dose referred to as a *shot*.



(a) Frame Fixation

(b) MRI or CT Scan

(c) Treatment Planning

(d) Radiation Delivery

Figure 2: Gamma Knife Treatment Procedure

A brief history: In 1968, Professor Lars Leskell of the Karolinska Institute in Stockholm, Sweden and Professor Borge Larsson of the Gustaf Werner Institute at the University of Uppsala, Sweden developed the Gamma Knife. As far back as the 1940's, Leskell recognized the need for an instrument to target deep-seated intracranial structures without the risks of invasive open skull surgery. Currently, there are about 200 Gamma Knife machines worldwide.

Treatment Procedure: Gamma Knife Radiosurgery begins by finding the location and the size of the tumor. After administering local anesthesia, a stereotactic coordinate head frame is fixed to the patient's head using adjustable posts and fixation screws (Figure 2(a)). This frame establishes a coordinate frame within which the target location is known precisely and serves to immobilize the patients head within an attached focusing helmet during the treatment. An "magnetic resonance imaging" (MRI) or "computed tomography" (CT) scan is used to determine the position of the treatment volume in relation to the coordinates determined by the head frame (Figure 2(b)). Once the location and the volume of the tumor are identified, the neurosurgeon, the radiation oncologist, and the physicist work together in order to develop the patient's treatment plan (Figure 2(c)). Multiple shots are often used in a treatment using a Gamma Knife due to the irregularity and size of tumor shapes and the fact that the focusing helmets are only available in four sizes (4, 8, 14 and 18mm). Figure 2(d) shows a patient with a collimator attached to the head for the treatment.

The determination of plans varies substantially in difficulty. For example, some tumors are small enough to apply one shot of radiation. On the other hand, when the tumor is large or has an irregular shape or is close to a sensitive structure, many shots of different sizes could be needed to achieve appropriate coverage of the tumor while sparing the surrounding tissue. The treatment planning process can be very tedious and time consuming and due to the variety of conflicting objectives, the quality of treatment plan produced depends heavily on the experience of the user. Therefore, a unified and fully automated Gamma Knife treatment process is desired. Further description of the treatment process, along with some explanatory figures can be found in [30].

Treatment Goal: The plan aims to deliver a high dose of radiation to the intracranial target volume with minimum damage to the surrounding normal tissue. The treatment goals can vary from one neurosurgeon to the next, so a planning tool must be able to accommodate several different requirements. Among these requirements, the following are typical, although the level of treatment and importance of each may vary.

1. A complete 50% isodose line coverage of the target volume. This means that the complete target must be covered by a dose that has intensity at least 50% of the maximum delivered dosage. This can be thought of as a "homogeneity" requirement.

2. To minimize the nontarget volume that is covered by a shot or the series of delivered shots. This requirement is clear and can be thought of as a "conformity" requirement.

3. To limit the amount of dosage that is delivered to certain sensitive structures

close to the target. Such requirements can be thought of as “avoidance” requirements.

In addition to these requirements, it is also preferable to use a small number of shots to limit the treatment times and thus increase the number of patients that can be treated.

1.2 Conformal Radiation Treatment Planning

Radiation Types for Cancer treatment: Common radiation types for cancer treatment are *X-rays and proton* These particles all have somewhat different biological effects on cells.

At present, the most common radiotherapy treatment uses high-energy x rays. Shaped beams of x rays are directed toward the patient (Figure 3). The beams pass through the patient, undergoing near-exponential attenuation as they interact with tissues, and deposit dose along the way. (The strength or dose of radiation is characterized by the energy imparted per unit mass. The unit of dose is the gray; 1 Gy = 1 J/kg.) It is the interactions of secondary electrons, set loose by the primary interactions of the x rays, that are the dominant cause of the molecular disruptions that eventually lead to cell death.

Protons differ from high-energy x rays in that they can deliver radiation dose up to an energy-dependent depth, and virtually none beyond it, whereas x rays continue to penetrate with near-exponentially decreasing intensity.

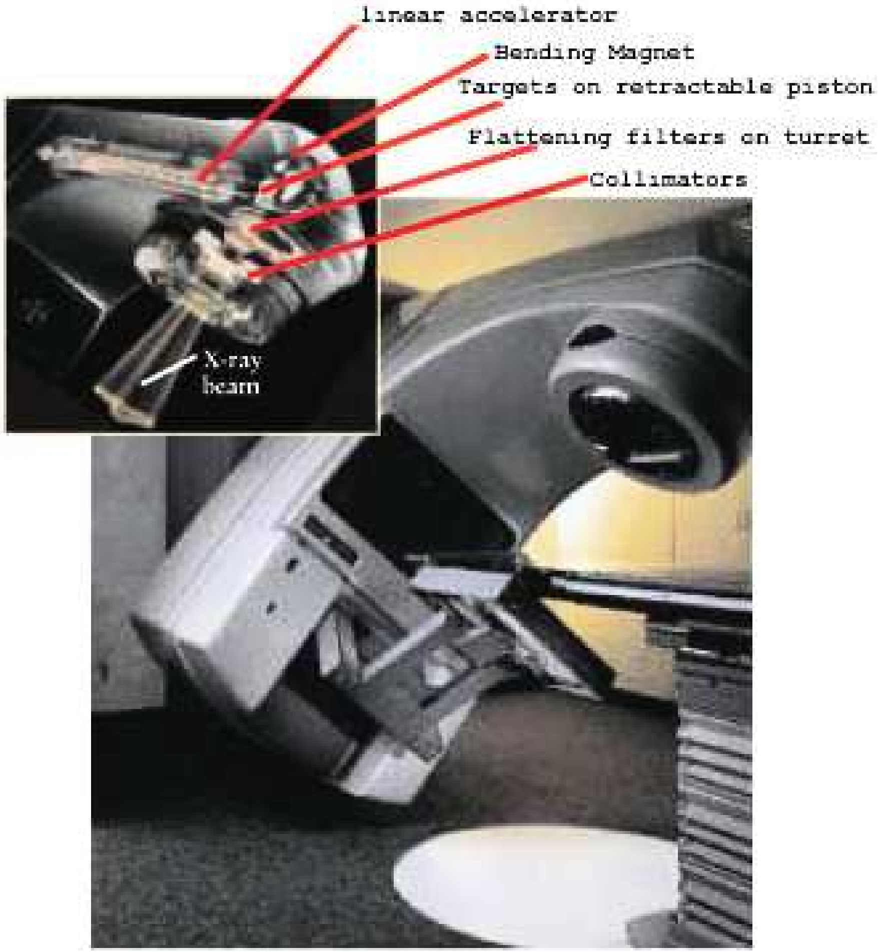


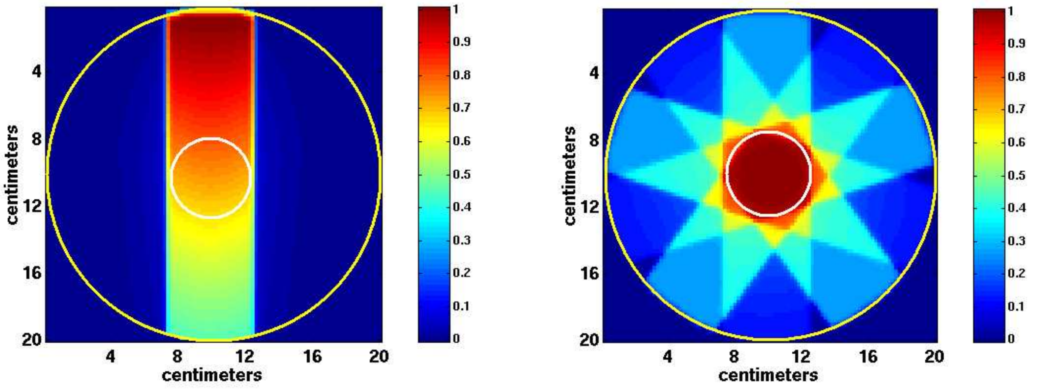
Figure 3: An X-ray therapy machine

Effect of Radiation in Cancer Treatment: Radiation can cause serious damage to organ cells. Secondary electrons create highly reactive radicals in the intracellular material. The radicals can chemically break bonds in the cellular DNA. This damage causes both the malignant cells and the normal cells to lose their ability to reproduce. The higher the dose, the greater the probability of killing cells.

There are two main strategies for normal tissues and organs to continue to function after a treatment with radiation. The first relies on a subtle but favorable difference between the radiation response of normal and malignant cells. That difference can be exploited to preserve the normal cells that permeate the tumor and the nearby tissues that are included in the target volume (that is, the region that includes demonstrable disease; possible sub-clinical extension of that disease, delineation of which depends on the treatment plan; and a safety margin for organ and patient motion and technical uncertainties).

To further the beneficial difference, dose is usually delivered in small daily fractions; this strategy, as compared with single-dose radiation delivery, is generally thought to improve the therapeutic advantage substantially. Consequently, in conventional radiotherapy, from 20 to 30 daily fractions of approximately 2 Gy each are delivered. These fractions are typically delivered once a day, with a two-day weekend break, so that a course of radiotherapy will typically last from four to six weeks. If a person is given 4 to 6 Gy at one time for the treatment period, it could be fatal.

The second strategy for minimizing morbidity is to reduce the dose delivered to normal tissues that are spatially well separated from the tumor. This can be



(a) Single Beam: Tissue on top receives significant dose

(b) Five Beams: a hot spot is formed by five beams

Figure 4: Effect of Multiple Beams

done by using multiple beams from different angles.

Effect of Multiple Beams: A single radiation beam leads to a higher dose delivered to the tissues in front of the tumor than to the tumor itself. In consequence, if one were to give a dose sufficient to control the tumor with a reasonably high probability, the dose to the upstream tissues would likely lead to unacceptable morbidity. A single beam would only be used for very superficial tumors, where there is little upstream normal tissue to damage and the skin-sparing properties of x rays help. For deeper tumors, one uses multiple cross-firing beams delivered within minutes of one another: All encompass the tumor, but successive beams are directed toward the patient from different directions to traverse different tissues outside the target volume. The delivery of cross-firing beams is greatly facilitated by mounting the radiation-producing equipment on a gantry, as illustrated

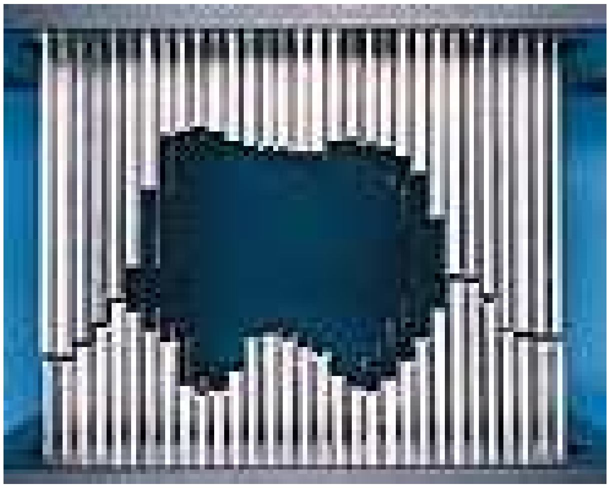


Figure 5: A multileaf collimator

in Figure 3.

Multiply directed beams noticeably change the distribution of dose, as is illustrated in Figure 4. As a result, dose outside the target volume can often be quite tolerable even when dose levels within the target volume are high enough to provide a substantial probability of tumor control.

Beam Shape Generation and Collimator: Radiation treatments are typically delivered using a linear accelerator, Figure 3, with a multileaf collimator, Figure 5, housed in the head of the treatment unit. The leaves of the multileaf collimator are computer controlled and can be moved to the appropriate positions to create the desired beam shape. From each beam angle, three-dimensional anatomical information is used to shape the beam of radiation to match the shape of the tumor. Given a gantry angle, the view on the tumor that the beam source can see through the multileaf collimator is called the *beams-eye-view* of the target [34].

This beams-eye-view (BEV) approach ensures adequate irradiation of the tumor while reducing the dose to normal tissue. Other research focuses on using different configurations of the collimator leaves. While this can be incorporated into our system, we assume throughout this thesis that the beams-eye-view is used.

Wedge Filters: The quality of the dose distribution can be improved by incorporating a wedge filter into one or more of the treatment beams. This metallic wedge varies the intensity of the radiation in a linear fashion from one side of the radiation field to the other. Wedge filters are particularly useful in compensating for a curved patient surface, which is particularly common in breast cancer treatments.

Treatment procedure

1. The patient is immobilized in an individual cast so that the location of treatment region remains the same for the rest of treatment process.

2. A CT scan is performed with the patient in the cast to identify the three-dimensional shapes of organs of interest.

3. Conformal treatment plans are generated using the organ geometries.

4. Treatments are performed 5 times a week for 4 to 5 weeks.

Treatment Goal To be clinically useful, a tool must be safe and efficient. Three requirements for the treatment plan are discussed in Section 1.1: *homogeneity, conformity,* and *avoidance.*

The goal in conformal radiation therapy is to provide a high probability of tumor control while minimizing the damage to the normal tissue. This is accomplished by cross-firing beams of radiation from a number of beam directions. A dosimetrist uses a trial-and-error approach to determine how many beams of radiation are needed, which beam angles are optimal, and what weight should be assigned to each beam. The dosimetrist also needs to determine when a wedge filter is appropriate, the orientation of each wedge, and the weights to be assigned to the wedged and non-wedged beams. Hence, an optimization model should include as variables: *a set of multiple beam angles, wedge orientations,* and *the beam intensities* corresponding to pairs of beam angles and wedge orientations.

Dose-volume control on organs becomes very important for treatment planners. The goal of dose-volume control is to keep the integrated dose (by active beams) of a voxel as close to the prescribed dose level as possible. Often, an acceptable treatment plan requires that nearly all voxels of the target volume are covered with dosages between typical values of 95% and 107% of the prescribed dose, and majority of the organs-at-risk (OAR) should receive less than α_{OAR}% of the prescribed dose, where α_{OAR} is suggested by the physician. The value of α_{OAR} can be different depending on the type of organ. At the same time, the integral dose on the normal tissue should be as small as possible.

1.3 Outline of the Thesis

We give an overview of the contents of this thesis. The key theme of this thesis is optimization in radiation treatment planning. In Chapter 2 we discuss optimization models for the Gamma Knife radiosurgery treatment planning problem. A new dose distribution is presented to approximate the effect of radiation dose as a function of distance from the isocenter (centers of shot locations). Nonlinear programs are presented to obtain treatment plans. Due to the fact that the dose delivery machine does not accept continuous coordinates of isocenters, a mixed integer linear program is used to find the treatment plan after rounding/fixing the continuous coordinates to their nearest discrete isocenters. The nonlinear programs not only require initial starting points, but do not guarantee global optimality. In fact, they may have many local solutions, some of which are close to a global optimal solution, and others may be far off from it. Two techniques are developed in Chapter 3 to enhance the optimization model developed in Chapter 2. First, a three-dimensional skeleton-based heuristic approach is developed to generate an initial starting solution for the isocenters and their corresponding shot sizes. A linear program is solved to find the initial radiation exposure time by fixing the values of the discrete variables given by the heuristic. Secondly, an iterative solution scheme for nonlinear program is presented. Since the amount of data used in the optimization is so large, the optimization problem is first solved using uniformly sampled data points to speed up the solution process. In general, the amount of data used in the first optimization process consists of about 13% of the original data. The resulting solution becomes a starting solution for the next optimization process with data points previously ignored. The entire treatment planning optimization process for Gamma Knife machine is fully automated.

In Chapter 4, we present optimization models for conformal radiotherapy problem using linear, quadratic, and mixed integer programming. We simultaneously optimize three key optimization parameters (beam angles, wedge orientations, and beam weights). Since the optimization models are large-scale, several techniques are addressed to find solutions quickly. We first show how to reduce the solution space by adding a constraint using an input parameter. Secondly, a uniform sampling approach is used to reduce data points on the normal tissue. Finally, an iterative scheme is used to further improve the solution time. In addition, the optimization models implicitly enforce dose-volume constraints discussed in Section 1.2.

To complete the thesis, we develop optimization tools and environments for radiation treatment planning in Chapter 5. Various GAMS models for optimizing radiation treatment are developed. Based on MATLAB environment, a routine is provided to generate necessary data for the optimization models using the MATLAB/C interface. MATLAB routines are developed to plot the dose-volume histogram, and to draw different shapes of structures for experiemts. Finally, we conclude this thesis with a summary in Chapter 6.

Chapter 2

Optimization Models for Gamma Knife Radiosurgery Treatment Planning

2.1 Introduction

We consider treatment planning for a specialized device known as the *Gamma Knife* described in Section 1.1 (see Figure 6.) The approach for treatment planning that will be used here is based on an optimization model of the physical system. Three characteristics are important in the optimization technique for Gamma Knife treatment planning: *speed, flexibility*, and *robustness*. A fast treatment plan is desired primarily for patient comfort. The system must be flexible because the treatment goals vary from patient to patient and neurosurgeon to neurosurgeon. The system also must be robust so that it produces a high quality solution regardless of the size and the shape of the target volume. The solution produced by the optimization must also be practical and implementable.

We assume throughout this chapter that the number of shots that will be delivered is specified to the optimization tool. While other approaches may try to

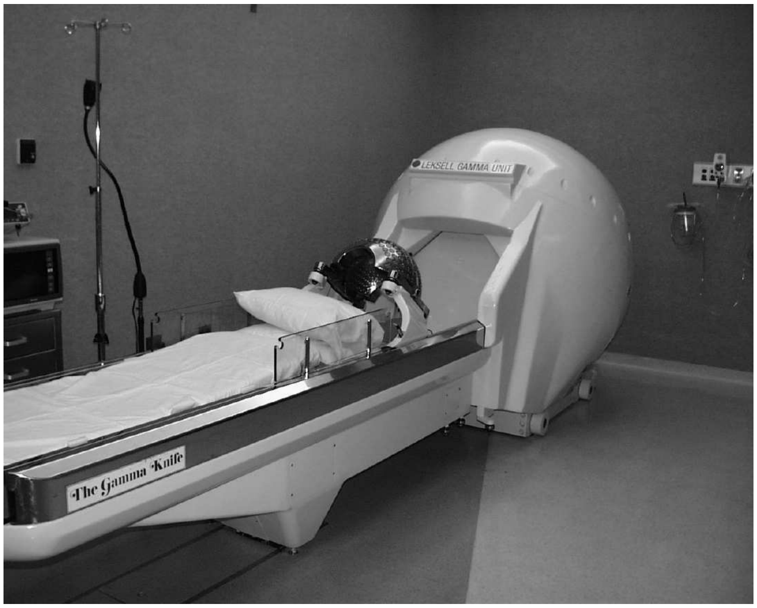


Figure 6: Gamma Knife treatment unit: The patient lies on the couch and is moved back into the shielded treatment area

minimize this number, it is typically straightforward to estimate this number and then develop a plan to optimize other important features for the treatment. In the model we propose, there are three types of decision variables:

1. *A set of coordinates* (x_s, y_s, z_s): for each shot the position of the shot centers is a continuous variable to be chosen.

2. *A discrete set of collimator sizes* $w \in \mathcal{W}$: currently four different sizes of focusing helmets are available (4mm, 8mm, 14mm, 18mm), $\mathcal{W} \in \{4, 8, 14, 18\}$.

3. *Radiation exposure time*: the dose delivered is a linear function of the exposure time.

The remainder of this chapter is organized as follows. A new dose model is described that allows the shots to be modeled as ellipsoids, along with a new conformity estimation problem and a continuation approach to solve the nonlinear program. Section 2.3 reviews existing optimization approaches for solving Gamma Knife radiosurgery treatment planning. We introduce an optimization approach using nonlinear programming in Section 2.4.

2.2 Dose Distribution Models

2.2.1 Existing dose distribution models

The first step in building a treatment planning tool is to model the dose delivered to the patient by a given shot that is centered at a given location. One approach assumes that a shot is approximately spherical [9, 20, 79, 80, 86, 87, 88]. This assumption makes the problem easier to solve. However, more realistic model of dose delivered to a location emanating from a center of a shot has ellipsoidal contours. Monte Carlo simulation techniques for the nonlinear dose model have been commonly used in practice [18, 90].

Cho *et al* [20] present the following spherical dose model. Let x be the distance from the isocenter of a dose sphere, r be a measure of the radius of the sphere, $\Pi(z)$ be a step function where $\Pi(z) = 1$, if $|z| \leq 1/2$, zero otherwise, and $h(x)$ be a dose spread convolution kernel. Then, the radial dose distribution for x can be expressed as

$$g(x) \quad = \quad \Pi\left(\frac{x}{2r}\right) * h(x). \tag{2.1}$$

The asterisk in (2.1) denotes convolution such that, assuming a Gaussian fit,

$$\begin{aligned} h(x) &= \frac{1}{\sqrt{2\pi\sigma}} exp\left(-\frac{x^2}{2\sigma^2}\right), \\ g(x) &= \tfrac{1}{2}\left[erf\left(\frac{x+r}{\sigma}\right) - erf\left(\frac{x-r}{\sigma}\right)\right], \end{aligned} \tag{2.2}$$

where the notation $erf(\cdot)$ represents the integral of the standard normal distribution from $-\infty$ to x..

2.2.2 A new dose distribution model

The complete dose distribution can be calculated as a sum of contributions from each shot delivered, once the location of the center of that shot (x_s, y_s, z_s) is known, and the length of time of delivery $t_{s,w}$ is known. In practice this means that for all (i, j, k)

$$Dose(i, j, k) = \sum_{(s,w)\in\mathcal{S}\times\mathcal{W}} t_{s,w} D_w(x_s, y_s, z_s, i, j, k), \tag{2.3}$$

where $D_w(x_s, y_s, z_s, i, j, k)$ is the dose delivered to the voxel (i, j, k) by the shot of width w centered at (x_s, y_s, z_s).

Our dose model is based on [30]. They simulated the delivery of a shot of width $w \in \mathcal{W}$, centered at the middle of the head of a previously scanned patient on the Gamma Knife. For each shot width, they determined the dose delivered in the x, y and z directions at given distances from the center of the shot from the simulation. The three values were then averaged to give a value of dose (for each width of shot) at a particular distance from the center. However, we observed that the actual dose delivered was ellipsoidal in nature rather than spherical, so we determined the principal axes and measured the values of dose $\bar{D}_w$ along them.

In practice, the axis location depended on whether the patient was lying prone or supine, and thus we rotate the target so its coordinate axes lie along the ellipsoid's principal axes in either case.

The problem is thus reduced to determining a functional form for the dose delivered at a voxel (i, j, k) from the shot centered at (x_s, y_s, z_s). A sum of error functions has been noted in the literature to approximate this dose distribution [20, 41]. We therefore used the following functional form

$$\begin{aligned} &D_w(x_s, y_s, z_s, i, j, k) = \\ &\quad \sum_{p=1}^{2} \lambda_p \left(1 - \operatorname{erf}\left(\frac{\sqrt{(i - x_s)^2 + \mu_p^y (j - y_s)^2 + \mu_p^z (k - z_s)^2} - r_p}{\sigma_p} \right) \right) \end{aligned} \tag{2.4}$$

and fit the ten parameters λ_p, μ_p^y, μ_p^z, r_p and σ_p to the data described above via least-squares, with different values for each shot width. The resulting nonlinear optimization problem

$$\min_{\lambda,\mu,r,\sigma} \left\| \begin{pmatrix} \bar{D}_w^x(i) - \sum_{p=1}^{2} \lambda_p \left(1 - \operatorname{erf}\left(\frac{\sqrt{(i - x_s)^2} - r_p}{\sigma_p} \right) \right) \\ \bar{D}_w^y(j) - \sum_{p=1}^{2} \lambda_p \left(1 - \operatorname{erf}\left(\frac{\sqrt{\mu_p^y (j - y_s)^2} - r_p}{\sigma_p} \right) \right) \\ \bar{D}_w^z(k) - \sum_{p=1}^{2} \lambda_p \left(1 - \operatorname{erf}\left(\frac{\sqrt{\mu_p^z (k - z_s)^2} - r_p}{\sigma_p} \right) \right) \end{pmatrix} \right\|^2$$

was solved using CONOPT. These values were then fixed in the nonlinear models used in the remainder of this thesis.

2.3 Existing Optimization Approaches

A number of researchers have studied techniques for solving the Gamma Knife treatment planning optimization [49, 87]. One approach incorporates the assumption that each shot of radiation can be modeled as a sphere. The problem is then reduced to one of geometric coverage, and a ball packing approach [74, 80, 87] can be used to determine the shot locations and sizes. The use of a modified Powell's method in conjunction with simulated annealing has been proposed [49, 93]. The paper [71] presents a simulated annealing approach incorporating a quasi-Newton method. A mixed integer programming and a nonlinear programming approach for the problem is presented in [30, 68]. We briefly review some of the existing optimization approaches in this section.

Sphere Packing Under the assumption that a shot is a non-elastic, solid 3D sphere, the paper [80] presents an optimization of packing unequal spheres into a three-dimensional bounded region in connection with radiosurgical treatment planning. Given an input (R, V, S, L), where R is a 3D bounded region, V a positive integer, S a multiset of spheres, and L a location constraint on spheres, we want to find a packing of R using the minimum number of spheres in S such that the covered volume is at least V, the location constraint L is satisfied; and the number of points on the boundary of R that are touched by spheres is maximized.

Wang [80] shows that not only finding an optimal solution to the problem is computationally intractable, but also optimization of the related problems is NP-hard. Therefore, some sort of approximation is needed. The paper [80] proposes

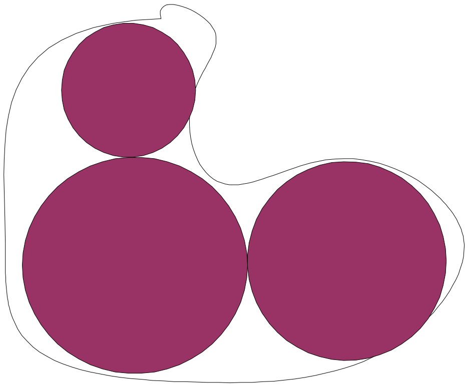


Figure 7: Sphere Packing Example in 2D: three circles cover inside of the target region assuming that a shot of radiation forms a circle

an improved approach to [9, 88] under the assumption that spheres of arbitrary diameters are available with unlimited supply, and there are no location constraints.

Dynamic Programming *Dynamic Programming* has been used to determine the number of shots, shot center locations and their sizes under the assumption that a shot is a sphere [9, 80, 87, 88]. Therefore the dose calculation is eliminated by finding the geometrical coverage. Suppose we apply a sequence of shots to the target area. It means that we determine the first shot location and its size, then apply the shot to the target area. The second shot location and and its size is determined on the remaining target area, then apply the second shot to the target area. This process continues until entire target area is covered within a tolerance. This is the main idea of using *Dynamic Programming* method. We review the paper [9]. By assuming a shot is approximately sphere, they find geometric coverage of a region using spheres without overlapping. A skeleton-based method is used. The justification of using this method is that dosimetric coverage is normally wider than the geometrical coverage. The dose distribution of a shot is not a simple step function with zero dose outside and positive dose inside of the coverage. There is finite dose delivered to outside of the geometrical coverage. For an irregular shaped target, the optimizer will prefer small shots to match the external boundary. This will result in an impractical solution to the treatment planning. Therefore, the number of shots to use must be included in the optimization parameter. They assume that an optimal plan should:

1. cover the entire target region to within the tolerance ϵ
2. use the least number of shots possible and,
3. confine all of the shots inside the target and without overlap.

Simulated Annealing Simulated annealing has been widely used because of its simplicity as well as the possibility of finding a global solution [4, 49, 54, 65, 71, 77, 94]. We review an example in the paper [71]. At first, the dose distribution is calculated. The optimization technique comprises two steps. First, the continuous parameters (position and weights of shots) are optimized using quasi-Newton method. The result obtained at the end of this step serves as the initial configuration for the next optimization step. Due to the fact that the quasi-Newton method can only be applied to optimize the continuous variables, simulated annealing method is used to optimize the discrete variables such as number of shots and collimator sizes, as well as the continuous variables.

After a set of parameters and a cost function are defined according to the meaning of a “good plan” in real clinical practice, a simulated annealing method is applied as follows:

1. The temperature T is decreased slower than $1/ln(n)$. This ensures that the solution can converge to the global minimum.

2. The isocenter location, shot size and the shot weights are randomly modified with positive or negative step size. Different step sizes are used for different parameters. Assumptions are made so that the collimator size and the shot weights cannot be negative, and the isocenter location has to be within the target.

3. A shot location will be randomly added or removed. By doing this, if the new cost value is reduced, the new configuration is accepted, or accepted with the Metropolis rule. When there is only one shot location, removal cannot occur. By the same token, if there are more shot locations than the prefixed maximum number of shots, additions cannot occur.

4. The optimization process is stopped if the final temperature is reached, or the cost value has not been changed for three different temperatures.

Mixed Integer Programming Mixed Integer Programming (MIP) technique has been presented in [30, 44] to solve the Gamma Knife treatment planning problem. The paper [30] presents an optimization model for solving Gamma Knife radiosurgery treatment planning optimization problem using a large-scale mixed integer programming. They restrict the shot locations to be within the target area. Binary variables are used to indicate if a pair of (shot location, shot size) is used or not. They also impose a constraint to limit the number of shots to use within the given maximum number of shots.

Solving the optimization problem using MIP is very difficult because of its enormous solution space. They propose some heuristics to reduce the size of the solution space. However, because of the large amount of data and many integer variables, the resulting constrained problem is impractical for medium to large size of problems.

Summary

Some techniques have the capability of finding the global optimal solution to our problem: sphere packing (SP), simulated annealing (SA), and mixed integer programming (MIP). However, those methods are generally impractical because of their running time to solve the problem.

The sphere packing method is proven to be NP-hard. In addition, some problems related to this technique are also proven to be NP-hard. A problem with the *Dynamic Programming* scheme discussed in the previous section is that it assumes a fairly regular tumor shape. Another problem is that lots of small shots will be used to fill up the gap between shots. Some other method should be incorporated to optimize a general shape of tumor. Technically, *Simulated annealing* can find the global solution to our problem within some tolerance. However, it is hard to enforce the uniformity constraints. While *MIP* finds a global optimal solution, a major drawback of using MIP is that it generates enormous amount of data for the problem. As a result, it does not find an optimal solution within the time available.

2.4 Optimization Models

Nonlinear optimization techniques have been proposed by several researchers: Nonlinear programming approach [30, 68], Modified Powell's method [70], and a Quasi-Newton method [71].

NLP is a flexible method to formulate the optimization problem. In general, it finds solutions faster than other techniques. A drawback is that the solution to a nonlinear and nonconvex problem is not guaranteed to be globally optimal. In fact, there may be many local minima to our optimization problem. However, with robust modeling techniques and good starting point generation techniques, NLP can be a very powerful approach to our optimization problem. These are the topics in the remainder of Chapters 2 and 3 respectively.

2.4.1 Original formulation

We consider a nonlinear programming approach for solving the treatment planning problem. The input to the nonlinear program consists of several pieces of information, namely the number of shots that are to be used, the widths of shots that are considered appropriate for the target volume, the required isodose level and the target volume itself. The initial locations of the shots are placed randomly within the target, and the initial levels for the exposure time are fixed appropriately.

Given the shot locations and exposure time, the dose distribution for each shot at a given voxel (volume element) on a three dimensional grid is calculated based on the ellipsoidal algebraic model outlined in Section 2.2.2. It is assumed that the dose model does not change due to movement of the shot center.

Once a description of the dose is determined, the optimization model can be formulated. The basic optimization problem is to determine *a set of coordinates* (x_s, y_s, z_s) of shot center locations, *a discrete set of collimator sizes* w, and *radiation exposure times* $t_{s,w}$. The basic variables of the optimization we consider include the coordinates of the center location of the shot (x_s, y_s, z_s), the width of the shot w, and the time $t_{s,w}$ that each shot is exposed. In practice, we consider a grid $\mathcal{G}$ of voxels. There are two types of voxels: $\mathcal{T}$ represents the subset of voxels that are within the target and $\mathcal{N}$ represents the subset of voxels that are out of the target. Since the number of voxels out of the target is vast, we typically use just a small subset of them, generated close to the target volume or in a sensitive structure.

Isodose line coverage. Neurosurgeons commonly use isodose curves as a means of judging the homogeneity of a treatment plan. The 50% isodose curve is a curve that encompasses all of the voxels that receive at least 50% of that maximum dose that is delivered to any voxel in the patient. A treatment plan is normally considered acceptable if a certain percentage isodose curve (typically 50%) encompasses the tumor. We model such a constraint by imposing strict lower and upper bounds on the dose allowed in the target, namely for all $(i, j, k) \in \mathcal{T}$

$$\Theta \leq Dose(i, j, k) \leq 1 \tag{2.5}$$

In this way, the $100\Theta\%$ isodose curve is guaranteed to cover the target. Other isodose curves can be generated by simply modifying the numerical value Θ.

Choosing shot widths. The number of shots to be used is given to the optimization model, and the location of the shot center is chosen by a continuous optimization process. Choosing the particular shot width at each shot location is a discrete optimization problem that is treated by approximating the step function

$$H(t) = \begin{cases} 1 & \text{if } t > 0 \\ 0 & \text{if } t = 0 \end{cases}$$

by a nonlinear function,

$$H(t) \approx H_\alpha(t) := \tfrac{2\arctan(\alpha t)}{\pi}$$

For increasing values of α, H_α becomes a closer approximation to the step function H. This process is typically called smoothing.

The set of shot widths for a given number of shots n is chosen by imposing the constraint:

$$n = \sum_{(s,w)\in\{1,\ldots,n\}\times\mathcal{W}} H_\alpha(t_{s,w}). \tag{2.6}$$

This states that the total number of shot/width combinations that are to be used is n. In practice, we solve a sequence of models, each time increasing the value of α to improve the approximation. Note that the optimization may place two shots of different widths at the same location, and hence none at another location. Typically, we relax the requirement for exactly n shot/widths, and instead impose a range constraint forcing lower and upper bounds on the number of shot/width combinations.

We have tested several optimization formulations. The most obvious model is to minimize the dose outside of the target subject to a constraint on the minimum isodose line that must surround the target:

$$\begin{aligned}
\min \quad & \sum_{(i,j,k)\in\mathcal{N}} Dose(i,j,k) \\
\text{subject to} \quad & Dose(i,j,k) = \sum_{(s,w)\in\mathcal{S}\times\mathcal{W}} t_{s,w} D_w(x_s, y_s, z_s, i, j, k) \\
& \Theta \leq Dose(i,j,k) \leq 1, \quad \forall (i,j,k) \in \mathcal{T} \\
& n = \sum_{(s,w)\in\{1,\ldots,n\}\times\mathcal{W}} H_\alpha(t_{s,w}) \\
& t_{s,w} \geq 0.
\end{aligned} \tag{2.7}$$

The most critical problem is that due to the large number of voxels that are needed when dealing with large irregular tumors (both within and outside of the target) the computational time to complete this treatment plan is too long. To make the solution process faster, we can remove a large number of the non-target voxels from the model. While this improves computational time, this typically weakens the conformity of the dose to the target.

Before we introduce a more practical optimization model we seek to solve (discussed in Section 2.4.3), a conformity estimation model is presented in the next section to estimate an input parameter for the optimization model.

2.4.2 Conformity estimation model

As mentioned in Section 1.1, a conformal solution is one of the requirements of treatment plans. The conformity of the plan is hard to deal with since it involves voxels outside of the target, of which there may be many. Furthermore, a reasonable conformity for a given patient plan is very hard to estimate *a priori* since it depends critically on the number of shots allowed and how the volume of the target interacts with the volumes of the allowable shots.

The conformity index P is an estimate of the ratio of the dose delivered to the target, divided by the total dose delivered to the patient. P can be formally defined as follows:

$$P := \frac{\displaystyle\sum_{\mathcal{T}} \mathcal{D}(\mathcal{T})}{\displaystyle\sum_{\mathcal{T}\cup\mathcal{N}} \mathcal{D}(\mathcal{T}\cup\mathcal{N}).} \tag{2.8}$$

Ideally, we wish to have P to be one, which means all the dose is deposited to the tumor region only. However, this is not possible because radiation is delivered from an external beam source. The radiation passes through some normal tissue before it reaches the tumor. Note that there are standard rules established by various professional and advisory groups that specify acceptable conformity requirements.

We first describe an approach to estimate the value of P. It is known how to simulate the delivery of a shot of width $w \in \mathcal{W}$ centered at the middle of the head of a previously scanned patient on the Gamma Knife. For each shot width we use this to estimate the total dose delivered (at unit intensity) to the complete volume and term this constant $\bar{D}_w$. This is then used to determine an estimate of the total dose delivered to the complete volume by the collection of shots as

$$\sum_{(s,w)\in\mathcal{S}\times\mathcal{W}} \bar{D}_w t_{s,w}, \tag{2.9}$$

without having to calculate the dose at any voxel external to the target. This expression can be used as the denominator of the conformity of a given plan without evaluating dose at voxels outside of the target. The numerator would obviously just be the total dose delivered to the target.

To generate an estimate for P for a particular patient case, We minimize (2.9), subject to the standard constraints of maintaining an appropriate isodose line around the target, and a limit on the number of shots of different widths and locations:

$$\begin{array}{ll} \min & \displaystyle\sum_{(s,w)\in\mathcal{S}\times\mathcal{W}} \bar{D}_w t_{s,w} \\ \text{subject to} & \Theta \le \mathit{Dose}(i,j,k) \le 1, \quad \forall (i,j,k) \in \mathcal{T} \\ & n = \displaystyle\sum_{(s,w)\in\mathcal{S}\times\mathcal{W}} H_\alpha(t_{s,w}) \\ & t_{s,w} \ge 0 \end{array} \tag{2.10}$$

Note that this model uses the data $\bar{D}_w$ instead of calculating the dose outside the target and thus is a much smaller optimization model even if the number of voxels in the complete volume is large. Some care is taken to choose the value of α appropriately. For large treatment volumes we typically only evaluate the bound constraints in (2.10) on a small but representative subset of the voxels in the target. After we solve for (2.10), P is calculated using the following expression:

$$P = \frac{\displaystyle\sum_{(i,j,k)\in\mathcal{T}} \mathit{Dose}(i,j,k)}{\sum_{(s,w)\in\mathcal{S}\times\mathcal{W}} \bar{D}_w t_{s,w}}. \tag{2.11}$$

In (2.10), we attempt to estimate P by minimizing the total dose to the target, subject to hard constraints on the amount of dose delivered at each voxel in the target. However, instead of enforcing these hard constraints, we propose an alternative optimization model as a mechanism to determine P using a notion of *UnderDose.* Underdose can be defined as follows:

$$\mathit{UnderDose}(i,j,k) := \max\{0, \Theta - \mathit{Dose}(i,j,k)\}. \tag{2.12}$$

More formally, a voxel is considered to be underdosed if it receives less than the prescribed isodose, which for the example formulation is assumed to be Θ. We actually use the optimization process to model UnderDose. UnderDose is constrained to be greater than or equal to $\max(0, \Theta - \mathit{Dose})$ at every voxel in the target.

The new conformity estimation model is:

$$\begin{aligned}
\min \quad & \sum_{(s,w) \in \mathcal{S} \times \mathcal{W}} \bar{D}_w t_{s,w} \\
\text{subject to} \quad & \mathit{Dose}(i,j,k) = \sum_{(s,w) \in \mathcal{S} \times \mathcal{W}} t_{s,w} D_w(x_s, y_s, z_s, i, j, k) \\
& \Theta \leq \mathit{UnderDose}(i,j,k) + \mathit{Dose}(i,j,k) \\
& 0 \leq \mathit{UnderDose}(i,j,k) \\
& 0 \leq \mathit{Dose}(i,j,k) \leq 1, \quad \forall (i,j,k) \in \mathcal{T} \\
& \textstyle\sum_{(i,j,k) \in \mathcal{T}} \mathit{UnderDose}(i,j,k) \leq \mathcal{N} \mathcal{P}_{\mathcal{U}} \\
& n = \sum_{(s,w) \in \{1,\ldots,n\} \times \mathcal{W}} H_\alpha(t_{s,w}) \\
& 0 \leq t_{s,w} \leq \bar{t}
\end{aligned} \tag{2.13}$$

The crucial constraint is the one involving both $\mathcal{N}$, the number of voxels in the target, and $\mathcal{P}_{\mathcal{U}}$ a user supplied estimate of the “average percentage” underdose allowable on the target. By increasing the value of $\mathcal{P}_{\mathcal{U}}$, the user is able to relax the homogeneity requirement, thereby reducing the total dose delivered to the patient. Notice that reducing the total dose delivered to the patient typically increases P. Thus, P is essentially a monotone function of $\mathcal{P}_{\mathcal{U}}$. The upper bound on exposure

Table 1: Comparison of conformity estimation models

	Old Conformity Model			New Conformity Model		
Patient	P	obj.val.	time	P	obj.val.	time
Patient 5	0.296 (0.007)	28.89 (13.93)	106.1 (32.9)	0.296 (0.005)	25.68 (12.93)	77.4 (17.3)
Patient 6	0.246 (0.011)	17.81 (14.54)	397.0 (90.5)	0.247 (0.009)	14.89 (13.21)	358.3 (56.2)
Patient 8	0.323 (0.007)	3.33 (2.73)	195.2 (60.8)	0.323 (0.003)	2.86 (1.79)	167.6 (56.3)

time $\bar{t}$ is typically chosen as a large fraction of the maximum dose delivered to $\mathcal{T}$ (here assumed to be 1) for the purposes of improving solver performance.

Table 1 indicates the motivation for this change. For a variety of patients, the estimate of P is essentially the same, but it has smaller standard deviation (indicated in parentheses) and smaller computing times. (For each of the patients, the starting point for the conformity problem was randomly perturbed by up to two voxels in each coordinate direction to generate the sample. The variance is calculated over a set of 30 runs.) Furthermore, it seems clear that the final objective values arising from the subsequent solves are better if these solves are seeded with the new conformity estimation model solutions.

Note that the value of P should be carefully chosen so that the value is fairly insensitive to changes in the starting point given to the model. This is shown below.

The effect of the conformity index value on the optimization objective function In Figure 8, we show how the conformity parameter, P, affects the final solution in a small tumor obtained from a clinician. This tumor contains 4006 voxels in a three-dimensional grid representation. As we increase P, the solution becomes more conformal (but at a cost in homogeneity) that we measure via the objective function in (2.10). The conformity estimation problem generates an average value for P of 0.248, with a standard deviation of 0.012 when we run the previously outlined process 50 times with slightly perturbed starting values. Recall that the planner can specify a scale parameter increase of this value to achieve higher conformity if desired.

2.4.3 The optimization model - Underdose model

The imposition of rigid bounds in the basic model (2.7) leads to plans that are overly homogeneous and not conformal enough, that is, they provide too much dose outside the target. To overcome this, we update the basic model to force more conformity at the expense of relaxing homogeneity. In essence, we interchange the homogeneity constraints and the conformity minimization for a model that controls the conformity of the plan using a constraint and then attempts to minimize the violation of (2.5) in the target. The constraint specifies that at least a portion (P) of the total dose must be deposited in the target:

$$P \leq \frac{\displaystyle\sum_{(i,j,k)\in\mathcal{G}} Dose(i,j,k)}{\displaystyle\sum_{(s,w)\in\mathcal{S}\times\mathcal{W}} \bar{D}_w t_{s,w}}$$

Instead of enforcing the strict lower bound of Θ on the dose in the target, in

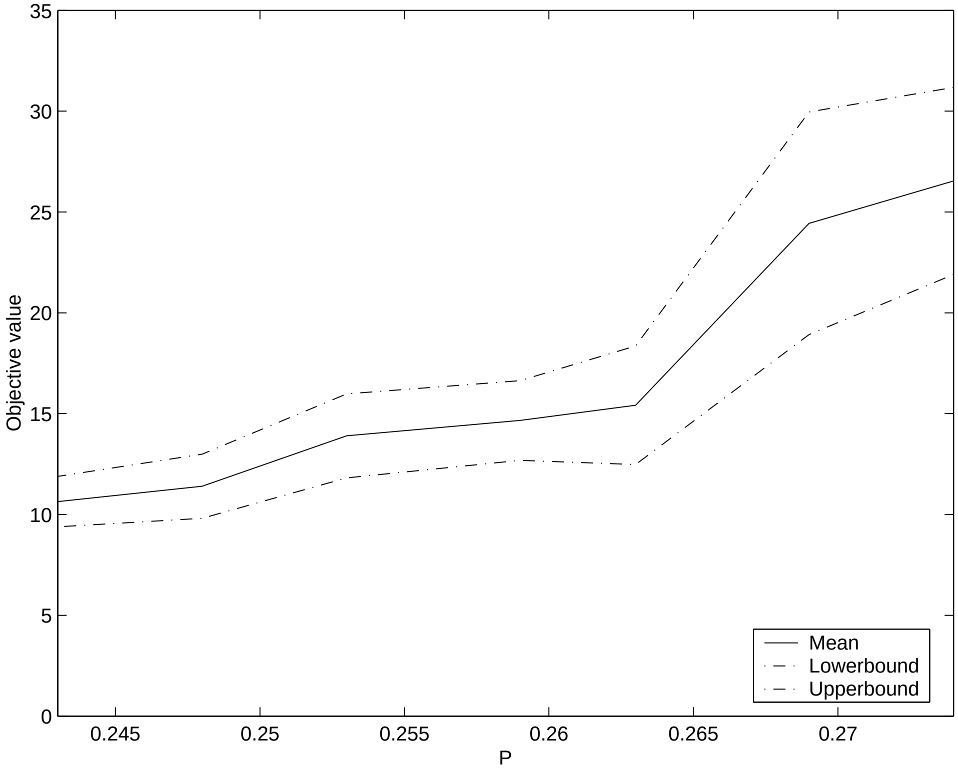


Figure 8: 90% confidence interval for the objective value of (2.10) as a function of conformity value P

the new optimization model, we calculate the amount of dose under this value at every voxel in the target, and sum the "underdose" (2.12) to form our objective. Since we minimize UnderDose, it will take on the maximum of these two values at optimality. An upper bound is still placed on the dose in the target, and the lower bound on dose is relaxed.

The model attempts to minimize the underdose to the target subject to (2.6) and a constraint that the conformity of the plan exceeds a certain (specified) value:

$$
\begin{aligned}
\text{min} \quad & \sum_{(i,j,k)\in\mathcal{G}} UnderDose(i,j,k) \\
\text{subject to} \quad & Dose(i,j,k) = \sum_{(s,w)\in\mathcal{S}\times\mathcal{W}} t_{s,w} D_w(x_s,y_s,z_s,i,j,k) \\
& \Theta \le UnderDose(i,j,k) + Dose(i,j,k) \\
& 0 \le UnderDose(i,j,k) \\
& 0 \le Dose(i,j,k) \le 1, \quad \forall (i,j,k) \in \mathcal{G} \\
& P \le \frac{\displaystyle\sum_{(i,j,k)\in\mathcal{T}} Dose(i,j,k)}{\displaystyle\sum_{(s,w)\in\mathcal{S}\times\mathcal{W}} \bar{D}_w t_{s,w}} \\
& n = \sum_{(s,w)\in\{1,\ldots,n\}\times\mathcal{W}} H_\alpha(t_{s,w}) \\
& 0 \le t_{s,w} \le \bar{t}
\end{aligned}
\tag{2.14}
$$

The constraints involving *UnderDose* coupled with the objective function enforce the definition given in (2.12).

The solution of (2.14) includes non-discrete coordinates of isocenters. This may not be implementable on the Gamma Knife since the continuous values of location coordinates cannot be keyed into the machine. To overcome this, we present a "Fixed Location Model" to translate the optimization output onto the Gamma Knife in the following section.

2.4.4 Fixed location model - mixed integer linear program

We round the location values of the solution and fix them at $\bar{x}_s$, $\bar{y}_s$ and $\bar{z}_s$ respectively. The values of $D_w(\bar{x}_s, \bar{y}_s, \bar{z}_s, i, j, k)$ can then be calculated at each location (i, j, k) as data. The final optimization involves the following mixed integer linear optimization problem:

$$
\begin{aligned}
\text{min} \quad & \sum_{(i,j,k)\in\mathcal{G}} \mathit{UnderDose}(i,j,k) \\
\text{subject to} \quad & \mathit{Dose}(i,j,k) = \sum_{(s,w)\in\mathcal{S}\times\mathcal{W}} t_{s,w} D_w(\bar{x}_s, \bar{y}_s, \bar{z}_s, i, j, k) \\
& \Theta \le \mathit{UnderDose}(i,j,k) + \mathit{Dose}(i,j,k) \\
& 0 \le \mathit{UnderDose}(i,j,k) \\
& 0 \le \mathit{Dose}(i,j,k) \le 1, \quad \forall (i,j,k) \in \mathcal{G} \\
& C\frac{\mathcal{N}_{\mathcal{G}}}{\mathcal{N}} \sum_{(s,w)\in\mathcal{S}\times\mathcal{W}} \bar{D}_w t_{s,w} \le \sum_{(i,j,k)\in\mathcal{G}} \mathit{Dose}(i,j,k) \\
& 0 \le t_{s,w} \le \psi_{s,w}\bar{t} \\
& \textstyle\sum_{(s,w)\in\mathcal{S}\times\mathcal{W}} \psi_{s,w} \le n \\
& \psi_{s,w} \in \{0,1\}
\end{aligned}
\tag{2.15}
$$

The key observation is the use of the binary variable $\psi_{s,w}$ to indicate whether a shot of size w is used at location s. The penultimate constraint in the model ensures that no more than n shots are used, while the upper bound on t ensures that no exposure time occurs if the corresponding shot is not used.

It may of course be possible to extend this model to include more locations, but this was not deemed necessary for our work. Furthermore, it could be argued that the basic model should use integer variables to enforce the discrete size choices. Our investigations found such approaches to be impractical and not as robust as the scheme we outline in the next chapter.

Chapter 3

Enhanced Solution Schemes for Radiosurgery Treatment Planning Optimization Models

3.1 Introduction

The nonlinear programming models discussed in Chapter 2 not only require initial starting solutions, but do not guarantee global optimality. In fact, they may have many local solutions, some of which are close to a global optimal solution, and others may be far off from it. Two techniques are developed in this chapter to enhance the optimization model developed in Chapter 2.

Firstly, an iterative solution scheme for nonlinear program is presented. Since the amount of data used in the optimization is so large, the optimization problem is first solved using uniformly sampled data points to speed up the solution process. In general, the amount of data used in the first optimization process consists of about 13% of the original data. The resulting solution becomes a starting solution for the next optimization process with data points previously ignored.

Secondly, three-dimensional skeleton-based heuristic approach is developed to generate initial starting solution for the isocenters and their corresponding shot sizes. A linear program is then solved to find the initial radiation exposure time by fixing the values of the discrete variables given by the heuristic. The approach leads to improved speed and quality of solutions (see Section 3.4).

The resulting tool is currently in use at the University of Maryland Medical School. The work described here has enabled the simple prototype to be enhanced to the state whereby it is usable without optimization expert intervention as a mechanism to robustly improve the operation of a complex medical system.

3.2 A Solution Scheme of Radiosurgery Treatment Planning Optimization Models

The optimization models considered here are discussed in Chapter 2; namely the "Conformity estimation model" from Section 2.4.2, the "Basic optimization model" of Section 2.4.3, and the "Fixed location model" of Section 2.4.4. Since we solve (2.12) three times, a total of five optimization problems are solved sequentially to determine the treatment plan. The reason the basic model (2.12) is solved iteratively (steps 2, 3, and 4 discussed below) is an effort to reduce the total time to find the solution. Our experience shows that combining three steps into one takes at least three times longer to converge, which is often not clinically acceptable.

Solution Process

1. Conformity estimation. In order to avoid calculating dose outside of the target, we first solve an optimization problem on the target to estimate an

"ideal" conformity for the particular patient for a given number of shots; details can be found in Section 2.4.2. The conformity estimate C is passed to the basic model as an input parameter.

2. Coarse grid estimate. Given the estimate of conformity C, we then specify a series of optimization problems whose purpose is to minimize the total underdose on the target for the given conformity. In order to reduce the computational time required to determine the plan, we first solve (2.14) on a coarse grid subset of the target voxels. We have found it beneficial to use one or two more shot locations in the model than the number requested by the user, that is $\mathcal{S} := \{1, ..., n+2\}$, and allowing the optimization to choose not only useful sizes but also to discard the extraneous shot locations.

3. Refined grid estimate. To keep the number of voxels in the optimization as small as possible, we only add to the coarse grid those voxels on a finer grid for which the homogeneity (bound) constraints are violated. This procedure improves the quality of the plan without greatly increasing the execution time.

   Note that it is possible for the solution from a previous optimization in this sequence to suggest multiple shots to be centered at the same location (i.e. for a given s there are several nonzero $t_{s,w}$). If, in addition, there are other locations s' that are not used at all in the solution at hand, we shift as many of the multiple shots as possible to these unused locations. This maintains the objective value of the current solution while giving any subsequent solves the ability to move the different size shots independently. In our automatic procedure we shift the largest value of $t_{s,w}$ to the unused location.

4. Shot reduction problem. In the solution steps given above, we use a small value of α, typically 6 to impose the constraint (2.6) in an approximate manner. In the fourth solve, we increase the value of α to 100 in an attempt to force the planning system to choose which size/location pairs to use. At the end of this solve, there may still exist some size/location pairs that have very small exposure times t. Also note that our solution technique does not guarantee that the shots are centered at locations within the target.

5. Fixed location model. The computed solution may have more shots used than the user requested and furthermore may not be implementable on the Gamma Knife since the coordinate locations cannot be keyed into the machine. Our approach to refine the optimization solution to generate implementable coordinates for the shot locations is to round the shot location values and then fix them. Once these locations are fixed, the problem becomes linear in the intensity values t. We reoptimize these values and force the user requested number of size/location pairs precisely using a mixed integer program. Further details can be found in Section 2.4.4.

Note that the starting point for each of the models is the solution point of the previous model. Details on how to generate an effective starting point for the first model are given in Section 3.3. All the optimization models are written in the GAMS [13] modeling language and solved using CONOPT [25] or CPLEX [38].

3.3 Starting Point Generation

A good starting point is very important for nonlinear programs, especially if the problem is not convex. This section will explore some techniques to find an initial starting solution for our solution process. The main focus is to find a set of good shot locations and their corresponding sizes. We propose a shot location and size determination (SLSD) process based on 3D medial axis transformation. Our results show that it takes no more than six seconds to produce a good starting solution for all the three-dimensional data considered in our research.

Our targets are collections of three-dimensional voxels. For the large scale problems of interest, the data manipulation and optimization solution times are much larger than allowable (typically 20-40 minutes is allowed for planning) and we must resort to data compression. One technique used extensively in computer vision and pattern recognition is the notion of a skeleton, a series of connected lines providing a simple representation of the object at hand [5, 33, 46, 87, 97]. Skeletons have been used by physicians and scientists to explore virtual human body organs with non-invasive techniques [37, 96]. The term skeleton was proposed in [5] to describe the axis of symmetry, based on the physical analogy of grassfire propagation, namely, the locus of centers of maximal disks (balls) contained in a two- (three-) dimensional shape.

Some applications require that the original object has to be reconstructed from the compact representation, and hence the normal measure of goodness is the error between the original and reconstructed object. However, in our case, we will just use the skeleton to quickly generate good starting shot locations for the nonlinear program. Thus we adapt techniques from the literature to achieve these goals.

Our process is in three stages. First we generate the skeleton, then we place shots and choose their sizes along the skeleton to maximize a measure of our objective. After this, we choose the initial exposure times using a simple linear program. Finally, we apply the five stage optimization process outlined in Section 3.2 to improve upon the starting points found.

3.3.1 Skeleton generation

In this section, we introduce a 3D skeleton algorithm that follows similar procedures to that of [96]. The first step in the skeleton generation is to compute the contour map containing distance information from the voxel to a nearest target boundary. The ideal distance metric is Euclidean, but this is too time consuming to implement in a three-dimensional environment.

To describe our simpler scheme, we first introduce some terminology.

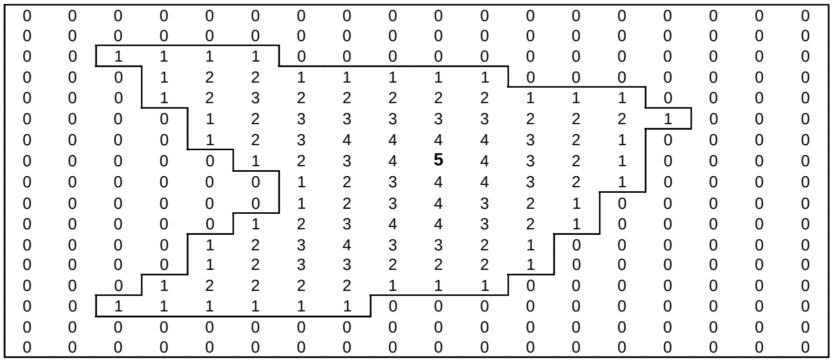
Definition 3.1 *Considering a voxel i as a three-dimensional box, an adjacent voxel j is called an F-neighbor of i if j shares a face with i, an E-neighbor of i if j shares an edge with i and a V-neighbor of i if j shares a vertex with i.*

Our procedure is as follows:

1. Assign 0 to the non-target area, and let $v = 0$.

2. Assign $v + 1$ to any voxel that is unassigned and has an F-neighbor with value v.

3. Increment v by 1 and repeat until all voxels in the target area are assigned.



0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	2	2	1	1	1	1	1	0	0	0	0	0	0	0
0	0	0	1	2	3	2	2	2	2	2	1	1	1	0	0	0	0
0	0	0	0	1	2	3	3	3	3	3	2	2	2	1	0	0	0
0	0	0	0	1	2	3	4	4	4	4	3	2	1	0	0	0	0
0	0	0	0	0	1	2	3	4	**5**	4	3	2	1	0	0	0	0
0	0	0	0	0	0	1	2	3	4	4	3	2	1	0	0	0	0
0	0	0	0	0	0	1	2	3	4	3	2	1	0	0	0	0	0
0	0	0	0	0	1	2	3	4	4	3	2	1	0	0	0	0	0
0	0	0	0	1	2	3	4	3	3	2	1	0	0	0	0	0	0
0	0	0	0	1	2	3	3	2	2	2	1	0	0	0	0	0	0
0	0	0	1	2	2	2	2	1	1	1	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 9: A contour map on a two-dimensional example

An example of a two-dimensional contour map generated is shown in Figure 9.

Note that if the maximum height in the contour map is less than 2, we terminate the skeleton generation process.

Extracting an initial skeleton. Based on the contour map, there are several known skeleton extraction methods in the literature [96]: *Boundary Peeling* (also called thinning) [51], *Distance Coding* (distance transformation) [58] and *Polygon-based Voronoi Methods* [10]. Because it is simple and fast, we use the distance transformation method to generate a skeleton. In our terminology, this means that we define a *skeleton point* as a voxel whose contour map value is greater than or equal to those of its E-neighbors.

Refinement for connectivity of a thin skeleton. We say that two skeleton points are *connected* if they are V-neighbors. Unfortunately, not all the skeleton points generated will be connected, and thus we use a two stage process to connect the pieces of the skeleton together.

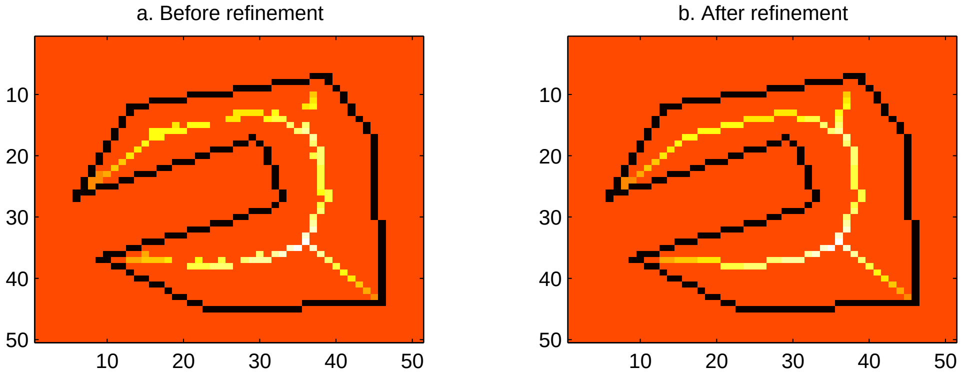


Figure 10: An example of skeleton refinement

For example, Figure 10(a) shows a raw skeleton with several disconnected components. We use two algorithms to join all the disconnected components. The first algorithm is a *directional search* algorithm. The second is the *shortest path* algorithm. After these refinements, we have a connected skeleton as seen in Figure 10(b).

We first use depth first search to label each skeleton point as belonging to a particular component of the skeleton. The first connection phase is a steepest ascent technique. Consider the contour map as a function f. We calculate an approximate gradient ∇f using coordinate-wise central divided differences. Thus, for each voxel (i, j, k), we use the values of f at each of its F-neighbors to generate a three-dimensional vector

$$\begin{aligned}\nabla f(i,j,k) := (&\operatorname{sgn}(f(i+1,j,k) - f(i-1,j,k)),\\ &\operatorname{sgn}(f(i,j+1,k) - f(i,j-1,k)),\\ &\operatorname{sgn}(f(i,j,k+1) - f(i,j,k-1)))\end{aligned}$$

and store these in a divided difference table. Given the voxel (i, j, k), we evaluate f at the V-neighbor $(i, j, k) + \nabla f(i, j, k)$, and accept the move if f does not decrease. We terminate the process if either f decreases or we move to a voxel in a different piece of the skeleton, thus connecting (i, j, k) to this piece. Including the paths generated in this fashion in the skeleton typically connects pieces that are close but not currently connected.

The directional search algorithm, while joining many of the disconnected pieces of the skeleton along ridges of the contour map may fail in cases where the contour map decreases in the gap between two disconnected pieces. Therefore, the second connection phase uses a shortest path algorithm to connect the skeleton (instead of using the *saddle point method* discussed in [96]).

Let $\mathcal{K}$ be the set of all skeletal points, divided into d disconnected components. In order to reduce the search space for the shortest path algorithm, we generate a cloud of voxels $\mathcal{C}$ in the target volume each of which are local maxima among their F-neighbors. Note that $\mathcal{C}$ contains $\mathcal{K}$ by definition, and can be thought of heuristically as a cloud of points encircling the skeleton. We will only join the disconnected components of $\mathcal{K}$ using points in $\mathcal{C}$.

Let each voxel in $\mathcal{C}$ be a node. An arc $(i, j) \in \mathcal{A} \subseteq \mathcal{C} \times \mathcal{C}$ is defined if voxels i and j are V-neighbors.

We choose an arbitrary voxel in an arbitrary component as the source node s. A representative node is chosen from each of the remaining components arbitrarily and joined to a dummy node t that will be the destination. The distance c_{ij} between voxels in a connected cluster is set to 0, whereas other V-neighbors of a given voxel are at distance 1. We attempt to send $d - 1$ units of flow from s to

t. We also add an arc from s to t directly with a high cost to allow for the fact that it may not be possible to join every component through $\mathcal{C}$. If this is the case, it will be signified by flow along these final arcs. The complete formulation of our problem follows.

$$
\begin{array}{ll}
\min & \displaystyle\sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij} \\
\text{subject to} & \displaystyle\sum_{\{j|(i,j)\in\mathcal{A}\}} x_{ij} - \sum_{\{j|(j,i)\in\mathcal{A}\}} x_{ji} = \begin{cases} (d-1) & if \;\; i=s \\ -(d-1) & if \;\; i=t \\ 0 & otherwise \end{cases} \\
 & 0 \le x_{ij}, \quad \forall (i,j) \in \mathcal{A}.
\end{array}
$$

Typically, this problem is solved very quickly by standard linear programming algorithms, even though specialized network flow algorithms could be applied.

3.3.2 Shot placement

At this stage, we recall that our goal is to determine where to place shots and how large to make them initially. The skeleton generation is a data reduction technique to facilitate this goal. We restrict our attention to points on the skeleton. This is reasonable, since the dose delivered (2.3) looks ellipsoidal in nature and hence being centrally located within the target (that is, on the skeleton) is preferable.

Our approach moves along the skeleton evaluating whether the current point is a good location to place a shot. There are two special types of skeleton points, an

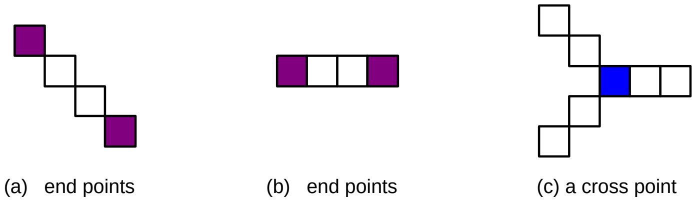


Figure 11: Examples of end-points and a cross-point

end-point and a cross point, that help determining the shot size and the location, see Figure 11.

We define an end-point and a cross point as follows:

Definition 3.2 *A voxel is an* end-point *if*

1. *It is in the skeleton.*

2. *It has only one V-neighbor in the skeleton.*

A voxel is a cross-point *if*

1. *It is in the skeleton.*

2. *It has at least three V-neighbors.*

3. *It is a local maximum in the contour map.*

These points are respectively the start (end-point) and finish (cross-point) points for our heuristic.

Let $\mathcal{K}$ be a set of skeletal points in the target volume. The first phase of the methods determines all end-points in the current skeleton. Given an end-point $(x, y, z) \in \mathcal{K}$, we carry out the following steps to generate a stack for the end-point.

1. Calculate a merit value at the current location. Save the location information, the best shot size, and the merit value on a stack.

2. Find all V-neighbors of the current point, in the skeleton, that are not in the stack. If there is exactly one neighbor, make the neighbor the current location and repeat these two steps. Otherwise, the neighbor is a cross-point or an end-point, and we terminate this process.

If the length of the stack is less than 3, then we discard these points from the skeleton. Otherwise, we choose the shot location and size determined by the smallest merit value on the stack. This shot will cover a subset of the voxels in the target; these voxels are removed from the target at this stage.

We then move to the next end-point and repeat the above process. Once all end-points have been processed, we attempt to generate a new skeleton based on the remaining (uncovered) voxels in the target. We then repeat the whole process with the new skeleton.

The key to this approach is the merit function. Ideally, we would like to place shots that cover the entire region, without overdosing within (or outside) of the target. Overdosing occurs outside the target if we choose a shot size that is too large for the current location, and hence the shot protrudes from the target. Overdosing also occurs within the target if we place two shots too close together for their chosen sizes.

Thus, if we label *height* as the approximate Euclidean distance from the current point to the target boundary, *spread* as the minimum distance between the current location and the end-point at which we started, and w as the shot size, we would like to ensure that all three of these measures are as close as possible. Therefore, we choose an objective function that is a weighted sum of squared differences between these three quantities.

1. $\Phi_{sh}(x, y, z) := (spread(x, y, z) - height(x, y, z))^2$
2. $\Phi_{sw}(x, y, z, w) := (spread(x, y, z) - w)^2$
3. $\Phi_{hw}(x, y, z, w) := (height(x, y, z) - w)^2$

The first function ensures that we pack the target volume as well as possible, that is the current spread between shots should be close to the distance to the closest target boundary. The second function is used to choose a helmet size that fits the skeleton best for the current location. The third function favors a location that is the appropriate distance from the target boundary for the current shot size.

Our objective function Φ is defined as a linear combination (with weights λ) of these penalty functions and a fourth $(\bar{w} - w)^2$, that is designed to favor large shot sizes. Note that $\bar{w}$ is the maximum shot width at hand, typically $18mm$. The weights can be adjusted based on a user's preference. In practice we use 1/3 for the first three objective weights, and 1/2 for the fourth.

3.3.3 Modifying the number of shots used

Often, the application expert knows how many shots will be needed to treat a specific tumor based upon experience. The planning tool accepts this information as input. However, the SLSD procedure only uses target information and it might suggest using fewer or more shots.

If the number of shots generated by SLSD is too large, the first $n+2$ shots are used as the starting point. We allow the nonlinear program to adjust the locations further and remove the least useful shots during the solution process.

If the number of shot locations obtained from the SLSD procedure is less than the requested number, we add extra shot locations using the following (SemiRand) heuristic. The key idea is to spread out the shot center locations with appropriate shot sizes over the target area.

We assume that we are given ρ, an estimate of the conformity that we require from any shot. In practice, we choose this value as 0.2. We then generate k different shot/size combinations as follows. First, a random location s is generated from the target area that is not covered by the current set of shots. Secondly, a random shot size w for the specific location is generated within the set of different shots available $\mathcal{W}$. For each shot/size combination we calculate the fraction $f(s,w)$ of the dose that hits the target by taking the ratio of the number of voxels that it hits in the target to the total number of voxels in a shot of the given size.

We decide the location and size (s,w) to use as follows. If $\max f(s,w) \leq \rho$, then we choose the combination that maximizes $f(s,w)$. Otherwise, amongst all those combinations that are acceptable (i.e. $f(s,w) \geq \rho$), we choose the largest one (i.e. the one that maximizes w among these).

Note that the SemiRand scheme can be used in cases where the SLSD procedure fails (when a 3D volume of the target cannot be defined), and also as an alternative scheme for locating starting points. In practice we use $k=5$.

3.3.4 A linear program to generate initial radiation exposure time

We generate good initial shot center locations and sizes by running SLSD. This is a starting solution for the NLP model except for radiation exposure times. These times $t_{s,w}$ are estimated using the following simple linear program.

$$\begin{aligned} \text{min} \quad & \sum_{(i,j,k)\in\mathcal{G}} UnderDose(i,j,k) \\ \text{subject to} \quad & Dose(i,j,k) = \sum_{(s,w)\in\mathcal{S}\times\mathcal{W}} t_{s,w} D_w(\bar{x}_s,\bar{y}_s,\bar{z}_s,i,j,k) \\ & \Theta \le UnderDose(i,j,k) + Dose(i,j,k) \\ & 0 \le UnderDose(i,j,k) \\ & 0 \le Dose(i,j,k) \le 1, \quad \forall (i,j,k)\in\mathcal{G} \\ & \underline{\mathrm{t}} \le t_{s,w} \le \bar{t} \end{aligned} \tag{3.1}$$

Note that we fix the locations of the shots at the points suggested by SLSD and only update the exposure times. Furthermore, we ensure that every size shot has positive weight in an initial solution by enforcing a lower bound (typically 0.1) on the exposure lengths.

3.4 Computational Results

In this section, we demonstrate how to use the techniques outlined above on two-dimensional testing problems as well as real patient data.

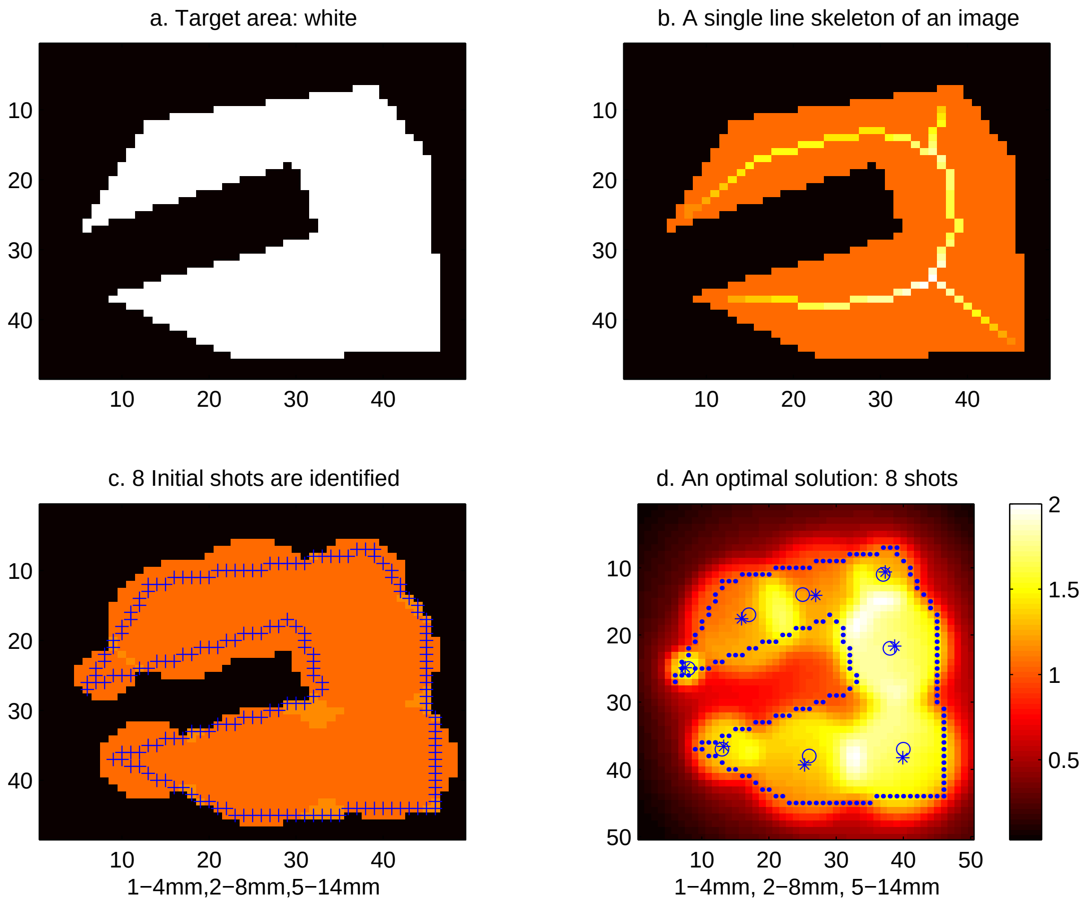


Figure 12: Computational results on a two-dimensional example

3.4.1 Examples on two-dimensional problems

We start with some simple two-dimensional examples that show the types of skeletons that are produced and portray the resulting optimization solutions.

Figure 12(a) depicts a particular target (*tumor*) area for our problem as white space. This tumor is approximately 3 inches square. The shape is not convex. It has a indentation that makes it difficult for a normal optimization model to obtain an acceptable plan. Figure 12(b) shows a thin line skeleton generated from the image. The skeleton generation process takes less than 1 second on Pentium III 800MHz workstation. We then apply the SLSD process to obtain the starting solution for the NLP model as shown in Figure 12(c). Eight shots of radiation are used for this example; one 4 mm, two 8 mm and five 14 mm shots. We use 0.9 as the initial exposure times. The solution covers the target area well. We solve the conformity estimation optimization model using CONOPT2 interface with the starting solution, finding an optimal solution of 8 shots in 61 seconds of execution time. Figure 12(d) shows the resulting plot using MATLAB image toolbox. The circles are the starting solution and the stars are the optimal solution from CONOPT. They are almost identical in shot center locations. The SLSD process outperforms a random starting solution. Given 8 shots to use, the NLP model using a random starting solution finds an optimal solution in 1122 seconds.

We show two more results on other examples in Figure 13. Figure 13(a) is a rectangular shaped target for which three shots are used. The optimization model finds the solution of two $4mm$ and one $14mm$ shots depicted in Figure 13(b). The total time to produce the solution is about 15 seconds. Another example is given in Figure 13(c,d). This is a small tumor (less than 1 in^2) for which three shots are again used. The SLSD model takes 1.5 seconds to generate the starting solution. The NLP model finds an optimal solution of two $4mm$ and one $8mm$ shots in 6 seconds.

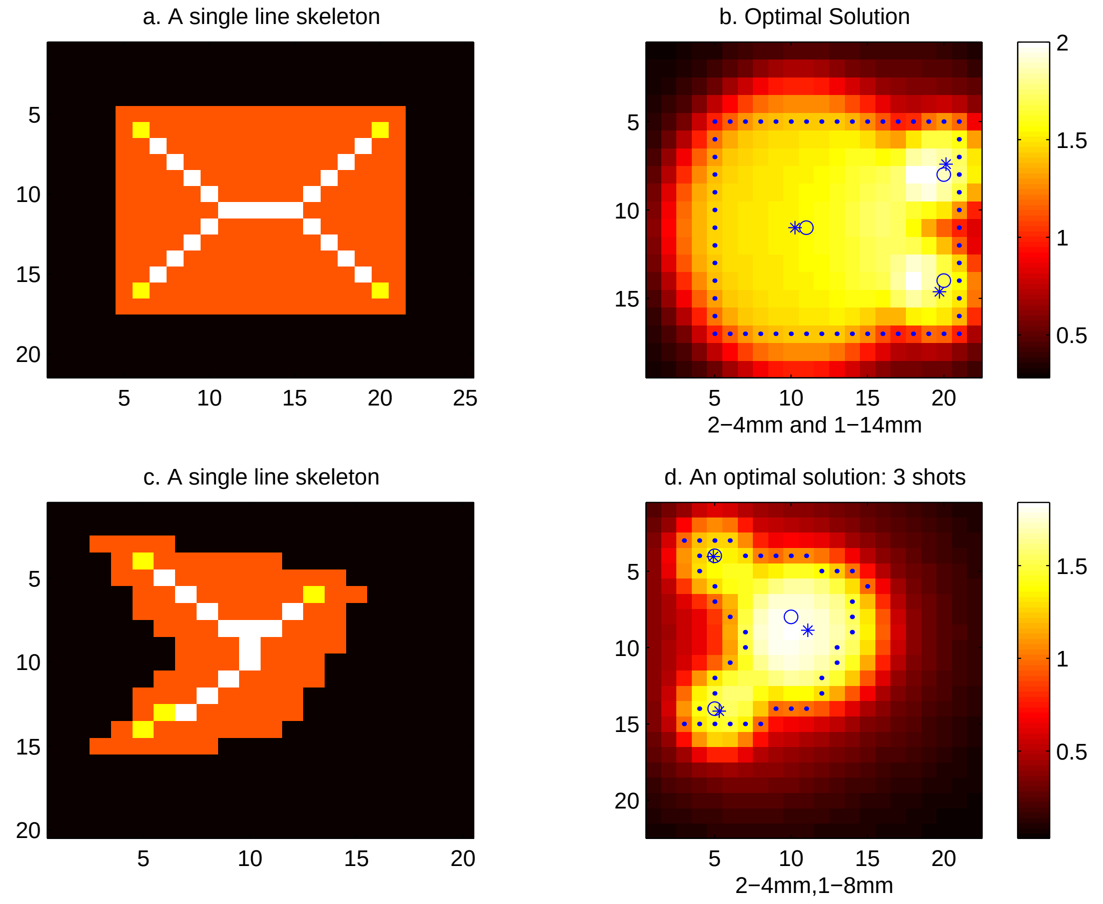


Figure 13: Two-dimensional examples: a rectangular target(a,b) and a small target(c,d)

Table 2: Shot number prediction using SLSD

Helmet Size	Number of Shots suggested
18mm and 14mm	7
18mm and 8mm	7
18mm and 4mm	7
14mm and 8mm	9
14mm and 4mm	10
8mm and 4mm	25

3.4.2 Predicting the number of shots of radiation

We use SLSD to guide the users in selecting the total number of shots of radiation for treatment. Since SLSD produces initial shot locations and their widths based only on the shape of the target, it has a capability of predicting a reasonable number of shots to cover the tumor volume.

First, a user specifies an input on how many different helmet sizes are allowed in the optimization. Using this number, SLSD generates an estimated number of shots for each possible helmet combination. Table 2 shows such an example. The numbers of shots for all possible helmet combinations are displayed using two different helmet sizes are allowed. Since we have four possible helmet sizes available on hand, it generated $6 = \binom{4}{2}$ possible combinations. This is just a suggestion (not the only option) for the user to choose. If the user wants to specify his/her own predicted number, the tool provides an option to do so.

3.4.3 Application to real patient data

We have tested our techniques on ten targets arising from real patient cases. The ten targets are radically different in size and complexity. The tumor volumes range from 28 voxels to 36088 voxels. Since our problems are not convex, the choice of parameters in their solution can also have dramatic effects. In this section, we demonstrate how to choose good parameters for the NLP models. Some further description of the medical implications of these results are given in [68].

The procedure for varying α (controlling the enforcement of the discrete choices) can have dramatic impact on solution quality and times. We generated solutions for a variety of patients under a number of different choices of α. These solutions were analyzed by an application expert. Based on his feedback, we suggest using initial values of α between 4 and 8.

Table 3 shows average objective values of three different starting solution generation techniques: Random, SemiRand, and SLSD. The objective value represents the total average underdose of the target when the solution is applied. The numbers in the parentheses are the standard deviations from a batch of 50 perturbed runs. (In each run, the set of initial solution locations (x, y, z) were perturbed voxel by voxel by a distance of no more than two voxels.) We compare the techniques based on the final objective value and the run time. By fixing $\alpha = 6$, 50 perturbed runs were made for each patient-method pair. In each run, we generated initial locations randomly within the target for the random scheme, while location perturbation was used for SemiRand and SLSD. The tumor was so small for Patient 1 that SLSD failed to generate a skeleton (maximum height in the contour map was less than 2).

Using standard statistical tests, the pairwise p-value [95] between Random and SemiRand was 0.013, between Random and SLSD was 0.0006, and between SemiRand and SLSD was 0.078. This leads to the conclusion that these results are significantly different at the 90% confidence level.

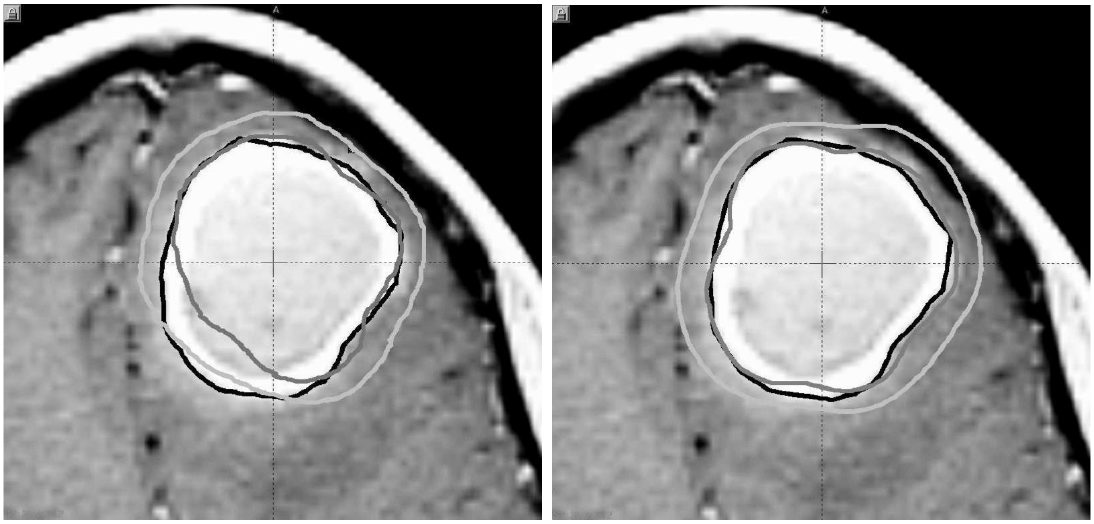
Table 3 also shows average run time of the entire model for the seven different patients. Although a gain of speed using SLSD depends on the shape and size of the tumor, the table shows that the model execution time can be substantially reduced using SLSD over the other two techniques regardless of the size of tumor. Again, these results are significantly different at the 90% confidence level. The pair-wise p-value between Random and SemiRand was 0.017, between Random and SLSD was 0.0006, and between SemiRand and SLSD was 0.063.

Figure 14 shows two pictures of the large tumor solutions that are used by the planners to understand the quality of the solutions. While these figures show the SLSD solution is much more conformal in this slice, and seems much better in quality, it is hard to make a definitive judgment from these figures.

To conclude this section, we show a dose volume histogram (Figure 15) relating various plans that were generated for patient 6. The histogram depicts the fraction of the volume that receives a particular dose for both the skull, and the target volumes. The curves on the right depict information related to the target, while on the left they refer to the skull. On the target, the curves that extend furthest to the right receive more dose. Since this can be effected by just delivering more dose to the patients skull, the lines to the left show that the fraction of the skull receiving a particular dosage is essentially unchanged. The figure compares the

Table 3: Average optimal objective value and solution times in seconds for different tumors

Patient	Objective			Time		
(#voxels)	Random	SemiRand	SLSD	Random	SemiRand	SLSD
1	2.17	0.88	NA	0.3	0.3	NA
(28)	(0.86)	(0.29)	NA	(0.05)	(0.03)	NA
2	14.70	8.21	6.64	32	30	26
(2144)	(6.90)	(4.68)	(2.61)	(6)	(9)	(9)
3	27.53	19.22	14.43	89	67	52
(3279)	(19.07)	(8.87)	(14.99)	(25)	(16)	(9)
4	16.55	12.89	9.85	97	94	84
(3229)	(4.45)	(6.70)	(4.88)	(18)	(22)	(19)
5	34.87	34.53	23.85	153	128	77
(4006)	(16.36)	(17.26)	(13.84)	(40)	(30)	(17)
6	33.32	28.49	15.00	556	513	355
(6940)	(17.25)	(13.09)	(13.22)	(103)	(100)	(52)
7	35.45	29.97	31.03	590	460	343
(10061)	(12.63)	(11.16)	(13.65)	(228)	(100)	(75)
8	9.31	3.22	2.78	887	240	168
(22124)	(2.73)	(2.80)	(1.72)	(157)	(68)	(56)
9	45.05	35.18	31.05	874	629	498
(24839)	(18.10)	(7.11)	(10.25)	(425)	(166)	(99)
10	18.55	11.57	8.59	3568	937	695
(36088)	(11.20)	(11.83)	(6.71)	(589)	(108)	(79)



(a) Random starting point solution. Note that the target and the 50% isodose curve do not match closely.

(b) SLSD starting point solution. Note that the target and the 50% isodose curve match closely.

Figure 14: Large patient example. Three contours drawn represent target, 50% and 30% isodose curves respectively in decreasing greyscale

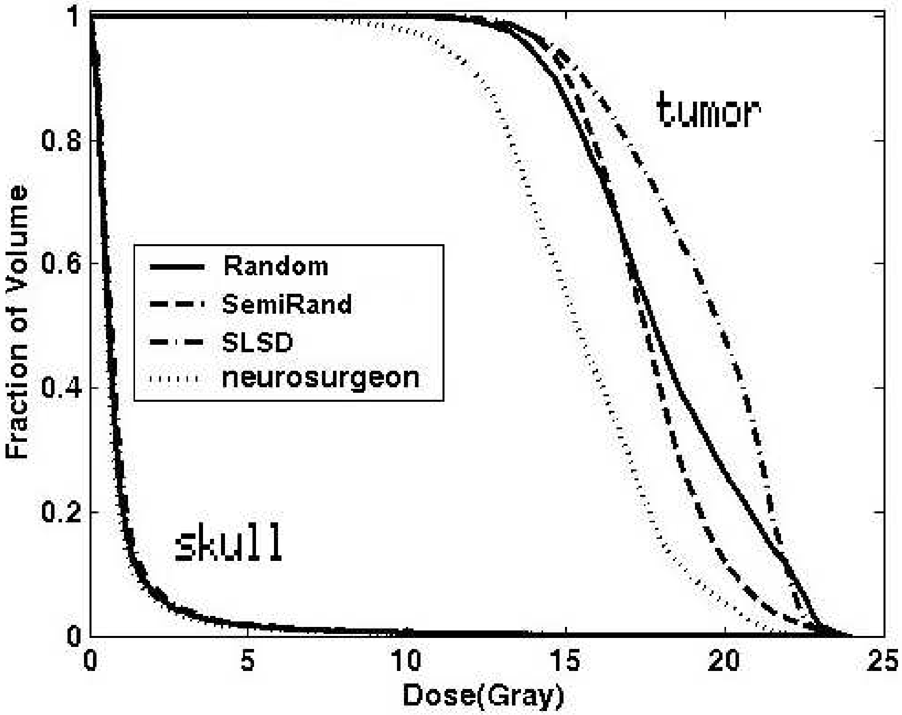


Figure 15: A dose volume histogram for patient 6

three techniques outlined here, along with the actual plan used on the patient example. Clearly, all of the automatic plans are better than the neurosurgeons plan, while the SLSD approach appears preferable to the other two automatic plans in quality.

3.5 Summary

We have used a variety of optimization techniques in this work to develop an approach for solving a planning problem for medical treatment. While our approach has been tailored to the specific application, we believe the methods and approaches used here can be effectively adapted to many other problem classes.

The work described in this chapter was motivated by feedback received from an initial prototype use of our planning tool at the University of Maryland Medical School. The key features that needed improvement were the speed and robustness of the process. This chapter has addressed both issues by using a variety of different optimization models and computational techniques. In particular, the speed of solving the sequence of nonlinear programming models has been substantially reduced by using the skeleton based starting point generation technique. Statistically, we have shown that SLSD outperforms two other heuristics for generating starting points. Furthermore, the use of an improved conformity estimation model, coupled with a "clean-up" mixed integer programming model, ensures the solutions generated are clinically acceptable and conform to the input specifications of the user. The modified tool is now in use at the hospital without intervention from any of the authors.

Chapter 4

Optimization Models for Conformal Radiation Treatment Planning

We introduce a collection of optimization models for three-dimensional conformal radiation therapy. We formulate an optimization problem that simultaneously optimizes the beam configuration and the beam weights as a mixed integer program. Another optimization model includes wedge filters, which are often placed in front of the beam to produce a gradient in the beam intensity across the aperture. We present several techniques to significantly improve solution time of the model without degrading the solution quality. We also demonstrate that the quality of the dose distribution can be improved significantly by incorporating wedge filters into the optimization. Using our algorithms, both the use (or non-use) of a wedge and the wedge orientation are optimized. We present methods to control the *dose volume histogram* on organs implicitly using *hot* and *cold* spot control parameters in the optimization model.

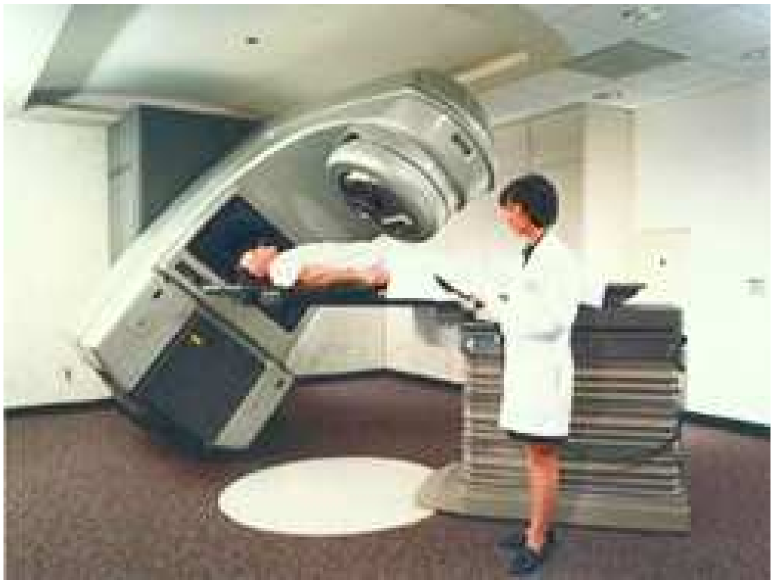


Figure 16: A Linear Accelerator

4.1 Introduction

Radiation treatments are typically delivered using a linear accelerator (see Figure 16) with a multileaf collimator (see Figure 17) housed in the head of the treatment unit. The shape of the aperture through which the beam passes can be varied by moving the computer-controlled leaves of the collimator. In *conformal* radiation therapy, the subject of this chapter, three-dimensional anatomical information is used to shape the beam of radiation at each angle to match the shape of the tumor, as viewed from that angle. We refer to this approach to selecting the beam shape as the *beam's-eye view (BEV)* technique.

The goal in conformal radiation therapy is to provide a high probability of tumor control while minimizing radiation damage to surrounding normal tissue. This goal can accomplished by cross-firing beams from a number of beam directions. In

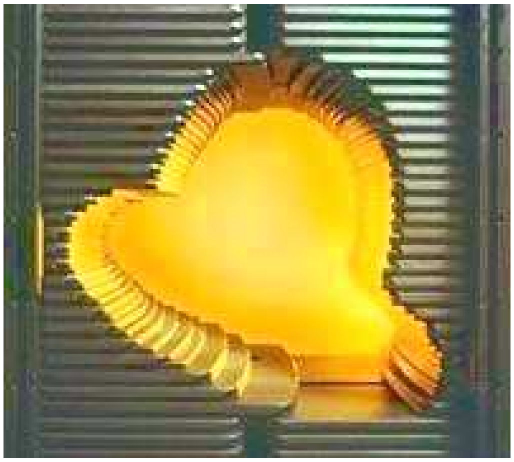


Figure 17: Beam's-eye-view can be produced using a multileaf collimator

practice, a dosimetrist usually uses a trial-and-error approach to determine how many beams of radiation should be used, which beam angles are most effective, and what weight should be assigned to each beam.

Often, additional flexibility is available to the dosimetrist, in the form of wedge filters that can be placed in front of the aperture to induce a gradient in the radiation field from one side of the aperture to the other. Wedge filters are particularly useful in treating cancers that lie near a curved patient surface, as is common in breast cancer. In addition to selecting beam directions and weights, the dosimetrist must decide whether it is appropriate to use a wedge, and if so, which orientation to choose for the wedge. It may be appropriate to use a combination of wedged and non-wedged beams from a single direction.

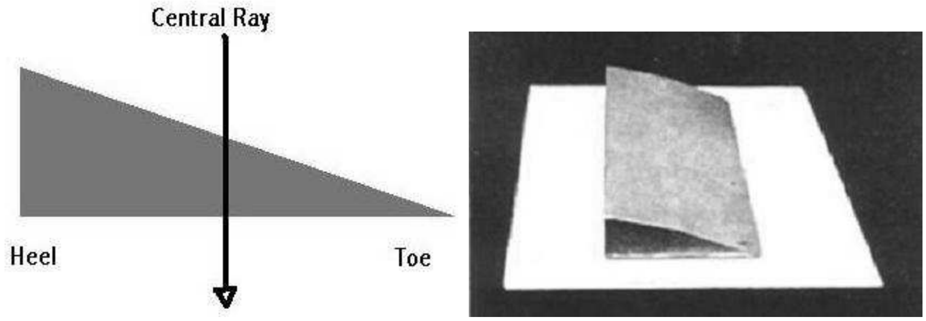
As we show in this thesis, optimization techniques can be used to design these treatment plans automatically. Although the conformal techniques described above are the current standard of care in radiation therapy, used in the treatment of the vast majority of patients today, the benefits of automated treatment planning have gone largely unrealized. We focus on the conformal approach because it requires little alteration to current clinical practices, and therefore has a good chance of rapid adoption. A more sophisticated treatment planning approach known as intensity modulated radiation therapy (IMRT), which is currently receiving a good deal of attention from optimization experts, allows a number of differently shaped beams to be delivered from each direction, thereby allowing a higher degree of flexibility in modulating the intensity of the radiation delivered from each beam angle. Although this approach is undoubtedly interesting, often its nonintuitive choice of aperture shapes represents a significant departure from current clinical practice, and therefore will require more time to be adopted widely.

In Section 4.3, we present several formulations of the treatment planning problem using linear programming (LP), quadratic programming (QP), and mixed-integer programming (MIP) approaches. In these optimizations, each "voxel" within the target volume typically requires at least a specified minimum amount of radiation to be delivered (a lower bound), while an upper bound is used for voxels in the sensitive structures and in the normal tissue. Since sensitive structures often are located close to target volumes, it is sometimes difficult or impossible to determine a treatment plan that satisfies the required bounds at every voxel. Instead, penalty terms can be included in the objective of the optimization problem that penalize violations of these bounds, with more significant violations incurring larger penalties.

Section 4.3.1 describes the problem in which the gantry angles for the treatment plan are fixed, and the task is merely to determine the beam weights for each angle. Several problem LP and QP formulations are presented; we explore the characteristics of each. In Section 4.3.2, we discuss the “angle selection” problem, in which the most effective angles (and their beam weights) are determined from among a set of candidate angles. A MIP model is used here, with binary variables indicating whether or not a particular angle used in the treatment. Treatments with fewer beams can be delivered more rapidly, and hence are generally preferred. We consider treatment plans using wedges in Section 4.3.3, using an extension of the MIP formulation from the angle selection problem. In Section 4.4, we describe several techniques for improving the formulation and reducing the solution time without degrading the solution quality for this model.

The quality of a treatment plan is typically specified and evaluated using a dose-volume histogram (DVH). Using the DVH as a guide, a planner may choose to allow a certain portion of voxels in each sensitive structure to exceed a specified dose, or require a large fraction of the volume to receive at least a certain dose. Due to the need to incorporate many binary variables into the optimization [29], formulation of a constraint of this type is not easy to handle using conventional optimization techniques. In Section 4.5, we show how the MIP formulations can be modified to account for the DVH constraints by using several control parameters.

In Section 4.6, we present computational results for the models described above on clinical data. We demonstrate in particular the usefulness of wedges in devising good treatment plans, and the effectiveness of our techniques for enforcing DVH constraints.



(a) A wedge filter

(b) An external wedge

Figure 18: Wedges

4.2 Use of Wedges

Wedges (also called wedge filters) are a very useful tool in radiation therapy. As shown in Figure 18, a wedge is a tapered metallic block with a thick side (the *heel*) and a thin edge (the *toe*). Less radiation is transmitted through the heel of the wedge than through the toe. Figure 18(b) shows an external 45° wedge, so named because it produces isodose lines that are oriented at approximately 45°, as illustrated in Figure 19. Figure 19(a) shows the dose attenuation pattern produced when no wedge is used, while Figure 19(b) is the dose contour map resulting from the use of a wedge. (In this example, the wedge is oriented with its heel on the right side of the figure.) As well as tilting the isodose lines, the wedge produces a general attenuation of the dose as compared with the open beam.

We include a wedge transmission factor τ in the model to account for the effect of the wedge on the dose delivered to the voxels in the treatment region. Wedges are characterized by two constants τ_0 and τ_1, with $0 \leq \tau_0 < \tau_1 \leq 1$ that indicate

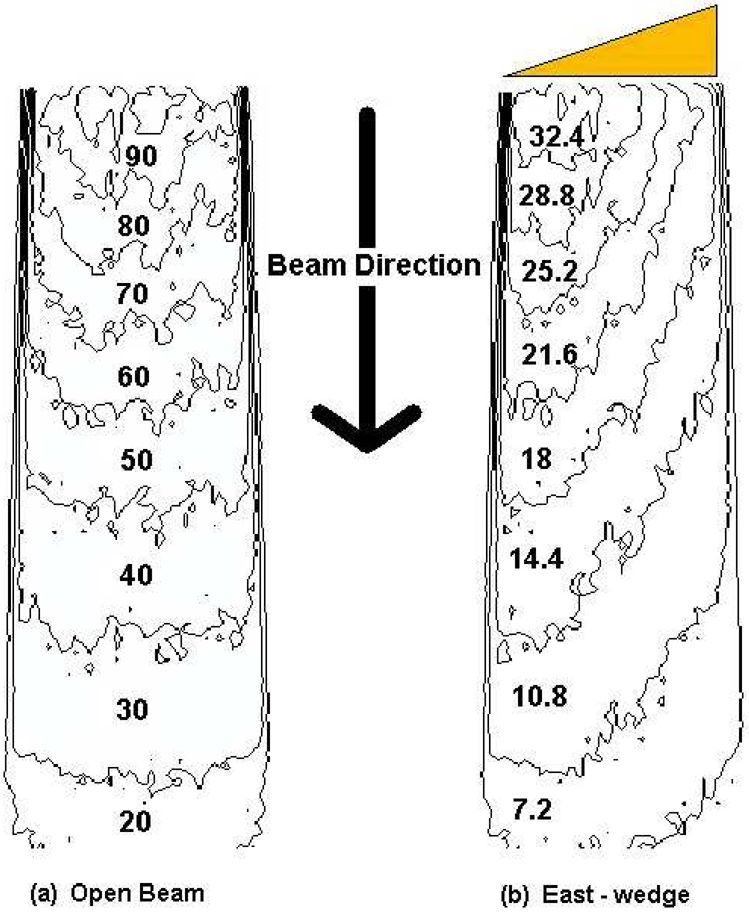


Figure 19: Dose contour maps: wedge effect on the dose distribution

the smallest and largest transmission factors for the wedge among all pencil beams in the field. Specifically, τ_0 indicates the factor by which the dose is decreased for the pencil beams along the edge of the aperture with which the heel of the wedge is aligned. Correspondingly, τ_1 indicates the transmission factor along the opposite (thin) edge. When the heel lies along the west edge, the transmission factor for beamlet (i, j) is calculated as follows:

$$\tau_{ij}^{\text{west}} = \tau_0 + \frac{j - 0.5}{N}(\tau_1 - \tau_0), \quad i = 1, 2, \ldots, M, \;\; j = 1, 2, \ldots, N. \tag{4.1}$$

When the wedge is oriented with its heel at the top, or "north," of the field, the weight applied to the (i, j) beamlet is

$$\tau_{ij}^{\text{north}} = \tau_0 + \frac{i - 0.5}{M}(\tau_1 - \tau_0), \quad i = 1, 2, \ldots, M, \;\; j = 1, 2, \ldots, N. \tag{4.2}$$

The shift of 0.5 is introduced in both formulae to capture the transmission factor at the *center* of each beamlet.

Two different wedge systems are used in clinical practice. In the first system, four different wedges with angles 15°, 30°, 45°, and 60° are available, and the therapist is responsible for selecting one of these wedges and inserting it with the correct orientation. In the second system, a single 60° wedge (the *universal wedge*) is permanently located on a motorized mount located within the head of the treatment unit. This wedge can be rotated to the desired orientation or removed altogether, as required by the treatment plan.

By devising appropriate combinations of wedged and non-wedged beams, we can achieve dose distributions equivalent to those available with a full set of external wedges:

Theorem 4.1 *All plans deliverable by four-wedge system can be produced by using the universal wedge.*

Proof Suppose that at some angle A and some wedge at a given orientation with parameters τ_0' and τ_1' (with $0 \leq \tau_0' < \tau_1' \leq 1$) we have a treatment plan that calls for delivering a weight $w'_{A,\text{open}}$ through the open beam, and $w'_{A,\text{west}}$ through the wedge. (We have supposed without loss of generality that the wedge is oriented to the west, so the attenuation parameter τ_{ij} for beamlet (i, j) is given by the formula (4.1).) We now ask whether it is possible to deliver an equivalent dose through every beamlet using a different wedge with the same (west) orientation, and different parameters τ_0 and τ_1, with $0 \leq \tau_0 < \tau_1 \leq 1$.

Using (4.1), we find that the total dose delivered through beamlet (i, j) is

$$\begin{aligned} & w'_{A,\text{open}} + w'_{A,\text{west}} \left[\tau_0' + \frac{j - 0.5}{N}(\tau_1' - \tau_0')\right] \\ = \quad & w'_{A,\text{open}} + w'_{A,\text{west}} \left[\tau_0' - \frac{0.5}{N}(\tau_1' - \tau_0')\right] + j \cdot w'_{A,\text{west}} \frac{(\tau_1' - \tau_0')}{N}. \end{aligned}$$

If we were to use the alternative wedge with parameters τ_0 and τ_1, and weights $w_{A,\text{open}}$ and $w_{A,\text{west}}$, we would find that the total dose delivered through beamlet (i, j) is

$$w_{A,\text{open}} + w_{A,\text{west}} \left[\tau_0 - \frac{0.5}{N}(\tau_1 - \tau_0)\right] + j \cdot w_{A,\text{west}} \frac{(\tau_1 - \tau_0)}{N}.$$

By equating the constant terms and the coefficient of j in the last two formulae, we find that the plans are equivalent if

$$w_{A,\text{west}}(\tau_1 - \tau_0) = w'_{A,\text{west}}(\tau_1' - \tau_0')$$

and

$$w_{A,\text{open}} + w_{A,\text{west}} \left[\tau_0 - \frac{0.5}{N}(\tau_1 - \tau_0)\right] = w'_{A,\text{open}} + w'_{A,\text{west}} \left[\tau'_0 - \frac{0.5}{N}(\tau'_1 - \tau'_0)\right].$$

By rearranging and substituting, we find that the weights for the new beam must be

$$w_{A,\text{west}} = \frac{\tau'_1 - \tau'_0}{\tau_1 - \tau_0} w'_{A,\text{west}}$$

and

$$w_{A,\text{open}} = w'_{A,\text{open}} + w'_{A,\text{west}} \left[\tau'_0 - \frac{\tau'_1 - \tau'_0}{\tau_1 - \tau_0}\tau_0\right]. \tag{4.3}$$

Note that $w_{A,\text{west}}$ is always nonnegative whenever $w'_{A,\text{west}}$ is nonnegative, but that $w_{A,\text{open}}$ is not necessarily nonnegative, even when the weights for the original wedge are both nonnegative. However, a sufficient condition for $w_{A,\text{open}}$ to be nonnegative for *any* nonnegative values of $w'_{A,\text{open}}$ and $w'_{A,\text{west}}$ is that

$$\frac{\tau'_0}{\tau_0} \geq \frac{\tau'_1 - \tau'_0}{\tau_1 - \tau_0},$$

since this condition ensures that the bracketed term on the right-hand side of (4.3) is nonnegative. This condition implies that given a solution using a particular wedge, we can always identify an equivalent plan using an alternative wedge with the same (or smaller) value of τ_0 and a larger value of $\tau_1 - \tau_0$ □

Hence, in the remainder of this thesis, we consider only approaches based on the universal wedge. Design of treatment plans involving wedges are discussed in [24, 47, 69, 91, 92]. The papers [69, 91, 92] discuss selection of wedge angles; in particular, [69] describes a mathematical basis for selection of wedge angle and orientation. However, the authors of [92] conclude that the inclusion of wedge angle selection in the model makes the optimization problem much harder to solve.

4.3 Formulating the Optimization Problems

4.3.1 Beam weight optimization

We start with the simplest model, in which we assume that the angles from which beams are to be delivered are selected in advance, that wedges are not used, and that the apertures are chosen to be the beam's-eye view from each respective angle. It remains only to determine the intensities of the beams (that is, the beam weights) to be used from each angle.

We now introduce notation that is used below and in later sections. The set of beam angles is denoted by $\mathcal{A}$. We use $\mathcal{T}$ to denote the set of all voxels that make up the target structure, $\mathcal{S}$ to denote the voxels in the sensitive structure, and $\mathcal{N}$ to be the voxels in the normal tissue. We use θ to denote the prescribed dose level for each target voxel, while the hot spot control parameter ϕ defines a dose level for each voxel in the critical structure that we would prefer not to exceed. The beam weight delivered from angle A is denoted by w_A, and the dose contribution to voxel (i,j,k) from a beam of weight 1 from angle A is denoted by $\mathcal{D}_{A,(i,j,k)}$. (It follows that a beam of weight w_A will produce a dose of $w_A\mathcal{D}_{A,(i,j,k)}$ in voxel (i,j,k).) We obtain the total dose $D_{(i,j,k)}$ to voxel (i,j,k) by summing the contributions from all angles $A \in \mathcal{A}$. We use $\mathcal{D}_{A,\Omega}$ to denote the submatrix consisting of the elements $\mathcal{D}_{A,(i,j,k)}$ for all $(i,j,k) \in \Omega$. We use D to denote the vector of doses $D_{(i,j,k)}$ for all voxels (i,j,k), while D_Ω is the vector consisting of $D_{(i,j,k)}$ for all $(i,j,k) \in \Omega$, where Ω is a given set of voxels.

The beam weights w_A, $A \in \mathcal{A}$, which are of course nonnegative, are the unknowns in the optimization problem. The general form of this problem is as follows.

$$\begin{aligned}
&\min_{w} \quad f(D) \\
&\text{s.t.} \\
&\qquad D_\Omega \;=\; \sum_{A\in\mathcal{A}} w_A \cdot \mathcal{D}_{A,\Omega}, \quad \Omega = \mathcal{T} \cup \mathcal{S} \cup \mathcal{N}, \\
&\qquad w_A \;\geq\; 0, \qquad\qquad\qquad\quad \forall A \in \mathcal{A}.
\end{aligned} \tag{4.4}$$

The choice of objective function $f(D)$ in (4.4) depends on the specific goal that the treatment planner wants to achieve. Two common goals are to control the integral dose to organs and to control *cold spots* (underdose to the target region) and *hot spots* (overdose). In general, the objective function measures the mismatch between the prescription and the delivered dose. For voxels in the target region $\mathcal{T}$, there are terms that penalize any difference between the delivered dose and the prescribed dose θ. For voxels in the sensitive structure $\mathcal{S}^i (i = 1, \cdots, |OAR|)$, there are terms that penalize the amount of dose in excess of ϕ_i, the desired upper bound on the dose to such voxels for a sensitive structure i. However, for simplicity of explanation, we only consider a single sensitive structure in the problem formulations in this chapter. For voxels in the normal region $\mathcal{N}$, the desired dose is zero, so the objective usually includes terms that increase as the amount of dose delivered to these voxels increases. There may be more than one sensitive structure in a treatment planning problem.

Let parameters λ_t, λ_s, and λ_n be nonnegative weighting factors applied to the objective terms for the target, sensitive, and normal voxels, respectively. Two common ways to define the objective are to use the L_1-norm (which penalizes the absolute value of deviation from the prescribed dose on each voxel, weighted by the factors just defined) and the sum of squares of the deviations, again weighted according to the region in which each voxel lies. These techniques lead to the following two definitions:

$$\lambda_t \|D_{\mathcal{T}} - \theta e_{\mathcal{T}}\|_1 + \lambda_s \|(D_{\mathcal{S}} - \phi e_{\mathcal{S}})_+\|_1 + \lambda_n \|D_{\mathcal{N}}\|_1, \tag{4.5}$$

$$\lambda_t \|D_{\mathcal{T}} - \theta e_{\mathcal{T}}\|_2^2 + \lambda_s \|(D_{\mathcal{S}} - \phi e_{\mathcal{S}})_+\|_2^2 + \lambda_n \|D_{\mathcal{N}}\|_2^2. \tag{4.6}$$

The notation $(\cdot)_+ := \max(\cdot, 0)$ in the second term defines the overdose to voxels in the sensitive region, while $e_{\mathcal{T}}$ is the vector whose components are all 1 and whose dimension is the same as the cardinality of $\mathcal{T}$. (Similarly for $e_{\mathcal{S}}$.) The terms in (4.5) and (4.5) are approximations to the L_1 and squared-L_2 integrals of the deviations from prescription over each region of interest.

A planner can also use an average dose deviation for each structure by dividing the integral dose over the number of voxels in the structure:

$$\lambda_t \frac{\|D_{\mathcal{T}} - \theta e_{\mathcal{T}}\|_p}{\operatorname{card}(\mathcal{T})} + \lambda_s \frac{\|(D_{\mathcal{S}} - \phi e_{\mathcal{S}})_+\|_p}{\operatorname{card}(\mathcal{S})} + \lambda_n \frac{\|D_{\mathcal{N}}\|_p}{\operatorname{card}(\mathcal{N})}, \quad p = 1, 2,$$

where $\operatorname{card}(\mathcal{T})$, $\operatorname{card}(\mathcal{S})$, and $\operatorname{card}(\mathcal{N})$ denote the number of voxels in the target region, the sensitive structure, and the normal regions, respectively. The use of these factors in the denominator facilitates easier choice of λ_t, λ_s, and λ_n.

An objective function based on L_∞-norm terms (4.7) allows effective penalization of "hot spots" in sensitive regions and of cold spots in the target. We define such a function as follows:

$$\lambda_t \|(D_{\mathcal{T}} - \theta e_{\mathcal{T}})\|_\infty + \lambda_s \|(D_{\mathcal{S}} - \phi e_{\mathcal{S}})_+\|_\infty + \lambda_n \|D_{\mathcal{N}}\|_\infty. \tag{4.7}$$

Combinations of the objective functions above can also be used to achieve specific treatment goals, as we describe later in this section.

Quadratic Programming Formulation

If we use a weighted sum-of-squares objective of the form (4.6), the 3D conformal radiation treatment planning problem is a quadratic program (QP). We slightly modify (4.6) by including the cardinality of the sets $\mathcal{T}$, $\mathcal{S}$, and $\mathcal{N}$ explicitly in the weighting terms. We arrive at the following QP formulation (a particular case of (4.4)):

$$\begin{array}{ll} \min\limits_{w} & \lambda_t \dfrac{\|D_{\mathcal{T}} - \theta e_{\mathcal{T}}\|_2^2}{\text{card}\,(\mathcal{T})} + \lambda_s \dfrac{\|(D_{\mathcal{S}} - \phi e_{\mathcal{S}})_+\|_2^2}{\text{card}\,(\mathcal{S})} + \lambda_n \dfrac{\|D_{\mathcal{N}}\|_2^2}{\text{card}\,(\mathcal{N})} \\ \text{s.t.} & \\ & \begin{array}{rcll} D_\Omega & = & \sum_{A \in \mathcal{A}} w_A \mathcal{D}_{A,\Omega}, & \Omega = \mathcal{T} \cup \mathcal{S} \cup \mathcal{N}, \\ w_A & \geq & 0, & \forall A \in \mathcal{A}. \end{array} \end{array} \tag{4.8}$$

By introducing variables $V_{(i,j,k)}$, $(i,j,k) \in \mathcal{S}$ to denote the excess dose over the upper bound ϕ in the sensitive region $\mathcal{S}$, we can rewrite (4.8) as follows:

$$\begin{array}{ll} \min\limits_{w} & \lambda_t \dfrac{\|D_{\mathcal{T}} - \theta e_{\mathcal{T}}\|_2^2}{\text{card}\,(\mathcal{T})} + \lambda_s \dfrac{\|V_{\mathcal{S}}\|_2^2}{\text{card}\,(\mathcal{S})} + \lambda_n \dfrac{\|D_{\mathcal{N}}\|_2^2}{\text{card}\,(\mathcal{N})} \\ \text{s.t.} & \\ & \begin{array}{rcll} D_\Omega & = & \sum_{A\in\mathcal{A}} w_A \mathcal{D}_{A,\Omega}, & \Omega = \mathcal{T} \cup \mathcal{S} \cup \mathcal{N}, \\ V_{\mathcal{S}} & \geq & D_{\mathcal{S}} - \phi e_{\mathcal{S}}, & \\ V_{\mathcal{S}} & \geq & 0, & \\ w_A & \geq & 0, & \forall A \in \mathcal{A}. \end{array} \end{array} \tag{4.9}$$

Least-Absolute-Value Formulation: Linear Programming

The absolute-value terms in (4.5) do not penalize large violations as much as the squared terms in (4.6). However, they allow the problem to be formulated as a linear program. By including the cardinalities of $\mathcal{T}$, $\mathcal{S}$, and $\mathcal{N}$ in the weighting factors of (4.5), we obtain another special case of (4.4):

$$\begin{array}{ll} \min\limits_{w} & \lambda_t \dfrac{\|D_{\mathcal{T}} - \theta e_{\mathcal{T}}\|_1}{\text{card}\,(\mathcal{T})} + \lambda_s \dfrac{\|(D_{\mathcal{S}} - \phi e_{\mathcal{S}})_+\|_1}{\text{card}\,(\mathcal{S})} + \lambda_n \dfrac{\|D_{\mathcal{N}}\|_1}{\text{card}\,(\mathcal{N})} \\ \text{s.t.} & \\ & \begin{array}{rcll} D_\Omega & = & \sum_{A\in\mathcal{A}} w_A \mathcal{D}_{A,\Omega}, & \Omega = \mathcal{T} \cup \mathcal{S} \cup \mathcal{N}, \\ w_A & \geq & 0, & \forall A \in \mathcal{A}. \end{array} \end{array} \tag{4.10}$$

To recast this problem as a linear program, we introduce variables $V_{(i,j,k)}$ for $(i,j,k) \in \mathcal{T} \cup \mathcal{S}$ to represent violations from the desired doses or dose intervals on the PTV and the OAR. We can then write (4.10) equivalently as follows:

$$\begin{array}{ll}\min\limits_{w} & \lambda_t \dfrac{e_{\mathcal{T}}^T V_{\mathcal{T}}}{\text{card}\,(\mathcal{T})} + \lambda_s \dfrac{e_{\mathcal{S}}^T V_{\mathcal{S}}}{\text{card}\,(\mathcal{S})} + \lambda_n \dfrac{e_{\mathcal{N}}^T D_{\mathcal{N}}}{\text{card}\,(\mathcal{N})} \\ \text{s.t.} & \\ & \begin{array}{rcll} D_\Omega & = & \sum_{A\in\mathcal{A}} w_A \mathcal{D}_{A,\Omega}, & \Omega = \mathcal{T}\cup\mathcal{S}\cup\mathcal{N}, \\ V_{\mathcal{T}} & \geq & D_{\mathcal{T}} - \theta e_{\mathcal{T}}, & \\ V_{\mathcal{T}} & \geq & \theta e_{\mathcal{T}} - D_{\mathcal{T}}, & \\ V_{\mathcal{S}} & \geq & D_{\mathcal{S}} - \phi e_{\mathcal{S}}, & \\ V_{\mathcal{S}} & \geq & 0, & \\ w_A & \geq & 0, & \forall A \in \mathcal{A}. \end{array}\end{array} \tag{4.11}$$

Note that since the elements $\mathcal{D}_{A,(i,j,k)}$ of the dose matrix and w_A of the weight vector are all nonnegative, the elements of the dose vector $D_{\mathcal{N}}$ are also nonnegative. Hence, in the last term of the objective, we are justified in making the substitution $\|D_{\mathcal{N}}\|_1 = e_{\mathcal{N}}^T D_{\mathcal{N}}$.

Min-Max Formulation: Linear Programming

Sometimes, it is important in radiation treatment to minimize the maximum dose violation on organs. Min-max formulations based on (4.7) can be used for this purpose:

$$\begin{array}{lrcll}
\min\limits_{w} & \multicolumn{4}{l}{\lambda_t \| D_{\mathcal{T}} - \theta \|_\infty + \lambda_s \| (D_{\mathcal{S}} - \phi)_+ \|_\infty + \lambda_n \| D_{\mathcal{N}} \|_\infty} \\
\text{s.t.} & & & & \\
& D_\Omega & = & \sum_{A \in \mathcal{A}} w_A \mathcal{D}_{A,\Omega}, & \Omega = \mathcal{T} \cup \mathcal{S} \cup \mathcal{N}, \\
& w_A & \geq & 0, & \forall A \in \mathcal{A}.
\end{array} \tag{4.12}$$

An LP formulation for (4.12) can be generated by introducing extra scalar variables, V_t, V_s, and V_n into the problem as follows.

$$\begin{array}{lrcll}
\min\limits_{w} & \multicolumn{4}{l}{\lambda_t V_t \ + \ \lambda_s V_s \ + \ \lambda_n V_n} \\
\text{s.t.} & & & & \\
& D_\Omega & = & \sum_{A \in \mathcal{A}} w_A \mathcal{D}_{A,\Omega}, & \Omega = \mathcal{T} \cup \mathcal{S} \cup \mathcal{N}, \\
& V_t e_{\mathcal{T}} & \geq & D_{\mathcal{T}} - \theta e_{\mathcal{T}}, & \\
& V_t e_{\mathcal{T}} & \geq & \theta e_{\mathcal{T}} - D_{\mathcal{T}}, & \\
& V_s e_{\mathcal{S}} & \geq & D_{\mathcal{S}} - \phi e_{\mathcal{S}}, & \\
& V_n e_{\mathcal{N}} & \geq & D_{\mathcal{N}}, & \\
& 0 & \leq & V_t, \ V_s, \ V_n, & \\
& 0 & \leq & w_A, & \forall A \in \mathcal{A}.
\end{array} \tag{4.13}$$

Composite Formulations

In the sections above, we introduced three possible problem formulations for the optimization problem (4.4) based on specific treatment goals. Often, the planner's goals are quite specific to the case at hand. For example, the planner may wish to keep the maximum dose violation on the target low, and also to control the integral dose violation on the OAR and the normal tissue. These goals can be met by defining the objective to be a weighted sum of the relevant terms. For the given example, we would obtain the following:

$$\begin{array}{lll} \min\limits_{w} & \lambda_t \|D_{\mathcal{T}} - \theta e_{\mathcal{T}}\|_\infty + \lambda_{\mathcal{S}} \dfrac{\|(D_{\mathcal{S}} - \phi e_{\mathcal{S}})_+\|_1}{\text{card}\,(\mathcal{S})} + \lambda_n \dfrac{\|D_{\mathcal{N}}\|_1}{\text{card}\,(\mathcal{N})} & \\ \text{s.t.} & & \\ & D_\Omega \;=\; \sum_{A\in\mathcal{A}} w_A \mathcal{D}_{A,\Omega}, \quad \Omega = \mathcal{T}\cup\mathcal{S}\cup\mathcal{N}, & \\ & w_A \;\geq\; 0, \qquad\qquad\quad \forall A\in\mathcal{A}. & \end{array} \tag{4.14}$$

Practical objective functions

In practice, voxels on the PTV that receive dose within specified limits may be acceptable as a treatment plan. Furthermore, voxels receive below the lower dose specification (cold spots) may get penalized more severely than *hot spots* on the PTV. Therefore, we consider the following two definitions of $f(D)$ in (4.4) as follows:

$$\begin{aligned} f(D) \;=\; & \lambda_t \frac{\|(D_{\mathcal{T}} - \theta_u e_{\mathcal{T}})_+\|_1 + \|(\theta_L e_{\mathcal{T}} - D_{\mathcal{T}})_+\|_1}{\text{card}\,(\mathcal{T})} \\ & + \lambda_s \frac{\|(D_{\mathcal{S}} - \phi e_{\mathcal{S}})_+\|_1}{\text{card}\,(\mathcal{S})} + \lambda_n \frac{\|D_{\mathcal{N}}\|_1}{\text{card}\,(\mathcal{N})}, \end{aligned} \tag{4.15}$$

$$\begin{aligned} f(D) &= \lambda_t \left(\|(D_{\mathcal{T}} - \theta_u e_{\mathcal{T}})_+\|_\infty + \|(\theta_L e_{\mathcal{T}} - D_{\mathcal{T}})_+\|_\infty \right) \\ & \quad + \lambda_s \frac{\|(D_{\mathcal{S}} - \phi e_{\mathcal{S}})_+\|_1}{\text{card}\,(\mathcal{S})} + \lambda_n \frac{\|D_{\mathcal{N}}\|_1}{\text{card}\,(\mathcal{N})}. \end{aligned} \tag{4.16}$$

In these objectives, θ_L is the target cold-spot control parameter. If the dosage of a voxel in $\mathcal{T}$ falls below a fraction θ_L of θ (assumed to be 1 throughout this chapter), a penalty term for the violation is added to the objective. Likewise, a voxel on the PTV incurs a penalty if the dosage at the voxel exceeds θ_u.

It should be understood that, in all models we describe in this chapter, such a separation of *hot* and *cold* spots is possible. However, we simplify the exposition throughout by using a combined objective function.

Building on the beam-weight optimization formulations described above, we now consider extended models in which beam angles and wedges are included in the optimization problem.

4.3.2 Beam orientation optimization

In the previous section, we showed how to choose the beam weights in an optimal fashion, given a set $\mathcal{A}$ of specified beam orientations. We now consider the problem of selecting a subset of at most K beam angles from a set $\mathcal{A}$ of candidates, simultaneously choosing optimal weights for the selected beams. A treatment plan involving few beams (say, 3 to 5) generally is preferable to one of similar quality that uses more beams because it requires less time and effort to deliver in the clinic.

We give a brief review of literature before the details of the problem formulation are introduced. Some theoretical considerations of optimizing beam orientations are discussed in [8]. In general, using more beams typically produces better quality of treatment plans. The down side, however, is that the time to treat the patients is longer when more beams are used. Furthermore, it has been shown that, when many beams are used, say (≥ 5), beam orientation becomes less important in the overall optimization. [19, 22, 72]. Therefore, the goal here is to find a minimum number of beams that satisfy the treatment goals.

The beam angles and the weights can be optimized either sequentially or simultaneously. Most of the earlier work in the literature uses sequential schemes [16, 35, 56, 63, 64], in which a certain number of beam angles are fixed first, and their weights are subsequently determined. Rowbottom *et al* [62] optimize both variables simultaneously. To reduce the initial search space, a heuristic approach to remove some beam orientations *a priori* is used. They use the simplex method and simulated annealing to solve the overall optimization problem.

A different approach has been proposed by Hass *et al* [36]. They address a geometrical formulation of the coplanar beam orientation problem combined with a hybrid multi-objective genetic algorithm. The approach is demonstrated by optimizing the beam orientation in two dimensions, with the objectives being formulated using planar geometry. Their algorithm attempts to replicate the approach of a treatment planner whilst reducing the amount of computation required. Hybrid genetic search operators have been developed to improve the performance of the genetic algorithm by exploiting problem-specific features. When the approach is applied without constraining the number of beams, the solution produces an indication of the minimum number of beams required.

Problem Formulation We introduce binary variables ψ_A, $A \in \mathcal{A}$, that indicate whether or not angle A is selected to be one of the treatment beam orientations. The value $\psi_A = 0$ indicates that angle A is not used, so the weight for this beam must satisfy $w_A = 0$. When $\psi_A = 1$, on the other hand, the beam from angle A may have a positive weight. Both conditions are enforced by adding the constraint $w_A \leq M \cdot \psi_A$ to the model, where M is a uniform upper bound on the beam weight (discussed below in Section 4.4.1). The resulting mixed programming formulation (4.11) is as follows:

$$\min_{w,\psi} \quad f(D)$$

s.t.

$$\begin{array}{rcll} D_\Omega & = & \displaystyle\sum_{A\in\mathcal{A}} \mathcal{D}_{A,\Omega} \cdot w_A, & \Omega = \{\mathcal{T} \cup \mathcal{S} \cup \mathcal{N}\} \\ 0 & \leq & w_A \leq M\psi_A, & \forall A \in \mathcal{A}, \\ \sum_{A\in\mathcal{A}} \psi_A & \leq & K, & \\ \psi_A & \in & \{0,1\}, & \forall A \in \mathcal{A}. \end{array} \tag{4.17}$$

4.3.3 Wedge orientation optimization

Wedges may be placed in front of a beam to deliver a nonuniform dose distribution across the aperture. Several researchers have studied treatment planning problem with wedge filters [23, 24, 47, 69, 91, 92]. Xing *et al* [91] demonstrate the use of optimizing the beam weights for an open field and two orthogonal wedged fields. Li *et al* [47] presents an optimization algorithm for the wedge orientation selection and the beam weights. Design of treatment plans involving wedges are discussed in [24, 47, 69, 91, 92]. The papers [69, 91, 92] discuss selection of wedge angles; in particular, Sherouse [69] describes a mathematical basis for selection of wedge angle and orientation. However, it has been noted that including wedge angle selection in the optimization makes for excessive computation time [92].

We consider four possible wedge orientations at each beam angle: "north", "south", "east", and "west". At each angle A, we calculate dose matrices for the beams-eye view aperture and for each of these four wedge settings, along with the dose matrix for the no-wedge setting (the open beam), as used in the formulations above. We use $\mathcal{F}$ to denote the set of wedge settings; $\mathcal{F}$ contains 5 elements in our case. Extending our previous notation, the dose contribution to voxel (i, j, k) from a beam delivered from angle A with wedge setting F is denoted by $\mathcal{D}_{A,F,(i,j,k)}$, and we use $\mathcal{D}_{A,F,\Omega}$ to denote the collection of doses for all (i, j, k) in some set Ω. The weight assigned to a beam from angle A with wedge setting F is denoted by $w_{A,F}$, while the binary variable $\pi_{A,F}$ determines whether or not we use a beam from angle A with wedge setting F in the treatment plan.

The optimization problem is to select beams and optimizing weights when wedges are present. The new formulation is obtained not by simply replacing $\mathcal{A}$ by $\mathcal{A}\times\mathcal{F}$ in the discussion above, since there are some additional considerations. First, in selecting beams, we do not wish to place a limit on the total number of beams delivered, as in Section 4.3.2, but rather on the total number of distinct angles used. In other words, we are prepared to allow multiple beams to be delivered from a given angle for the same "cost" as a single beam from that angle; that is,

$$\psi_A \quad \geq \quad \pi_{A,F}, \quad \forall A \in \mathcal{A},\, \forall F \in \mathcal{F}.$$

This constraint models the clinical situation reasonably well, since changing the wedge orientation takes little time relative to the time required to move the gantry and possibly shift the patient.

The second consideration is that we do not wish to deliver two beams from the same angle for two diametrically opposite wedge settings. We enforce this restriction by adding the following constraints:

$$\begin{aligned} 1 &\geq \pi_{A,north} + \pi_{A,south}, \\ 1 &\geq \pi_{A,west} + \pi_{A,east}. \end{aligned} \tag{4.18}$$

(4.18) limits the number of wedge orientations to be less than three in each angle for the treatment. The resulting mixed integer programming model is now as follows:

$$\min_{w,\psi,\pi} \quad f(D)$$

s.t.

$$\begin{array}{rcll}
D_\Omega & = & \displaystyle\sum_{A\in\mathcal{A},F\in\mathcal{F}} w_{A,F}\mathcal{D}_{A,F,\Omega}, & \Omega\in\mathcal{T}\cup\mathcal{S}\cup\mathcal{N}, \\
M\pi_{A,F} & \geq & w_{A,F}, & \\
\psi_A & \geq & \pi_{A,F}, & \\
K & \geq & \sum_{A\in\mathcal{A}}\psi_A, & \\
1 & \geq & \pi_{A,north}+\pi_{A,south}, & \\
1 & \geq & \pi_{A,west}+\pi_{A,east}, & \\
w_{A,F} & \geq & 0, & \forall A\in\mathcal{A},\ \forall F\in\mathcal{F}, \\
\psi_A\ ,\pi_{A,F} & \in & \{0,1\}, & \forall A\in\mathcal{A},\ \forall F\in\mathcal{F}.
\end{array} \tag{4.19}$$

In comparing (4.19) with (4.17), we see that the amount of data to be stored increases by a factor of $|\mathcal{F}|$. The number of binary variables also increases by a factor of $|\mathcal{F}|+1$, although the nature of the new variables $\pi_{A,F}$ and the new constraints is such that the complexity of the problem is increased by less than this factor would suggest. Still, the increase in size and complexity of the integer programming model is nontrivial. As we show in Section 4.6, it is often crucial to use problem reduction to obtain a formulation that can be solved in reasonable time without degrading the solution quality.

The following theorem justifies the use of (4.18):

Theorem 4.2 *A treatment plan calling for two nonzero weights for two diametrically opposite wedge settings can be replaced by an equivalent plan requiring a positive weight for an open beam along with a positive weight for one of the two original beams.*

Proof Consider the "west" and "east" wedge orientations. For beamlet (i, j), $i = 1, 2, \ldots, M$, $j = 1, 2, \ldots, N$, the attenuation factor when the west wedge is present is given by (4.1). For the east wedge, it is as follows.

$$\tau_{ij}^{\text{west}} = \tau_0 + \frac{N - j + 0.5}{N}(\tau_1 - \tau_0), \quad i = 1, 2, \ldots, M, \;\; j = 1, 2, \ldots, N. \qquad (4.20)$$

Suppose now that we have a treatment plan in which, at some angle A, the weight corresponding to the open beam (no wedge) is $w_{A,\text{open}} \geq 0$, while the weights corresponding to the west and east beams are $w_{A,\text{west}} > 0$ and $w_{A,\text{east}} > 0$, respectively. Suppose for the moment that $w_{A,\text{west}} \geq w_{A,\text{east}}$. The contribution of these three weights to the total intensity delivered by beamlet (i, j) is then

$$w_{A,\text{east}} \left[\tau_0 + \frac{N - j + 0.5}{N}(\tau_1 - \tau_0)\right] + w_{A,\text{west}} \left[\tau_0 + \frac{j - 0.5}{N}(\tau_1 - \tau_0)\right] + w_{A,\text{open}}$$

which is equal to

$$(w_{A,\text{west}} - w_{A,\text{east}}) \left[\tau_0 + \frac{j - 0.5}{N}(\tau_1 - \tau_0)\right] + (w_{A,\text{open}} + w_{A,\text{east}}(\tau_1 - \tau_0)).$$

Hence, the same beamlet intensity could be delivered at every (i, j) pair by using weight $w_{A,\text{open}} + w_{A,\text{east}}(\tau_1 - \tau_0)$ for the open beam, $(w_{A,\text{west}} - w_{A,\text{east}})$ for the west wedge, and 0 for the east wedge. Similarly, for the case of $w_{A,\text{west}} \leq w_{A,\text{east}}$, we achieve identical beamlet intensities by using weight $w_{A,\text{open}} + w_{A,\text{west}}(\tau_1 - \tau_0)$ for the open beam, 0 for the west wedge, and $(w_{A,\text{east}} - w_{A,\text{west}})$ for the east wedge.

Therefore when there are positive beam weights for two diametrically opposed wedge orientations, we can obtain an equivalent treatment plan by zeroing the smaller of the two weights, and adjusting the weights on the open beam and on the remaining wedge orientation $\square$

4.3.4 Other solution approaches in the literature

There are many other ways for solving radiation treatment planning optimization problems. We give a brief review of a few approaches frequently appearing in the literature.

Simulated Annealing Webb [81] applies simulated annealing approach on two dimensional problem. First, ideal profiles and relative weights are computed for 8 to 128 beams. Each beam is subdivided into 64 beamlets. A desired dose distribution is obtained by assigning to each pixel within the patient a dose value equal to the prescribed dose value for the organ containing that pixel. Then a cost function to be minimized is defined as the root mean square difference between the computed dose distribution and the desired dose distribution. The ideal beams are *grown* by adding a *grain* of beam weight to a randomly selected beamlet at each iteration. Grain size is kept constant for about the first one million iterations. The size then is decreased linearly until a preset number of iterations are reached. Three-dimensional problems are addressed in [62, 82, 83, 84].

The strength of this approach is that simulated annealing has no limit on the mathematical form of the cost function. The weakness is that the treatment planning time is rather long.

Gradient Projection Method Gradient projection [21, 45] is an iterative technique for improving a previously defined solution. At each iteration, values of the decision variables are adjusted based on the derivatives of the objective function with respect to those variables. Derivatives may be obtained numerically or analytically. The strength of this method is that optimization function can be any form if numerical derivatives are use. The weakness is that it does not guarantee a global optimization. The computation time can be very long also.

Score function approach Score functions [59, 66] are typically nonlinear and nonanalytic functions that assign a single numerical value to a treatment plan. They can be used to evaluate and compare different plans. Score functions are evaluated for all combinations of predefined values of some set of decision variables. The set of values that yields the best score function is selected to be the best plan. The strength of this approach is the generality of the evaluation criteria. But the weakness is that the exhaustive search technique becomes very time consuming as the number of discrete variables increases.

Alternative approaches Alternative approaches [2, 14, 15] to the mathematical techniques have been proposed: AMS (Agmon, Motzkin , Schoenber) and Cimmino algorithms. Their preference for the alternative approach is due to the difficulties inherent in mathematically defining the ideal treatment plan. A feasible solution is obtained rather than an optimum solution to treatment planning.

With an initial solution using 18 or more beams, the number of beams are reduced by eliminating ones with small relative weights. This process is repeated until a feasible solution is found. The Cimmino algorithm has the advantage in that it allows priorities to be assigned to the tumor and normal structures to reflect the relative importance assigned to each by the physician. A strength of the feasible solution search algorithm is that when no solutions exist to the optimization problem, it finds an approximated solution. The weakness is that it ignores an optimal solution.

4.4 Reducing the Solution Time

The formulation (4.19) includes beam angles, weights, and wedges as variables in the formulation. It involves a large amount of data—the beam shapes and dose matrices must be computed for each beam angle and wedge orientation—along with many discrete variables, and so is time-consuming to set up and solve. In this section, we describe a number of techniques for reducing the solution time. First, we show how to choose a reasonable value of M in the formulations (4.19), (4.17). (This choice is important in practice, as an excessively large value of M can lead to a significant increase in run time.) Second, we show how normal-tissue voxels in the treatment region some distance away from the target region can be merged, thereby reducing the number of variables without sacrificing solution quality. Third, we describe a scheme for solving a lower-resolution problem to identify the most promising beam angles, then consider only these angles in solving the full-resolution problem.

4.4.1 Computing tight upper bounds on the beam weights

The formulation (4.19) requires an upper bound M on the beam weights $w_{A,F}$ which is not known *a priori*. If M is too small, the optimization problem can be infeasible or produce a suboptimal result. On the other hand, if the value is too large (usually the case), the algorithm can be considerably slower. A key preprocessing technique to overcome this problem is to use tight bounds on the decision variables [57].

Let ρ_A be the maximum dose deliverable to the target by a beam angle A with a unit beam intensity. Since the open beam delivers more radiation to a voxel (per unit beam weight) than any wedged beam, we have

$$\begin{aligned}\rho_A \;&:= \max_{F\in\mathcal{F},\,(i,j,k)\in\mathcal{T}} \mathcal{D}_{A,F,(i,j,k)} \\ &= \max_{(i,j,k)\in\mathcal{T}} \mathcal{D}_{A,(i,j,k)}, \quad A = 1, 2, \cdots, |\mathcal{A}|,\end{aligned} \tag{4.21}$$

where, as before, $\mathcal{D}_{A,(i,j,k)}$ denotes the dose delivered to voxel (i, j, k) from a unit weight of the open beam at angle A. In Section 4.2, we defined a constant $\tau_1 \in [0, 1]$ as the largest radiation transmission factor by a wedge filter. Using this definition, we have for a given angle A that the maximum dose deliverable to a target voxel using wedge filters is

$$\rho_A \left(w_{A,0} + \tau_1 \sum_{F\in\mathcal{F}\setminus\{0\}} w_{A,F} \right), \tag{4.22}$$

where $0 \in \mathcal{F}$ denotes the open beam.

Suppose now that we modify the model in (4.19) to include explicit control of "hot spots" by introducing an upper bound u on the dose allowed in any target voxel. That is, we assume that the constraint

$$D_{\mathcal{T}} \leq ue_{\mathcal{T}} \tag{4.23}$$

is added to (4.19). (Such a constraint may also be added to the other models of Section 4.3.) By combining (4.23) with (4.22), we deduce that

$$w_{A,0} + \tau_1 \sum_{F \in \mathcal{F} \setminus \{0\}} w_{A,F} \leq \frac{u}{\rho_A}, \ \ \forall A \in \mathcal{A}.$$

We can therefore use this constraint to bound $w_{A,F}$ for $F \in \mathcal{F}$, provided the angle A is selected. If the angle A is not selected, of course, we must enforce $w_{A,F} = 0$ for all $F \in \mathcal{F}$. We can accomplish these goals by replacing the somewhat arbitrary bound in (4.19):

$$M\pi_{A,F} \geq w_{A,F}$$

by

$$w_{A,0} + \tau_1 \sum_{F \in \mathcal{F} \setminus \{0\}} w_{A,F} \leq \left(\frac{u}{\rho_A}\right) \psi_A, \ \ \forall A \in \mathcal{A}, \tag{4.24}$$

where ψ_A is the binary variable that indicates whether or not the angle A is selected. Our modification of (4.19) that includes “hot spot” control and the bound (4.24) is therefore as follows:

$$\min_{w,\psi,\pi} \quad f(D)$$

s.t.

$$\begin{aligned}
D_\Omega &= \sum_{A\in\mathcal{A},F\in\mathcal{F}} w_{A,F}\mathcal{D}_{A,F,\Omega}, && \Omega \in \mathcal{T}\cup\mathcal{S}\cup\mathcal{N}, \\
\psi_A &\geq \pi_{A,F}, \\
\tfrac{u}{\rho_A}\psi_A &\geq w_{A,0} + \tau_1 \sum_{F\in\mathcal{F}\backslash 0} w_{A,F} \\
K &\geq \textstyle\sum_{A\in\mathcal{A}} \psi_A, \\
1 &\geq \pi_{A,north} + \pi_{A,south}, \\
1 &\geq \pi_{A,west} + \pi_{A,east}, \\
w_{A,F} &\geq 0, && \forall A \in \mathcal{A}, \forall F \in \mathcal{F}, \\
\psi_A\ , \pi_{A,F} &\in \{0,1\}, && \forall A \in \mathcal{A}\ , \forall F \in \mathcal{F}.
\end{aligned} \tag{4.25}$$

Note that if we also impose an upper bound on dose level to normal-tissue voxels, we can derive additional bounds on the beam weights using the same approach as is used for the target voxels above.

However, without a constraint on number of wedges being used, we can further simplify (4.25) as follows:

$$
\begin{aligned}
&\min_{w,\psi} \quad f(D)\\
&\text{s.t.}\\
&\qquad \begin{array}{rcll}
D_\Omega &=& \displaystyle\sum_{A\in\mathcal{A},F\in\mathcal{F}} w_{A,F}\mathcal{D}_{A,F,\Omega}, & \Omega\in\mathcal{T}\cup\mathcal{S}\cup\mathcal{N},\\
\frac{u}{\rho_A}\psi_A &\geq& w_{A,0}+\tau_1\displaystyle\sum_{F\in\mathcal{F}\backslash 0} w_{A,F} & \\
K &\geq& \sum_{A\in\mathcal{A}}\psi_A, & \\
w_{A,F} &\geq& 0, & \forall A\in\mathcal{A},\,\forall F\in\mathcal{F},\\
\psi_A &\in& \{0,1\}, & \forall A\in\mathcal{A}\ ,\forall F\in\mathcal{F},
\end{array}
\end{aligned}
\tag{4.26}
$$

Post-processing can be used in the cases where

$$\{w_{A,south}>0\ \text{ and }\ w_{A,north}>0\ \}\ \text{ or }\ \{w_{A,west}>0\ \text{ and }\ w_{A,east}>0\},$$

to avoid a treatment plan that calls for two nonzero weights for two diametrically opposite wedge settings as discussed in Theorem 4.2.

4.4.2 Reducing resolution in the normal tissue

The main focus in solving the optimization problem is to deliver enough dose to the target while avoiding organs at risk as much as possible. Therefore, the dosage to normal regions that are some distance away from the PTV does not need to be resolved to high precision. It suffices to compute the dose only on a representative subset of these normal-region voxels, and use this subset to enforce constraints and to formulate their contribution to the objective.

Given some parameter Δ, we define a neighborhood of the PTV as follows:

$$\mathcal{R}_\Delta(\mathcal{T}) := \{(i,j,k) \in \mathcal{N} \mid \text{dist}\ ((i,j,k),\mathcal{T}) \leq \Delta, \},$$

where dist $((i,j,k),\mathcal{T})$ denotes the Euclidean distance of the center of the voxel (i,j,k) to the target set. We also define a reduced version $\mathcal{N}_1$ of the normal region, consisting only of the voxels (i,j,k) for which i, j, and k are all even; that is:

$$\mathcal{N}_1 := \{(i,j,k) \in \mathcal{N} \mid \quad \bmod (i,2) = \quad \bmod (j,2) = \quad \bmod (k,2) = 0\}.$$

Finally, we include in the optimization problem only those voxels that are close to the target, or that lie in an OAR; or that lie in the reduced normal region. Formally, we consider voxels (i,j,k) with

$$(i,j,k) \in \mathcal{T} \cup \mathcal{S} \cup \mathcal{R}_\Delta(\mathcal{T}) \cup \mathcal{N}_1.$$

Since each of the voxels $(i,j,k) \in \mathcal{N}_1$ effectively represents seven neighboring voxels, the weights applied to the terms for the voxels $(i,j,k) \in \mathcal{N}_1$ in the L_1 and sum-of-squares objective functions ((4.5) and (4.6), respectively) are increased. In effect, the objective quantity $\dfrac{\|D_\mathcal{N}\|_1}{\text{card}\,(\mathcal{N})}$ is smaller than $\dfrac{\|D_{\mathcal{N}_1 \cup \mathcal{R}\Delta(\mathcal{T})}\|_1}{\text{card}\,(\mathcal{N}_1 \cup \mathcal{R}\Delta(\mathcal{T}))}$. If this is an issue, it is possible to replace the latter by

$$\frac{\|\mathcal{D}_{\mathcal{R}\Delta(\mathcal{T})}\|_1 + \|\mathcal{D}_{\mathcal{N}_1}\|_1 \left(\frac{\text{card}(\mathcal{N} \setminus \mathcal{R}\Delta(\mathcal{T}))}{\text{card}(\mathcal{N}_1)}\right)}{\text{card}\,(\mathcal{N})}$$

in the objective function.

4.4.3 A three-phase approach

We now discuss an approach in which rather than attacking the full-scale optimization problem directly, we “ramp up” to the solution via a sequence of models.

Each model in the sequence is easier to solve than the next model, and the solution of each provides a good starting point for the next model. The models differ from each other in the selection of voxels included in the formulation, and in the number of beam angles allowed. The idea is to include only voxels that are significant, in the sense that they affect the solution, and to identify interesting beam angles, discarding those that are unlikely to appear in the solution of the full problem.

One simple approach for removing unpromising beam angles is to remove from consideration those that pass directly through any sensitive structure [62]. A more elaborate approach [59] introduces a score function for each candidate angle, based on the ability of that angle to deliver a high dose to the target without exceeding the prescribed dose tolerance to OAR or to normal tissue located along its path. Only beam angles with the best scores are included in the model.

These heuristics can reduce solution time appreciably, but their effect on the quality of the final solution cannot be determined *a priori*. We propose instead the following incremental modeling scheme, which obtains a near-optimal solution within a small fraction of the time required to solve the original formulation directly. Our scheme proceeds by three phases.

Phase 1: Data Point Reduction. Our aim in this phase is to construct a subset of voxels that are significant for the optimization problem (4.19). A similar technique is applicable to (4.25) and (4.26). Let $\mathcal{S}_1 \subset \mathcal{S}$ be a small subset of voxels of organs at risk, $\mathcal{N}_1 \subset \mathcal{N}$ be the subset of voxels of the normal tissue defined in Section 4.4.2. Note that the way of constructing $\mathcal{S}_1$ is similar to that of $\mathcal{N}_1$. We solve (4.19) with the set $\Omega_1 = \mathcal{T} \cup \mathcal{S}_1 \cup \mathcal{N}_1$ replacing Ω, and a value K_1 replacing the limit K on the number of angles, where $K_1 > K$. The resulting model is smaller than the original formulation (4.19) with many more feasible solutions because we allow more beam angles to be used. It can therefore be solved in a considerably shorter time since it is typically the feasibility enforcement that takes computational time.

Phase 2: Selection of Promising Beam Angles. In the next phase, we augment the set of significant voxels in the OAR. Using the solution w^* of Phase I, dose on the OAR is calculated as follows:

$$D_{\mathcal{S}} = \sum_{A \in \mathcal{A}, F \in \mathcal{F}} w^*_{A,F} \mathcal{D}_{A,F,\mathcal{S}}.$$

Voxels whose dose in the subsequent models is likely to be higher than the hot-spot control parameter ϕ are included by setting

$$\mathcal{S}_2 = \{(i,j,k) \in \mathcal{S} \mid D_{(i,j,k)} \geq \gamma \cdot \phi\},$$

for some parameter $\gamma \in (0, 1]$. (The subset $\mathcal{N}_1$ of normal voxels is not updated at this stage; our experience showed that the effect of augmenting this set was negligible.) We define the set of voxels for Phase 2 as $\Omega_2 = \Omega_1 \cup \mathcal{S}_2$, and choose the number of allowable angles to be K_2, where $K_1 \geq K_2 \geq K$. We now replace Ω by Ω_2 and K by K_2 in (4.19) and re-solve. We denote by $\mathcal{A}_2$ the set of optimal beam angles chosen in the second solve (where $\mathcal{A}_2 \subset \mathcal{A}$).

Phase 3: Final Approximation. In the final phase, we replace Ω by Ω_2 and $\mathcal{A}$ by $\mathcal{A}_2$ in (4.19). We have assumed that in replacing the set $\mathcal{A}$ by the (typically much smaller) set $\mathcal{A}_2$, we have not omitted any angles that would have appeared in the solution of the full-scale model. (If we are correct, we sacrifice nothing in solution quality.) The final approximation typically takes much less time to solve than the full-scale model, both because of the smaller amount of data (due to fewer voxels and fewer beam angles) and fewer binary variables.

We have found that this three-phase scheme reduces the total time required to compute the treatment plan considerably. Although it will not in general produce the same solution as the original full-scale model (4.19), we have found the quality of its approximate solution to be very close to optimal. Computational experience with this approach is given in Section 4.6.

4.5 Techniques for DVH control

Dose-volume histograms (DVH) are a compact way of representing dose distribution information for subsets of the treatment region. By placing simple constraints on the shape of the DVH for a particular region, radiation oncologists can exercise control over fundamental aspects of the treatment plan. For instance, the oncologist often is willing to sacrifice some specified portion of a sensitive structure (such as the lung) in order to provide an adequate probability of tumor control, when the sensitive structure lies near the tumor. This aim can be realized by requiring that at least a specified percentage of the sensitive structure must receive a dose less than a specified level. DVH constraints can also be used to control uniformity of the dose to the target, and to avoid cold spots (regions of underdose). For example, the planner may require all voxels in the target volume to receive doses of between 95% and 107% of the prescribed dose (θ).

DVH constraints that require some fraction of voxels in a region to receive less than a given dose, without specifying which individual voxels must satisfy this requirement, cannot be implemented in a straightforward way using traditional optimization formulations. However, by manipulating the objective function, we can set up and solve a sequence of problems that leads to a satisfactory approximate solution. We describe these techniques with reference to the formulation (4.17). The results are equally valid for (4.19), but the computational requirements are of course higher.

There are three typical requirements for the radiation treatment: *homogeneity*, *conformity*, and *avoidance* as discussed in Chapter 1. In our formulations, homogeneity is controlled by the DVH control parameters θ_L and θ_u ($\theta_L \leq 1 \leq \theta_u$), which specify the lower and upper bounds on the dose to target voxels. (If the prescribed dose to the voxels in $\mathcal{T}$ is θ, then we wish to deliver at least $\theta_L \cdot \theta$ and at most $\theta_u \cdot \theta$ to each voxel.) The conformity constraints, which require that the dose to the normal tissue is as small as possible, can be controlled by the penalty parameter on the normal-tissue voxels in the objective function. As we increase the value of λ_n, it typically reduces the integral dose on the normal tissue. The avoidance constraints, which require the dose to be below certain thresholds on at least some fraction of the sensitive structure, can be implemented by including terms in the objective that involve the OAR voxels and a hot-spot control parameter ϕ.

One might argue that the homogeneity and avoidance requirements can be controlled by adding hard constraints to the optimization model. However, the optimization problem might not be able to find a feasible solution with hard constraints. Even when it is possible to obtain a solution with a hard-constraint formulation, the solutions are typically too homogeneous, and physicians prefer the ability to relax or tighten the constraints using parametrized terms in the objective to achieve a specific treatment goal. We describe ways of controlling DVH on organs, and show results based on a clinical case in the following subsections.

4.5.1 Effects of different objective functions

We introduced different types of objective functions in Section 4.3.1; see in particular (4.5), (4.6), (4.7), (4.15), and (4.16). One can use infinity-norm penalty terms in the objective function to control hot and cold spots in the treatment region, while L_1-norm penalty terms are useful for controlling the integral dose over a region.

Here we illustrate the effectiveness of using *both* types of terms in the objective, by comparing results obtained from an objective with only L_1 terms, with results for an objective with both L_1 and infinity-norm terms. We use the typical values $\theta_L = 0.95$, $\theta_u = 1.07$, $\phi = 0.2$, and $K = 4$ in this experiment. As can be expected, Figure 20 shows that (4.16) has better control on the PTV; the infinity-norm terms yielded a stricter enforcement of the constraints on the PTV. The two objective functions can produce a similar solution if the values of λ_t's are chosen appropriately. However, the choice of such values is not intuitive. We believe that it is easier to choose the value of λ_t for the L_∞ penalty, and use these values in the sequel. We note that on the normal and OAR regions, the difference in quality of the solutions obtained from these two alternative objectives was insignificant.

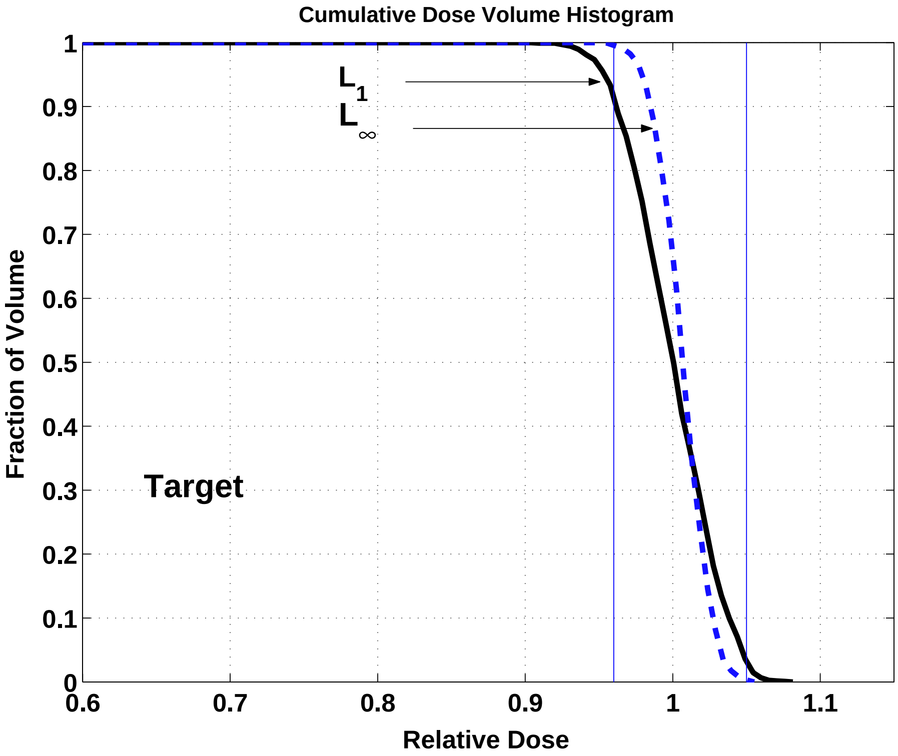


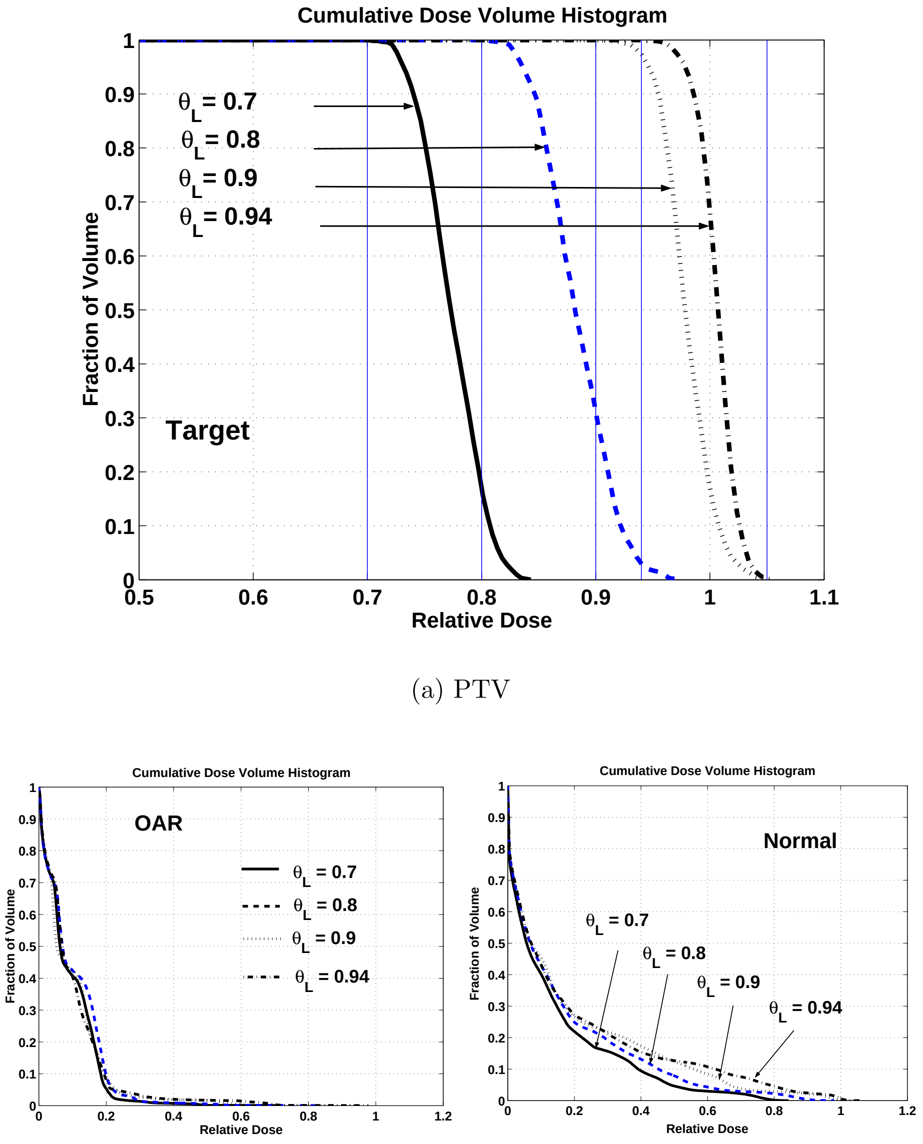
Figure 20: Dose Volume Histogram on the PTV

4.5.2 DVH control on the PTV

Because of our experience reported in Section 4.5.1, we consider the optimization problem (4.17) with objective function $f(D)$ defined by (4.16).

Modelers usually are advised to update the weights $(\lambda_t, \lambda_s, \lambda_n)$ to achieve DVH control on the PTV. However, based on extensive numerical experiments, we believe that this is a less effective way to provide DVH control. We suggest fixing $(\lambda_t, \lambda_s, \lambda_n)$ at appropriate value, say 1, and updating them only for fine tuning of a solution.

Our aim in controlling DVH on the PTV is to attain homogeneity of the dose on $\mathcal{T}$ without significant loss of quality in the dose profile for the normal region and OAR (that is, without significant change to the DVH plots for these regions). As discussed above, the key parameters used to achieve this goal in (4.16) are θ_u and θ_L, which define the desired maximum and minimum fractions of the prescribed dose that the planner wishes to deliver to the target voxels. In this experiment, we fix $\theta_u = 1.07$, and try the values 0.7, 0.8, 0.9, 0.94 for the lower-bound fraction θ_L. Figure 21 shows four DVH plots based on the four different values of θ_L. For each value, we observe that in fact 100% of the target volume receives more that the desired lower bound θ_L. In other words, we manage to avoid completely cold spots in the PTV in this example. We may expect that larger values of θ_L (which lead us to confine the target dose to a tighter range) will result in a less attractive solution in the OAR and the normal tissue. However, as can be seen in Figure 21, the loss of treatment quality is not significant. We conclude that this technique for implementing homogeneity constraints is effective.



(a) PTV

(b) OAR

(c) Normal

Figure 21: DVH plots: DVH control on the PTV

4.5.3 DVH control on the OAR

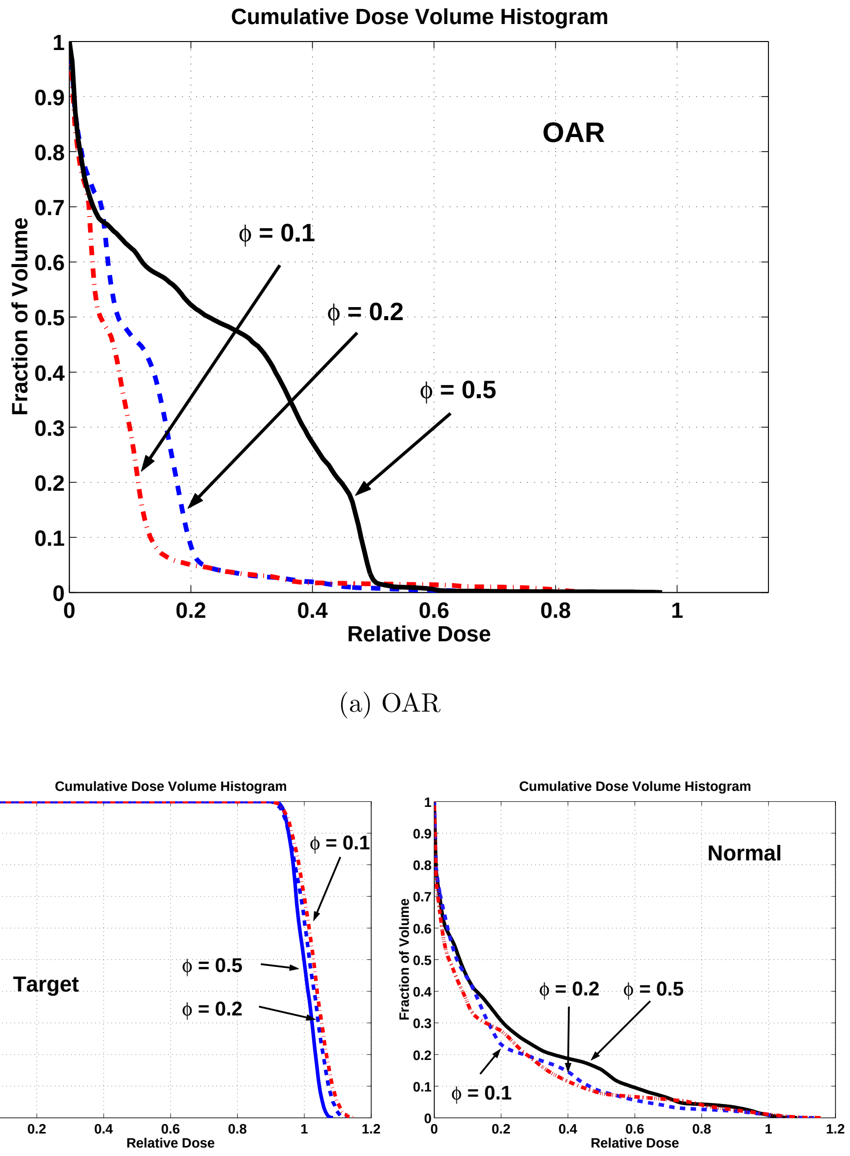
The objective (4.16) also contains terms that penalize the integral of the dose violation over the OAR and normal regions. Here, we show that the dose to the OAR can be controlled by means of the parameter ϕ, assuming that the weights λ_t, λ_s, and λ_n have been fixed appropriately. If our goal is for voxels in the OAR to receive a dose of at most β, where $\beta \in (0, 1)$, we set $\phi = \beta$ in (4.16). Figure 22(a) illustrates the effect of changing values of ϕ on DVH of the OAR. When ϕ is set to 0.5, most of the OAR receives dose less than 50% of the prescribed target dose. Similar results hold for the values 0.2 and 0.1, though constraint is not as "hard" in these cases. (For $\phi = 0.1$, about 20% of the OAR receives more than 10% of the prescribed dose, but only about 5% receives more than 20% of the prescribed dose.) As expected, the costs of achieving better control on the OAR is the loss of treatment quality on the PTV and the normal tissue. However, Figure 22 shows that there is little sacrifice in treatment quality.

4.5.4 Remarks

We conclude this section with several remarks.

1. If our goal is to control hot spots in the OAR rather than the integral dose, we could replace the term $\|(D_{\mathcal{S}} - \phi e_{\mathcal{S}})_+\|_1$ in the objective (4.16) by its infinity-norm analogue $\|(D_{\mathcal{S}} - \phi e_{\mathcal{S}})_+\|_\infty$.

2. In applying the three-phase approach of Section 4.4.3 to the objective function (4.16), we can update ϕ on a per-organ basis and re-solve the optimization problem if the DVH requirement for the OAR is not satisfied at the end



(a) OAR

(b) PTV

(c) Normal

Figure 22: DVH plots: DVH control on the OAR

of Phase 3.

3. There can be some conflict between the goals of controlling DVH on target and non-target regions. Ideally, all target voxels should receive the exact prescription dose θ, while the non-target region should receive zero dose. In practice, this is not possible, as the target is always adjacent either to normal tissue or sensitive structures. Therefore, we need to reach a compromise based on the the relative priorities of meeting the prescription on the target and avoiding excessive dose to the OAR and normal tissues. If the PTV dose control is most important, as is usually the case, the control parameters θ_L, θ_u, ϕ should be chosen with $(\theta_u - \theta_L)$ small and ϕ as a fairly large (but smaller than 1) fraction of θ. However, if the OAR dose control is most important, a smaller value of ϕ can be used in conjunction with L_1-norm penalties for the OAR terms in the objective. In addition, a larger value of $(\theta_u - \theta_L)$ is appropriate in this case.

4.6 Application to Clinical Data

In this section, we use two sets of clinical data to explain how to use our model to achieve treatment planning goals.

4.6.1 Solution time reduction

The specific optimization model considered in this section is as follows:

$$\begin{aligned}
\min_{w,\psi} \quad & \lambda_t \left(\|(D_{\mathcal{T}} - \theta_u e_{\mathcal{T}})_+\|_\infty + \|(\theta_L e_{\mathcal{T}} - D_{\mathcal{T}})_+\|_\infty \right) \\
& +\lambda_s \frac{\|(D_{\mathcal{S}} - \phi e_{\mathcal{S}})_+\|_1}{\text{card}\,(\mathcal{S})} + \lambda_n \frac{\|D_{\mathcal{N}}\|_1}{\text{card}\,(\mathcal{N})}
\end{aligned}$$

s.t.

$$\begin{array}{rcll}
D_\Omega & = & \sum_{A \in \mathcal{A}} \mathcal{D}_{A,\Omega} \cdot w_A, & \Omega = \mathcal{T} \cup \mathcal{S} \cup \mathcal{N}, \\
D_{\mathcal{T}} & \leq & u, & \\
0 & \leq & w_A \leq M\psi_A, & \forall A \in \mathcal{A}, \\
\sum_{A \in \mathcal{A}} \psi_A & \leq & K, & \\
\psi_A & \in & \{0,1\}, & \forall A \in \mathcal{A}.
\end{array} \tag{4.27}$$

Note that we have introduced hard upper bound constraints on the target $D_{\mathcal{T}} \leq u$ (where u typically is somewhat larger than θ_u). We fix some of the control parameters in the optimization model (4.27) throughout the experiments: $\theta_L = 0.95$, $\theta_u = 1.07$, $\phi = 0.2$, $K = 4$, $\lambda_t = \lambda_s = \lambda_n = 1$, $u = 1.15$, $\gamma = 0.95$, and $|\mathcal{A}| = 36$. In fact, the set of angles $\mathcal{A}$ consists of angles equally spaced by 10° in a full 360° circumference.

We attempt to solve (4.27) using the full set of voxels. Note that the optimality criterion is set such that the solution process terminates with the relative error of the objective value being less than or equal to 1%. Figure 23 shows changes of upper and lower bounds of the objective values as the iteration number increases.

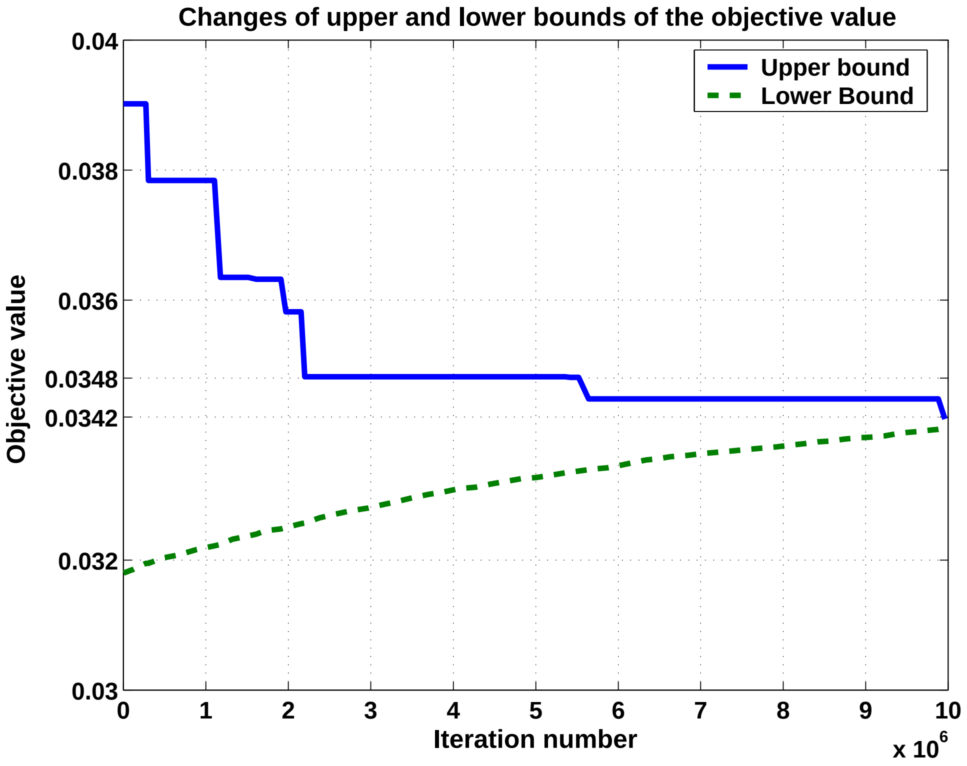


Figure 23: Changes of upper and lower bounds of the objective value

Table 4: Comparisons among different solution schemes.

	I	II	III	IV
Approach	Single Solve	Single Solve	Reduced Model	Three-Phase
Bound (M)	2	u/ρ_A	u/ρ_A	u/ρ_A
Final Objective	0.0342	0.0342	0.0342	0.0348
Time (hours)	112.3	93.5	29.9	3.3
Time saved(%)	-	16.8	73.3	97.0

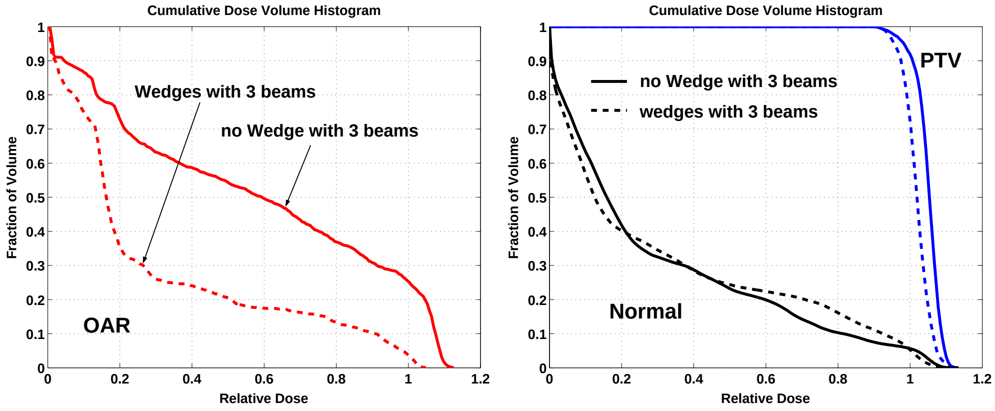
We notice that a large number of iterations are used to slightly improve the feasible solution found at iteration 2.2×10^6. We also notice that the lower bound of the objective value increases slowly. We addressed techniques to overcome these problems in Section 4.4. Effects of using the techniques are discussed in the following paragraph.

Table 4 summarizes results of four different experiments using a data set from a patient with pancreatic cancer. Column I shows the results obtained by solving (4.27) directly, with M set to 2. In column II, we use the tight bound (4.24) on w_A, specialized to the case in which no wedges are used. That is, we replace the constraint $w_A \leq M\psi_A$ in (4.27) by $w_A \leq (u/\rho_A)\psi_A$. (This tighter bound is also used in columns III and IV.) Column III shows the solution time for the reduced-voxel version of the problem discussed in Section 4.4.2. Finally, column IV shows results obtained with the three-phase approach of Section 4.4.3. Here we used parameter values $K_1 = 8$ and $K_2 = 6$, allowing 8 angles to be selected in the first phase and 6 in the second phase. Note that the objective values were calculated on the full set of voxels for the comparison.

Table 4 shows that the final objective values obtained from the first three schemes were the same, to at least three significant digits, while the final objective attained by the three-phase approach was very slightly suboptimal (less than 2% greater). The next rows in Table 4 show the CPU times required (in hours) for each of the four experiments, and the savings in comparison with the time in column I. By comparing columns I and II, we see that a modest reduction was obtained by using the tighter bound. Column III shows that more significant savings were obtained, with essentially no degradation in the quality of the solution plan, by using a reduced model. The full problem contains 1244 voxels in the PTV, 69270 voxels in the OAR, and 747667 voxels in the normal region, while the reduced model has 1244 voxels in the PTV, 14973 voxels in the OAR, and 96154 voxels in the normal tissue. The reduction in computing time was over 73%. Column IV shows that the use of the three-phase scheme resulted in a savings of 97% over the direct solution scheme, again with little effect on the quality of the solution.

Note that, if the solution time is very important, we could relax the cold-spot and hot-spot control parameter values on the PTV. Relaxing these parameter values typically speeds up the the solution time.

We believe our iterative technique is equally effective in the general case in which wedges are included in the formulation. Hence, our subsequent computations used the iterative scheme with wedges.



(a) Organ at Risk

(b) Target & Normal

Figure 24: Dose Volume Histogram: effect of wedges with 3 beam angles

4.6.2 The effect of using wedges on DVH

In general, the use of wedges gives more flexibility in achieving adequate coverage of the tumor volume while sparing normal tissues. To show the effect of wedges, we test our optimization models on a different set of data, from a prostate cancer patient. Figure 24 shows DVH graphs obtained for a treatment plan using wedges (4.25) and one using no wedges (4.27). Three beam angles, $K = 3$, are used in both cases. As can be seen in Figure 24(a), a significant improvement on DVH on the OAR is achieved by adding wedges. In Figure 24(b), we see that there is also a slight improvement in the DVH for the PTV. The line is closer to the prescribed dose level of one when wedges are used. The DVH on the normal tissue, however, does not show much difference between the wedges and no-wedges cases.

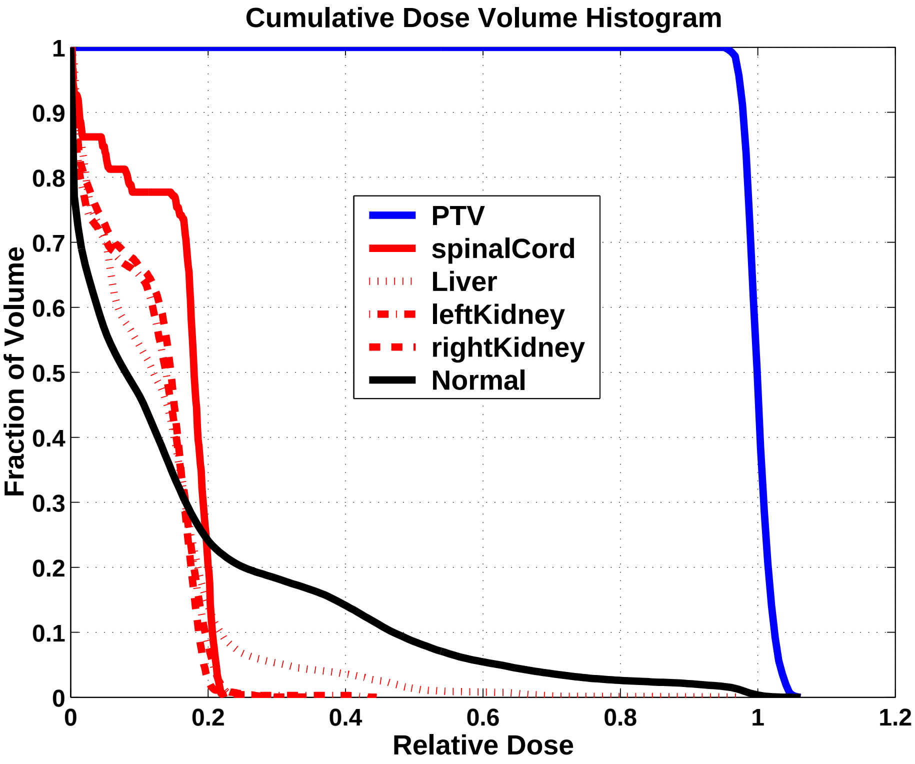


Figure 25: Dose Volume Histogram: optimal solution

4.6.3 A Clinical case study - Pancreas

We now apply the full optimization approach (including DVH controls and wedges) to a pancreatic tumor. This case is made particularly difficult by the close proximity to the PTV of several sensitive structures, including the spinal cord, liver, left kidney, and right kidney. The set $\mathcal{A}$ contains 36 equispaced candidate beam angles. Wedges are also used for the beam angles. The goals of the treatment plan are as follows:

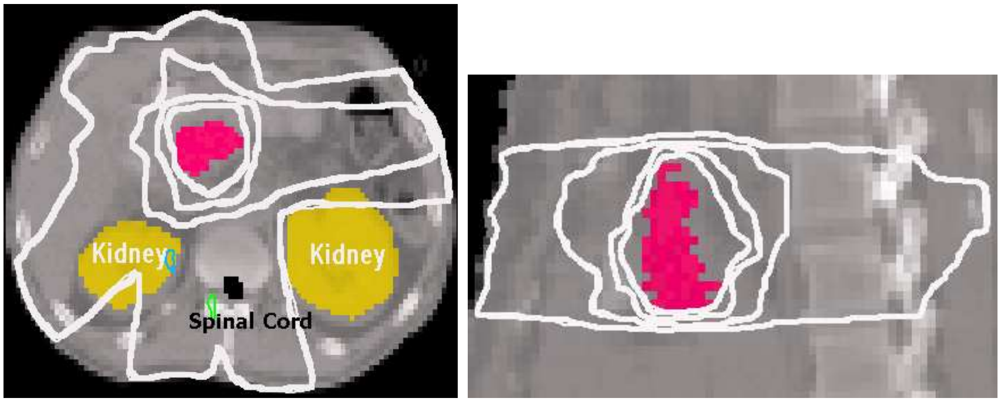
1. Choose four beam angles for the treatment.
2. As the first priority, the target volume should receive dose between 95% and 107% of the presribed dose.
3. 90% of each organ-at-risk should receive less than 20% of the target prescribed dose level.
4. The integral dose delivered to the normal tissue should be minimized.

To achieve these goals, we set DVH control parameters as follows:

$$\theta = 1.0,\ \theta_L = 0.95,\ \theta_u = 1.07,\ K_1 = K_2 = 8,\ K = 4,\ \gamma = 0.95,\ \text{and}$$

$$\phi_i = 0.2,\ i \in \{\text{spinal cord, liver, left kidney, right kidney}\}.$$

Figure 25 shows DVH plots of this experiment. Note first that the homogeneity constraints are satisfied for the PTV: every voxel in the target volume receives 95% and 107% of the prescribed dose. It is also clear that approximately 90% of each sensitive structure receives at most 20% of the target prescribed dose, as specified; the DVH plot for each sensitive structure passes very close to the point $(0.2, 0.1)$ that corresponds to the aforementioned treatment goal.



(a) Axial

(b) Sagittal

Figure 26: Isodose Plots: four lines represent 20%, 50%, 80%, and 95% isodose lines; the 20% line is outermost.

Figure 26 shows isodose lines on the CT slices. The target (tumor) is outlined within four isodose lines. The outermost line is 20% isodose line, which encloses a region in which the voxels receive a dose of 20% of the target prescribed dose level. Moving inwards towards the target, we see 50%, 80%, and 95% isodose lines. Figure 26(a) shows an axial slice. The kidneys are outlined as two circles right below the target. As can be seen, the target lies well inside the 95% isodose line, while the dose to the organs at risk remains reasonable. Figure 26(b) shows a sagittal view of the target with those four iso-dose lines also.

All computations in this chapter were performed on Pentium 4 1.8 GHz machine running on Linux. All optimization problems were modeled in the GAMS modeling language [12]. We use CPLEX 7.1 as LP and MIP solver, and MINOS 5.5 for QP solver.

4.7 Summary

We have developed an optimization framework for 3D conformal radiotherapy. The key features of our methodology are as follows:

1. Simultaneous optimization of three key parameters (beam angles, wedge orientation, and beam weight);
2. Fast delivery of the treatment plan; and
3. Capability of controlling DVH on organs implicitly depending on the specific treatment goal of the planner.

The optimization problems were formulated as mixed integer linear programming and quadratic programming problems. We presented different objective function formulations for different treatment goals. Since the data set required by the obvious optimization formulations was very large, techniques were introduced to reduce the data requirements and the complexity of the problem. Specifically, we introduced tighter *a priori* bounds on the beam weights, reduction of the number of voxels to be considered in the optimization, and a three-phase scheme in which a sequence of progressively more realistic optimization models is solved to obtain an approximate solution. Using all these techniques, we demonstrated a 97% improvement in computational time over direct solution of the full-resolution problem on a clinical data set.

Chapter 5

Optimization Tools and Environments for Radiation Treatment Planning

The optimization of radiation treatment for cancer has become an active research topic in recent years [6, 7, 8, 17, 39, 40, 75, 85, 89]. Many types of cancer are treated by applying radiation from external sources, firing beams into a patient from a number of different angles in such a way that the targeted tumor lies at the intersection of these beams. The increasing sophistication of treatment devices—the aperture through which the beams pass can take on a variety of shapes, multiples apertures can be delivered for each beam angle, and wedges can be used to vary the radiation intensity across the beam—allows delivery of complex and sophisticated treatment plans, achieving a specified dose to the target area while sparing surrounding tissue and nearby critical structures. Optimization techniques are proving to be useful in the design of such plans.

This chapter describes automated treatment planning tools and environments for radiation treatment. The original data for the problem contains the dose distribution information. It consists of the radiation deposited by the beam into each of the small three-dimensional regions ("voxels") into which the treatment area is divided. We divide the beam from each direction into a rectangular array of *pencil beams*, or *beamlets*, calculating the dose matrix independently for each, as described in Section 5.1.2. The beamlet dose matrices are used to identify the BEV, and the aggregate dose matrix for the BEV aperture is obtained by simply adding the contributions from the does matrices for the beamlets that make up the BEV.

The second important component of the data is specification of the tumor region and critical structures. Three-dimensional organ geometries are outlined by a physician on a set of CT or MRI images. The physician labels some of the voxels as PTV (for "Planning Target Volume," the tumor region) and others as OAR (for "Organ At Risk," also known as "sensitive structure" or "critical structure"). Finally, the desired or required dose information for each region is specified by the user.

Optimization software is developed to aid radiation treatment planning as follows. First, a MATLAB routine generates appropriate dose matrices based on the beam's-eye-view approach. A variety of GAMS optimization models for the beam angles, beam weights, and wedge orientations are provided to optimize the treatment plans. Often optimal values of the radiation treatment planning optimization models do not provide sufficient information to judge whether the treatment plans are clinically acceptable or not. Therefore, people in practice rely on other types of measures such as dose volume histogram (DVH) as well as visual aids. A MATLAB routine is provided to examine the quality of treatment plans.

Some optimization modelers may have interest in creating unique shapes of organs to tune their models. But, it is often not easy to obtain such organ structures. To meet this need, we provide a MATLAB routine to create simulated organ structures.

5.1 Beam Aperture Generation and Multileaf Collimator Specification

5.1.1 A literature review on beam aperture generation

Beam's-eye-view (BEV) has been widely used in computerized radiation treatment planning [11, 16, 19, 34, 52, 56]. Goitein [34] presents a three-dimensional treatment planning program using BEV. The paper [52] integrates the BEV into computerized treatment planning. Their contribution enables beam's-eye-view graphics to be mixed with gray-scale images such as simulator and verification radiographs, and digital reconstructed radiographs. This is an early work where BEV is used to calculate three-dimensional dose distribution. To speed up the generation of beam aperture, Brewster *et al* [11] present a method that generates beam aperture for computer-aided optimization of radiation therapy. The notion of target-eye-view (TEV) map is discussed in [19]. In TEV, both the target and the organs-at-risk are considered. This can visually help planners to choose which beam angles should be avoided *a priori*.

5.1.2 Beam's-eye-view and dose matrices

A multileaf collimator located inside the head of the linear accelerator is used to shape the beam of radiation generated by the linear accelerator [34, 84]. To calculate the radiation dosages that can be delivered by a beam applied from a given angle, the rectangular aperture obtained by opening the collimator as widely as possible is divided into rectangular subfields arranged in a regular $M \times N$ rectangular pattern, as shown in Figure 27. Each of the subfields is called a *pencil beam* or *beamlet*. M represents the number of leaf pairs in the multileaf collimator, while N represents the number of possible settings we allow for each leaf. We identify each beamlet by the index pair (i, j), where $i = 1, 2, \ldots, M$ and $j = 1, 2, \ldots, N$. In our work, the leaves of the multileaf collimator are 1 cm wide, and a pencil beam is assigned a length of 0.5 cm. Thus, for a 10 cm by 10 cm field, we would use $M = 10$ and $N = 20$, giving a total of 200 beamlets.

A separate three-dimensional dose distribution is computed for each pencil beam. The dose distribution matrix for each pencil beam from each angle is calculated using a Monte Carlo technique, which simulates the track of individual radiation particles, for a large number of particles. A unit-intensity, non-wedged beam is assumed for the purposes of these calculations.

In conformal radiotherapy, the shape of each beam is set to match the beam's-eye view (BEV) of the tumor volume, which is essentially the projection of the three-dimensional shape of the tumor onto the plane of the multileaf collimator. One technique for determining the BEV is to employ a ray-tracing algorithm from the radiation source to the tumor volume, setting the beam's-eye view to include

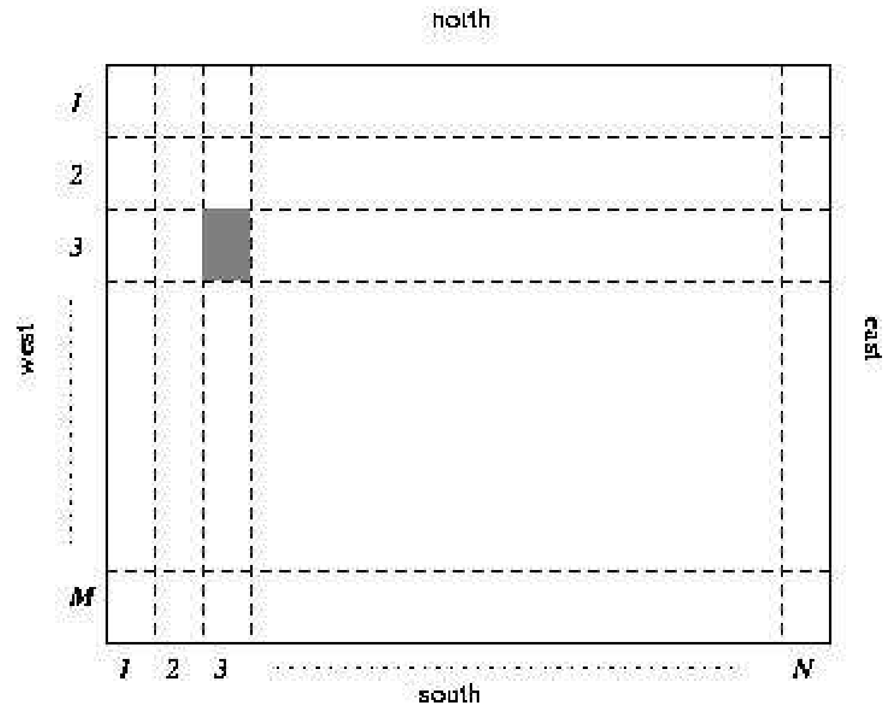


Figure 27: Division of Aperture into Pencil Beams (shaded area represents one beamlet)

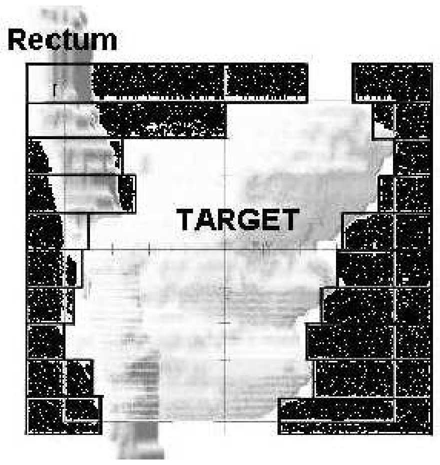


Figure 28: An example of Beam's-eye-view

all of the rays that pass through the tumor volume. We use an alternative approach based on the dose matrices of the pencil beams. We include in the BEV all pencil beams whose field of significant dose intersects with the target region. To be specific, given a threshold value T, we include a pencil beam in the BEV if its dose delivered to at least one voxel within the target region is at least $T\%$ of the dose delivered by that pencil beam to *any* voxel. Figure 28 shows an example of a BEV. Implementation of a BEV by a multileaf collimator is shown in Figure 28(b).

Once the BEV from a particular angle has been chosen, we can construct the dose matrix for the BEV aperture by simply summing the dose matrices of all the pencil beams that make up the BEV.

The choice of threshold parameter T is critical. If the value of T used in the determining the BEV is too small, the BEV overestimates the target, producing an aperture that irradiates not only the target but also nearby normal tissue and organs at risk. On the other hand, if the value of T is too large, the BEV underestimates the target, and the optimizer might not be able to find a solution that adequately delivers radiation dose within the required range to all parts of the target. The best value of T to use depends somewhat on the shape of the tumor. We choose T as the minimum value such that the resulting BEVs provide a complete 3D coverage of the target from all beam angles considered in the problem. Based on our experiments, a value of T of between 10% and 15% appears to be appropriate.

5.2 Optimization Models

A variety of optimization models are provided to users. Although they are described more fully in Chapter 4, we give a few examples of optimization models that are available in GAMS files on our website *http://www.cs.wisc.edu/˜ferris/3dcrt/.* All definitions of variables are therefore given in Chapter 4.

5.2.1 Beam weight optimization

Linear Programming Linear programming (LP) is a powerful mathematical tool for radiation treatment planning optimization. It has been used to improve conventional treatment planing techniques [1, 43, 53, 61, 67]. The strength of LP is its ability to control *hot* and *cold* spots or integral dose on the organs, and the presence of many state-of-the-art LP solvers. There are two weaknesses of LP in a practical sense. The first is that LP fails to approximate a solution when a solution does not exist. The second weakness is that clinically desirable objective functions sometimes may not be well approximated by linear functions.

An example of LP:

$$
\begin{array}{lrcll}
\min\limits_{w} & \multicolumn{4}{l}{\lambda_t \|D_{\mathcal{T}} - \theta e_{\mathcal{T}}\|_\infty + \lambda_s \dfrac{\|(D_{\mathcal{S}} - \phi e_{\mathcal{S}})_+\|_1}{\text{card}\,(\mathcal{S})} + \lambda_n \dfrac{\|D_{\mathcal{N}}\|_1}{\text{card}\,(\mathcal{N})}} \\
\text{s.t.} & D_{(i,j,k)} & = & \sum_{A \in \mathcal{A}} w_A \mathcal{D}_{A,(i,j,k)}, & \forall (i,j,k) \in \mathcal{T} \cup \mathcal{S} \cup \mathcal{N}, \\
 & 0 & \leq & w_A \leq \frac{u}{\rho_A} \psi_A, & \forall A \in \mathcal{A}.
\end{array}
\tag{5.1}
$$

Quadratic Programming One difference between linear programming (LP) and quadratic programming (QP) is in the objective function formulation: LP uses a linear objective function, while QP uses a quadratic objective function. Most of the works in the literature try to minimize the sum of the deviation of the dose of voxels to the prescribed dose [17, 60, 67, 73].

A QP example:

$$\begin{array}{lrcl} \min\limits_{w} & \multicolumn{3}{l}{\lambda_t \dfrac{\|D_{\mathcal{T}} - \theta e_{\mathcal{T}}\|_2^2}{\text{card}\,(\mathcal{T})} + \lambda_s \dfrac{\|(D_{\mathcal{S}} - \phi e_{\mathcal{S}})_+\|_2^2}{\text{card}\,(\mathcal{S})} + \lambda_n \dfrac{\|D_{\mathcal{N}}\|_2^2}{\text{card}\,(\mathcal{N})}} \\ \text{s.t.} & D_\Omega & = & \sum_{A \in \mathcal{A}} w_A \mathcal{D}_{A,\Omega}, \quad \Omega = \mathcal{T} \cup \mathcal{S} \cup \mathcal{N}, \\ & 0 & \leq & w_A \leq \frac{u}{\rho_A} \psi_A, \qquad \forall A \in \mathcal{A}. \end{array} \tag{5.2}$$

5.2.2 Beam angle selection and weight optimization

Mixed Integer Programming (MIP) is a straight-forward technique for selecting beams angles among many candidates. The weakness, however, is its long runtime.

An example of MIP:

$$\begin{array}{lrcl} \min\limits_{w,\psi} & \multicolumn{3}{l}{\lambda_t \|D_{\mathcal{T}} - \theta e_{\mathcal{T}}\|_\infty + \lambda_s \dfrac{\|(D_{\mathcal{S}} - \phi e_{\mathcal{S}})_+\|_1}{\text{card}\,(\mathcal{S})} + \lambda_n \dfrac{\|D_{\mathcal{N}}\|_1}{\text{card}\,(\mathcal{N})}} \\ \text{s.t.} & D_\Omega & = & \sum\limits_{A \in \mathcal{A}} \mathcal{D}_{A,\Omega} \cdot w_A, \quad \Omega = \{\mathcal{T} \cup \mathcal{S} \cup \mathcal{N}\} \\ & 0 & \leq & w_A \leq \frac{u}{\rho_A} \psi_A, \qquad \forall A \in \mathcal{A}, \\ & \sum_{A \in \mathcal{A}} \psi_A & \leq & K, \\ & \psi_A & \in & \{0, 1\}, \qquad\qquad \forall A \in \mathcal{A}. \end{array} \tag{5.3}$$

Note that (5.3) can also be used to optimize only beam intensities by setting $K = |\mathcal{A}|$.

5.2.3 Optimizing beam angles, wedge orientations, and beam weights

Finally, we described an optimization model that simultaneously optimizes beam angles, wedge orientations, and beam intensities in Chapter 4.

A MIP example:

$$\begin{array}{lrcll}
\min\limits_{w} & \multicolumn{4}{l}{\lambda_t \|D_{\mathcal{T}} - \theta e_{\mathcal{T}}\|_\infty + \lambda_s \dfrac{\|(D_{\mathcal{S}} - \phi e_{\mathcal{S}})_+\|_1}{\text{card}\,(\mathcal{S})} + \lambda_n \dfrac{\|D_{\mathcal{N}}\|_1}{\text{card}\,(\mathcal{N})}} \\
\text{s.t.} & D_\Omega & = & \sum\limits_{A \in \mathcal{A}, F \in \mathcal{F}} w_{A,F} \mathcal{D}_{A,F,\Omega}, & \Omega \in \mathcal{T} \cup \mathcal{S} \cup \mathcal{N}, \\
& \frac{u}{\rho_A} \psi_A & \geq & w_{A,0} + \tau_1 \sum\limits_{F \in \mathcal{F} \setminus 0} w_{A,F} & \\
& K & \geq & \sum_{A \in \mathcal{A}} \psi_A, & \\
& w_{A,F} & \geq & 0, & \forall A \in \mathcal{A}, \\
& \psi_A & \in & \{0,1\}, & \forall A \in \mathcal{A}.
\end{array} \qquad (5.4)$$

5.3 Optimization Software

5.3.1 Directory setup

Three directories are recommended to store necessary data and programs (see Figure 29). The directory *Beamdata* stores the original data that contains the

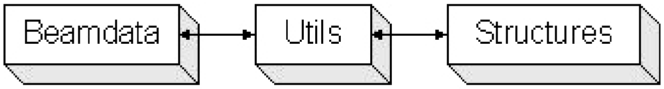


Figure 29: Three directories are recommended

dose distribution for beamlets from each angle explained earlier.

The directory *Structures* stores three-dimensional organ geometries of the tumor region, organs-at-risk, and normal tissue. They are outlined by a physician on a set of CT or MRI images. The physician labels some of the voxels as PTV and others as OAR.

Utils directory contains all programs that utilize the information given above two data directories for the treatment planning optimization. This is the directory where treatment plans are designed. The programs in this directory include GAMS optimization models, a MATLAB routine to generate appropriate data for the optimization models, a MATLAB routine to execute GAMS optimization models, a MATLAB routine to make DVH plots, and a MATLAB program to generate simulated organ structures.

5.3.2 Environment and system requirements

The treatment planning process is carried out in MATLAB environment. We describe the following system requirement to run the provided programs properly.

1. A PC running on Linux (or Solaris, or Windows 98/ME/NT/2000/XP operating system).
2. At least 500 MB free hard drive space (1 GB free space is recommended).

3. A licensed MATLAB as a working environment.
4. A mathematical modeling language GAMS with appropriate solvers for LP, MIP, and NLP.

5.3.3 An overview of the optimization software

There are three stages in generating a treatment plan in our approach.

1. *gendata(probName):* converts/saves the original data (stored in continuous coordinates) into corresponding discrete coordinates for the optimization. The input argument is a MATLAB structure array that defines the basic problem configuration (see Section 5.4.1.) It also generates a file that defines basic GAMS sets and parameters on the fly using an input file that specifies machine configuration as well as the user's preference for the treatment plan.

   ```
   >> gendata(probName);
   ```

2. *rungms:* generates a treatment plan. It can take multiple input strings such as the name of a GAMS file, MATLAB structure arrays of organs, and a MATLAB structure array of input parameters.

   ```
   >> [Dose,PTV,OAR] = rungms('qp',target,sensitive);
   ```

   Note that outputs of target and sensitive are mapped internally to "PTV" and "OAR" respectively. The corresponding sets are returned from the "rungms" program.

3. *dvh:* makes dose-volume histogram (DVH) plots for inputs specified.

   ```
   >> dvh(Dose,PTV,OAR);
   ```

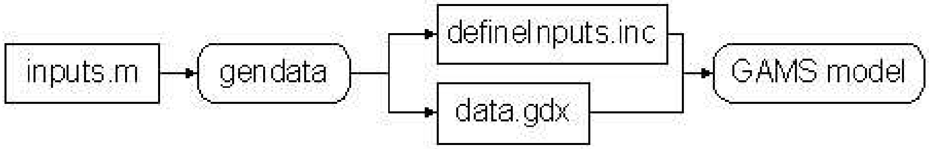


Figure 30: Generating data for GAMS optimization model

4. *neworgans:* enables users to create simulated organ structures within a cylinder. This type of structures can be useful for tuning optimization models.

5.4 Treatment Planning Process

We demonstrate the entire treatment planning process (presented in Section 5.3.3) using a prostate cancer data. There are two organs in this example, namely "tumor" and "rectum". The tumor volume has 5245 voxels, while the rectum consists of 1936 voxels. Suppose we are interested in optimizing beam intensities of 36 beam angles for a treatment plan. Using this basic input data, we walk through each treatment planning process in the next few sections.

5.4.1 Data generation

All coordinates stored in both *Beamdata* and *Structures* directories are continuous, which are not usable as GAMS set indices. The first step is to generate appropriate data for the GAMS optimization model. Figure 30 illustrates the data generation process. *gendata* first reads in a user input (generated using a MATLAB function *inputs.m*) that specifies the total number of beam angles considered in the problem, the number of beam angles for the treatment, where to find the original

Beamdata and *Structures* of interest, GAMS set names for the organ structures, and the GAMS include file name that will store set and parameter definitions, organ geometries, and dose matrix.

inputs.m uses the MATLAB's "struct" command to collect necessary information to generate appropriate data for the treatment planning. An example of *inputs.m* looks as follows:

```
function data = inputs(prob_name)
%****************************************************************************
%Input data required for conformal radiation treatment planning optimization:
%****************************************************************************
% nAngle       := number of beam angles
% beamcutoff   := cutoff dose value to generate beams-eye-view (ex. 12%)
% margin       := margin of voxels to generate inNormal(I,J,K)==rind of PTV
% use_wedge    := whether a wedge filter is considered in the optimization
% is_cylinder  := if the full data for all angles are given, say ’no’
% baseDir      := base directory where all the data is stored
% beamDir      := Directory name of the intial beam data
% structDir    := Directory name of the intial structures
% beamID       := beam data identification, ex. 10x10 or 20x20 or imat
% rindSetName  := a set of voxels in rind of tumor
% oarSetName   := a set that contains all voxels in orans-at-risk
% gdxDataName  := a file that contains all GDX data for GAMS
% structurefile:= input structure file names in Structures directory
% setName      := set names for the structures in the GAMS file
```

```
if strcmp(prob_name,'prostate')
  st = struct('files',{'ptv.dat','rectum.dat'},...
              'sets',{'prostate','rectum'});
elseif strcmp(prob_name,'pancreas')
  st = struct('files',{'GTV08.dat','COR01.dat','LIV00.dat',...
                       'LTT03.dat','RTT04.dat'},...
              'sets',{'Pancreas','spCord','Liver',...
                     'LKidney','RKidney'});
else
  st = [];
end
data = struct(...
  'nAngle',36,...
  'kBeams',10,...
  'beamcutoff',11,...
  'margin',2,...
  'use_wedge','no',...
  'is_cylinder','yes',...
  'baseDir','/p/cure-cancer/work1/LIM/',...
  'beamID','10x10',...
  'beamDir','Beamdata_cylinder',...
  'structDir','Structures',...
  'rindSetName','inNormal',...
  'oarSetName','Sensitive',...
  'gdxDataName','data.gdx',...
  'gdxIncludeFile','defineInputs',...
  'structures',st);
```

We take two steps to generate data that is needed for the rest of treatment planning. The first step is to collect appropriate input data for the MATLAB command "gendata":

```
>> prob = inputs('prostate');
```

This creates a MATLAB structure array "prob" with all default values and strings of "inputs.m" for the treatment planning problem.

The second step is to produce the data using the inputs created above:

```
>> gendata(prob);
```

This generates both a GAMS include file (defineInputs.inc) and a GDX (GAMS Data Exchange) file [76], typically named *data.gdx*, that are required for any of the GAMS models.

The file *defineInputs.inc* is a problem specific file that defines sets and parameters that are used in the optimization model. It has six components. First, basic sets and their dimensions are defined for solving the optimization problems. A large value of the maximum index is typically assigned to each three-dimensional coordinate because the dimension of the coordinate is not known in advance. Since each set of the coordinate overestimates its maximum index, this generates unnecessarily large number of voxels in the problem; which can lead to a very slow solution time. To overcome this, sets of the three-dimensional coordinates and their dimensions are defined when the locations of the organ structures are identified. Once organ structures are constructed for the optimization, dimensions of sets can be defined based on the three-dimensional locations of organs and normal tissue surrounding the organs. An example is shown below.

```
sets
       I       /0*127/
       J       /0*127/
       K       /0* 33/
       nAngle  /0*35/;
```

The next component is to define GAMS global variables used in the GAMS file. In our example, the following two lines

```
$setglobal target ’prostate’
$setglobal sensitive ’rectum’
```

make the global variable target contain “prostate” and “sensitive” for “rectum” in the GAMS file.

The third component is to define necessary sets for the GAMS file. Sets “prostate” and “rectum” provide coordinates of organs, “PTV” and “OAR” are auxiliary set definitions for the target and the sensitive structures respectively. The set “inNormal” ($\mathcal{R}_\Delta(\mathcal{T})$) was defined in Chapter 4 to denote the normal tissue from a rind around PTV, “outNormal” is for a set of voxels that does not belong to the organs of interest (excluding voxels in “inNormal”), “Normal” is defined as a union of “inNormal” and “outNormal”, and the name of the parameter to store the dose distribution is also defined:

```
sets
   prostate(I,J,K), rectum(I,J,K),   PTV(I,J,K), OAR(I,J,K),
   Normal(I,J,K),   inNormal(I,J,K), outNormal(I,J,K);
parameter Dose(I,J,K,nAngle);
```

In the next component, a set “allorgans” defines a collection of sensitive structures of interest. The next line is a gateway between MATLAB and GAMS that allows the user to update the values of global strings in the GAMS file. For example, we may wish to update the global variable “sensitive” to contain a subset of all sensitive structures.

```
set allorgans /%sensitive%/;
$if exist matglobs.gms  $include matglobs.gms
set organs(allorgans) /%sensitive%/;
```

All necessary data for the optimization is stored (by “gendata”) in GAMS GDX format. Therefore, the next component is written to retrieve the data in the GAMS file:

```
$GDXIN data.gdx
$LOAD  PTV=%target% prostate rectum inNormal
$LOAD  Dose
$GDXIN
```

“$GDXIN data.gdx” opens the GDX file “data.gdx” for reading. The second line is used to load sets from “data.gdx”. Note that a set can be renamed at this stage:

set “target” is now named “PTV”. The last line “$GDXIN” closes the access to the file “data.gdx”.

Finally, a set “Sensitive” of the sensitive structures is defined as a collection of “allorgans” in the GAMS file. Each organ must be defined explicitly as shown in the second line below:

```
  set Sensitive(I,J,k,allorgans);
  Sensitive(I,J,K,'rectum') = yes$rectum(I,J,K);
```

The GAMS include-file *defineInputs.inc* that contains all of these components needs to be included at the very beginning of any GAMS files in our toolbox:

```
$include defineInputs.inc;
```

5.4.2 Constructing GAMS optimization models

We illustrate a GAMS optimization model for (5.2) below. Most of the notation used in this GAMS file tries to imitate the mathematical symbols used in (5.2) with a few exceptions: PTV represents $\mathcal{T}$, OAR is for $\mathcal{S}$, Normal is used for $\mathcal{N}$, and sumDose represents D.

```
* qp.gms
* This program solves 3D conformal radiation treatment problem.
* The solution includes: optimal beam weights
option limrow=0, limcol=0, solprint=off;
OAR(I,J,K) = yes$Sensitive(I,J,k,allorgans);
scalar    theta   'dose level prescribed for target'          / 1 /;
scalar    ubar    'dose upper bound on the target voxels'     /1.15/;
$include defineInputs.inc
parameter phi(allorgans) 'hot spot control parameter for OAR';
phi(allorgans) = 0.3;
parameter Rho(nAngle) 'maximum dose level deposited to the target';
Rho(nAngle) = smax(PTV,Dose(PTV,nAngle));
OAR(I,J,K) = yes$(not PTV(I,J,K) and sum(organs$Sensitive(I,J,K,organs),1));
outNormal(I,J,K)=yes$(not PTV(I,J,K) and not OAR(I,J,K) and
                      not inNormal(I,J,K) and
        (mod(ord(I),2)=0 and mod(ord(J),2)=0 and mod(ord(K),2)=0));
Normal(I,J,K) = yes$(inNormal(I,J,K) or outNormal(I,J,K));
positive variables  w(nAngle), dS(I,J,K,allorgans), sumDose(I,J,K);
variable    z;
equations   Def4sens(I,J,K),Def4sumDose(I,J,K,allorgans), Obj;
Def4sumDose(I,J,K)$(PTV(I,J,K) or OAR(I,J,K) or Normal(I,J,K)) ..
  sumDose(I,J,K) =e= sum(nAngle,Dose(I,J,K,nAngle)*w(nAngle));
Def4sens(OAR,allorgans)..
  -sumDose(OAR) + dS(OAR,allorgans) =g= -phi;
Obj ..
z =e=  sum(PTV,sumDose(PTV)*sumDose(PTV))/card(PTV)
     + sum(allorgans,sum(OAR,dS(OAR,allorgans)*dS(OAR,allorgans))
       /card(allorgans))
     + sum(Normal,sumDose(Normal)*sumDose(Normal))/card(Normal);
sumDose.up(PTV) = ubar*theta;
```

```
w.up(nAngle) = ubar/Rho(nAngle);

model conf / all/;

solve conf using nlp minimizing z;
```

For debugging purposes, this model can be executed directly at the command prompt:

```
  % gams qp
```

Note that any user-defined GAMS files are not allowed to be named either "matglobs.gms" or "matdata.gms" because they already exist in the system.

5.4.3 Generating a treatment plan

The GAMS file can be run within MATLAB [26]. Some of the specified three-dimensional organ geometries can be returned back into the MATLAB workspace with the final dose distribution being the first (required) output. The GAMS library utility "matout" can be used for this. First, we calculate the final dose distribution right after the "solve" statement in the GAMS file because some of the normal voxels were not considered in the optimization:

```
    Normal(I,J,K) = yes$(not PTV(I,J,K) and not OAR(I,J,K));

    sumDose.l(I,J,K)$(PTV(I,J,K) or OAR(I,J,K) or Normal(I,J,K))

                    = sum(nAngle,Dose(I,J,K,nAngle)*w.l(nAngle));
```

We then add the following line to the GAMS file for returning the final dose distribution as four-dimensional matrix into MATLAB.

```
  $libinclude matout sumDose.l I J K
```

All matrices of organ geometries are three-dimensional sets. In order to return these matrices back into MATLAB, GAMS *$libinclude matout* command must be used for each organ of interest. Since the organs will be specified in MATLAB, the line corresponding to each organ is written on the fly when *rungms* command is triggered in MATLAB.

The routine "rungms" can take up to several inputs. It (*rungms*) can have three different types of inputs: a string containing the GAMS file name, MATLAB structure arrays that define organs, and a MATLAB structure array that defines values of parameters used in the GAMS file. A general input format for *rungms* looks as follows:

```
rungms('GAMS file name [-options]',organ1,organ2,organ3,...,data);
```

The first (and only required) input string must be the GAMS file name that is currently located at *Utils* directory. GAMS options can be added followed immediately after the GAMS file name. We can run the GAMS file "qp.gms" in MATLAB as follows:

```
>> Dose = rungms('qp');
```

However, it is typical that a user will wish to visualize the DVH plots of various structures in the problem. To facilitate this, we use optional input and output arguments to pass the coordinates used by the GAMS model back to MATLAB.

The optional arguments are MATLAB structures representing organs. Each structure must have a name field. The name field must have the string value that is identical to the set name used in GAMS. For an example, we define a MATLAB structure array for the target, another for the sensitive structure, and the third for the normal tissue as follows:

```
  >> target    = struct('name',{'prostate'});
  >> sensitive = struct('name',{'rectum'});
  >> normal    = struct('name',{'Normal'});
```

For example, the following line

```
  >> [Dose,PTV,OAR,Normal] = rungms('qp',target,sensitive,normal);
```

first creates a file *matoutDef.inc* that instructs the GAMS model to return the specified organ coordinates:

```
* matoutDef.inc
  $libinclude matout prostate I J K
  $libinclude matout rectum I J K
  $libinclude matout Normal I J K
```

Therefore, we must add the following line

```
  $if exist matoutDef.inc  $include matoutDef.inc
```

at the end of the GAMS file as follows:

```
  .... GAMS program ....
  $libinclude matout sumDose.l I J K
  $if exist matoutDef.inc  $include matoutDef.inc
```

The specified GAMS file is executed and returns the four-dimensional matrix of the final dose distribution. In addition, it can also return as many sets as specified in *matoutDef.inc.* In our example, a set of three-dimensional coordinates of the prostate and another for the rectum are returned along with the final dose matrix. Any structure coordinates that are returned to the MATLAB workspace can be used to evaluate the treatment quality using DVH plots. The last line in “matoutDef.inc” is to return either the solution vector (if wedges are not used) or the solution matrix (if wedges are used) back into MATLAB. For an example to retrieve a solution w, just execute the following line:

```
 >> [D,PTV,w] = rungms('qp',target,[]);
```

Note that it is important to place w at the end of the output argument. Furthermore, in the “rungms” example above, the “qp” model will have an empty sensitive structure.

5.4.4 Solution examination using DVH plot

The quality of a treatment plan is typically specified and evaluated using the DVH. To make a DVH plot of the current solution, final dose distribution and three-dimensional organ coordinates are passed through a MATLAB routine *dvh.* “Dose” must be the first input argument for “dvh”. In MATLAB prompt,

```
  >> dvh(Dose,PTV,OAR);
```

This invokes a MATLAB figure with dose volume histograms of the specified organs. The user can also specify (not required) the line property on the DVH plot as follows:

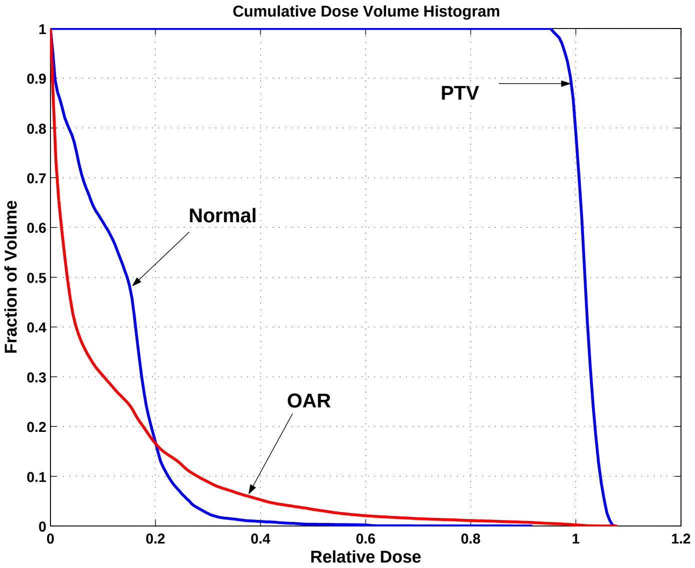


Figure 31: Dose Volume Histogram

```
>> dvh(Dose,PTV,'b-',OAR,'r');
```

where the color blue ('b-') with a solid line is specified for the "PTV" and red ('r') for the "OAR." More choices of the line properties can be found in MATLAB by typing:

```
>> help plot
```

An example of DVH is shown in Figure 31. The *X-axis* is normalized so that the target prescribed dose (θ) is one. The *Y-axis* represents the fraction of the volume. For example, the line of the normal tissue approximately passes through the coordinate $(0.2, 0.2)$. This means that 80% of the normal tissue receives 20% or less of the target prescribed dose level. Note that the labels of structures are created manually after the DVH plot is made using MATLAB figure editor.

5.5 Creating Simulated Organ Structures

5.5.1 Simulated organ structure generation procedure

We provide a MATLAB routine to create simulated organ structures. This routine allows users to outline two-dimensional slices of a target and a sensitive structure in order to generate three-dimensional organ shapes. Once a user executes the MATLAB routine “neworgans”, MATLAB immediately asks the user a series of questions about the organ structures to generate:

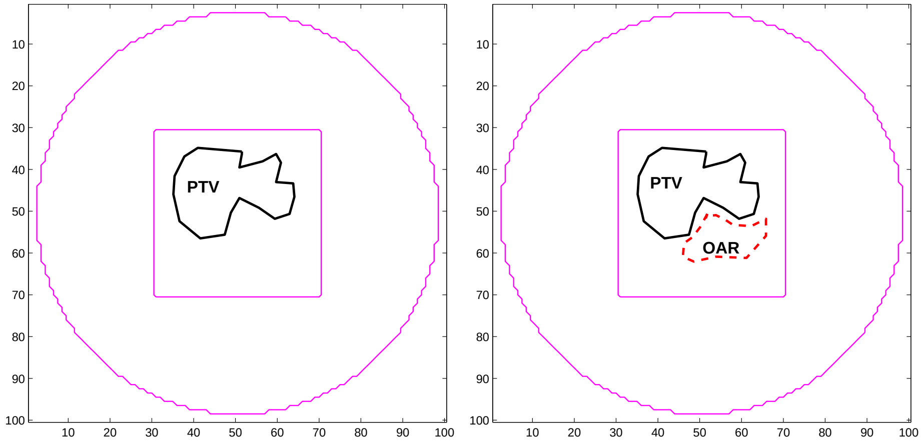
```
>> neworgans
Enter the target file name :=> 'head'
Enter the OAR file name:=> 'neck'
z slice index runs from 1 to 32.
Input first slice for target: 10
Input last slice for target:  15
Please contour the target at slice 10
```

At this point, a MATLAB figure window is invoked for the user to start outlining organ shapes. The user is asked to continue for the following process:

```
Please contour the target at slice 10
Repeat (1=yes,0=no):0
```

The “Repeat” gives the user an option to redraw the most recent outline of the slice if necessary.

An example of this is shown in Figure 32(a). Next, the user is asked to outline the shape of the sensitive structure on the same slice shown in Figure 32(a):



(a) A 2D shape of the new target is outlined

(b) A 2D shape of the new sensitive structure is also outlined

Figure 32: Creating simulated organ structures

```
Please contour the region at risk at slice 10

Repeat (1=yes,0=no):0

Please contour the target at slice 11
```

Figure 32(b) shows such an example. This process continues until the organ shape of the last slice is outlined. Once the complete organ structures are defined, the coordinates are stored into *Structures* directory with the filename specified by the user with *.dat* filename extension (for example, head.dat, neck.dat).

5.5.2 An example of treatment planning procedure with the new organs

The following MATLAB steps illustrate a treatment planning procedure with the organ structures produced in the previous section.

1. Generating model data: Since we have created new organ structures, we need to update the problem structure. For example, to use the “head” and “neck” data just generated, we just execute:

```
>> prob = inputs(’organs1’);
>> prob.structures = struct(’files’,{’head.dat’, ’neck.dat’},...
               ’sets’,{’head’,’neck’});
```

Then the model data can be generated using “gendata” command:

```
>> gendata(prob);
```

2. Treatment Planning: First, organ structures are defined using MATLAB “struct” command:

```
 >> target    = struct(’name’,{’head’});
 >> sensitive = struct(’name’,{’neck’});
 >> normal    = struct(’name’,{’Normal’});
```

We then run GAMS file “qp.gms” using two organ structures (”head” and “neck”) as follows:

```
 >> [Dose,P,S,N]= rungms(’qp’,target,sensitive,normal);
```

3. Solution Examination: The dose-volume histogram can be made to examine the treatment quality:

```
 >> dvh(Dose,P,S,N);
```

5.6 Techniques to improve solution time

5.6.1 Data sampling

The given data for radiation treatment optimization problem is typically very large. The reason is that the initial dose matrix for the optimization model stores dose contribution to each voxel considered in the optimization. Typical (fixed default) dimensions considered in the radiation treatment planning problems are $150 \times 150 \times 100 = 2.25$ million. In our example, the corresponding dimensions were reduced to $128 \times 128 \times 34 = 0.557$ million, which is about 25% of the fixed dimension.

Vast amount of voxels comprise the normal tissue. Although voxels in the normal tissue are important for the final treatment plan, some voxels that are away from the target structure are less significant to the optimal treatment plan dose distribution. A random sampling of voxels is used to speed up the computation in the literature [62]. 10% of each structure is randomly sampled. The sampling scheme is also noted elsewhere [42]. The sampling approach we use was discussed in Section 4.4.3. This can be seen in several of the example GAMS files available at *http://www.cs.wisc.edu/˜ferris/3dcrt/.*

5.6.2 Robust modeling and iterative solution approach

We have used some techniques to enhance the performance of the optimization models. If the optimization problem is nonlinear and nonconvex, generating a good starting solution becomes very important to ensure that the resulting solutions are robust and reliable (an example is given in Section 3.3.) When solving a mixed integer programming model, tightening the solution space can significantly improve the solution performance as discussed in Section 4.4.1.

The second and very powerful technique in solving a large-scale optimization problem is the use of the "iterative solution scheme" discussed in Section 4.4.3. In the iterative solution scheme, optimization model is typically solved using a set of sampled data points and relaxed constraints. Based on the solution from the previous solve, the next optimization process narrows down the solution search space for the treatment goals the planner wants to achieve.

5.6.3 GAMS options

Since the amount of data for radiation treatment planning problems are typically very large, dealing with beam data in a text-file format may take a lot of storage space. A drawback with large data in GAMS is that often users have to sit and wait for GAMS to load and unload the data. It can be very time-consuming. Recently, GAMS published a contributed utility GDX [76]. The GDX utility handles data in a binary format, which can save a lot of storage space. It is also much faster to work with the GDX data in GAMS environment.

We give a brief description of a few useful GAMS options:

1. *optfile* can be very useful for solving LP and MIP models. GAMS [31] provide a number of LP and MIP solvers to choose from. Some algorithms work better than others depending on the problem of interest. For example, CPLEX gives four options to solve an LP: 1 for the primal simplex, 2 for the dual

simplex, 3 for the network simplex, and 4 for the barrier method. In general, barrier method seemed to generate solutions faster than other methods in solving optimization problems we have tried. For example, we used the following schemes for solving all of our MIP problems. The file “cplex.opt” contains the following lines:

```
lpmethod 4
startalg 4
cuts no
covers -1
```

“lpmethod 4” specifies that the barrier method is used to solve an LP. “startalg 4” is to use barrier with crossover for solving the initial relaxation of a MIP. Other options for startalg are: 1 for the primal simplex, 2 for the dual simplex, 3 for network followed by dual simplex, 4 for barrier with crossover, 5 for the dual simplex to iteration limit, then barrier, 6 for the barrier without crossover.

“cuts no” is to turn off all CPLEX cut generation options. Default is “yes”. “covers (integer)” can be used to determines whether or not cover cuts should be generated during optimization (default = 0).

```
-1 Do not generate cover cuts
 0 Determined automatically
 1 Generate cover cuts moderately
 2 Generate cover cuts aggressively
```

To use this CPLEX option file, a line

```
model_name.opt = 1;
```

must be inserted right before the GAMS "solve" statement.

2. *profile* can be used to find where the excessive time is being used. GAMS generates information on statement execution time and associated memory usage by employing profile. This option can be invoked either in the GAMS file

   ```
   option profile = 3 (or 0,1,2)
   ```

   or on the prompt

   ```
   gams example profile = 3 (or 0,1,2)
   ```

3. *prioropt:* instructs CPLEX to use priority branching information passed by GAMS through the "Variable.prior" parameters. The syntax is

   ```
   VariableName.PriorOpt = 1 (or 2,3,4,...);
   ```

   A variable with a smaller number gets higher priority.

5.7 Summary

Optimization tools are developed in MATLAB and GAMS environments. We have generated a variety of GAMS optimization models that implement the models discussed in Chapter 4. These are available at *http://www.cs.wisc.edu/˜ferris/3dcrt/.*

Chapter 6

Conclusions

We have developed a collection of optimization frameworks for radiation treatment planning. First, we presented a unified and fully automated radiosurgery treatment planning framework for the Gamma Knife machine. The optimization model is a nonlinear, non-convex, and mixed integer program. We showed how to approximate the solution of this problem by a sequence of nonlinear programs and a single linear mixed integer program. To obtain reliable solutions, we developed a new and efficient technique to generate a good starting point for the nonlinear program. Based on the fact that a shot of radiation (ellipsoid) forms approximately a sphere, we introduced a technique that uses a variant of a sphere packing approach combined with the Medial Axis Transformation (Skeleton), often used in computer graphics. Using a good starting point, the nonlinear optimization problem is solved using CONOPT (generalized reduced gradient method.) The key optimization parameters were the isocenters for radiation doses, the collimator (helmet) sizes, and the intensity for each shot of radiation. We showed that the optimization model was fast enough to generate an optimal treatment plan (within 20 minutes), flexible enough to apply to a variety of tumor types, and robust enough to obtain high quality (conformal and uniform) treatment plans for any size and any shape of tumor. This tool is currently in use at the Radiation

Oncology Department at the University of Maryland School of Medicine at the University of Maryland.

Secondly, we have developed a framework for three-dimensional conformal radiation treatment planning. In this framework, a variety of optimization models were introduced for treatment planning problems. The optimization problems were formulated as mixed integer linear programming and quadratic programming problems. We showed that different objective function formulations could be used for different treatment goals. We presented the optimization model that simultaneously optimizes three key parameters: beam weights, beam angles, and wedge orientations. The framework offers fast delivery of the treatment plan as well as the capability for control of dose-volume constraints on organs as typically described by the planner. Since the data set required by the optimization formulations was very large, we introduced techniques to reduce the data requirements and the complexity of the problem. Specifically, we introduced tighter *a priori* bounds on the beam weights, reduction of the number of voxels to be considered in the optimization, and a three-phase scheme in which a sequence of progressively more realistic optimization models is solved to obtain an approximate solution. Using all these techniques, we demonstrated a 97% improvement in computational time over direct solution of the full-resolution problem on a clinical data set.

Finally, optimization software was developed for radiation treatment planning. We demonstrated a treatment planning procedure with this software. First, a MATLAB routine was used to generate appropriate dose matrices based on the beam's-eye-view approach. Secondly, a GAMS optimization model was executed to find a solution for the beam angles, beam weights, and wedge orientations.

A MATLAB routine was used to examine the quality of the resulting treatment plan. Since some optimization modelers may also be interested in creating unique shapes of organs to tune their models, we provided a MATLAB routine to create simulated organ structures.

Bibliography

[1] G. Bahr, J. Kereiakes, H. Horwitz, R. Finney, J. Galvin, and K. Goode, *The method of linear programming applied to radiation treatment planning*, Radiology, 91 (1968), pp. 686–693.

[2] G. Bednarz, D. Michalski, C. Houser, M. Huq, Y. Xiao, P. Anne, and J. Galvin, *The use of mixed-integer programming for inverse treatment planning with pre-defined field segments*, Physics in Medicine and Biology, 47 (2002), pp. 2235–2245.

[3] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, Athena Scientific, 1995.

[4] ——, *Network Optimization: Continuous and Discrete Models*, Athena Scientific, 1998.

[5] H. Blum, *A transformation for extracting new descriptors of shape*, in Models for the Perception of Speech and Visual Form, W. Wathen-Dunn, ed., MIT Press, 1967, pp. 362–380.

[6] T. Bortfeld, A. L. Boyer, W. Schlegel, D. L. Kahler, and T. J. Waldron, *Realization and verification of three-dimensional conformal radiotherapy with modulated fields*, International journal of radiation oncology: Biology, Physics, 30 (1994), pp. 899–908.

[7] T. Bortfeld, J. Burkelbach, R. Boesecke, and W. Schlegel, *Methods of image reconstruction from projections applied to conformation radiotherapy*, Physics in Medicine and Biology, 25 (1990), pp. 1423–1434.

[8] T. Bortfeld and W. Schlegel, *Optimization of beam orientations in radiation-therapy - some theoretical considerations*, Physics in Medicine and Biology, 38 (1993), pp. 291–304.

[9] J. D. Bourland and Q. R. Wu, *Use of shape for automated, optimized 3D radiosurgical treatment planning*, SPIE Proceedings of International Symposium on Medical Imaging, (1996), pp. 553–558.

[10] J. W. Brandt and V. R. Algazi, *Continuous skeleton computation by voroni diagram*, CVGIP:Graph Models Image Processing, 55 (1992), pp. 329–338.

[11] L. Brewster, G. S. Mageras, and R. Mohan, *Automatic-generation of beam apertures*, Medical Physics, 20 (1993), pp. 1337–1342.

[12] A. Brooke, D. Kenderick, A. Meeraus, and R. Raman, *GAMS: User's Guide*, GAMS Development Corporation: http://www.gams.com/docs/, 1998.

[13] A. Brooke, D. Kendrick, and A. Meeraus, *GAMS: A User's Guide*, The Scientific Press, South San Francisco, California, 1988.

[14] Y. Censor, M. Altschuler, and W. Powlis, *On the use of cimmino's simultaneous projections method for computing a solution of the inverse problem*

in radiation therapy treatment planning, Inverse Problems, 4 (1988), pp. 607–623.

[15] Y. Censor and S. Schwartz, *An iterative approach to plan combination in radiotherapy*, International Journal of Bio-Medical Computing, 24 (1989), pp. 191–205.

[16] G. Chen, D. R. Spelbring, C. Pelizzari, J. Balter, L. C. Myrianthopoulous, S. Vijayakumar, and H. Halpern, *The use of beam eye view volumetrics in the selection of noncoplanar radiation portals*, International Journal of Radiation Oncology: Biology, Physics, 23 (1992), pp. 153–163.

[17] Y. Chen, D. Michalski, C. Houser, and J. M. Galvin, *A deterministic iterative least-squares algorithm for beam weight optimization in conformal radiotherapy*, Physics in Medicine and Biology., 47 (2002), pp. 1647–1658.

[18] J. Y. C. Cheung, K. N. Yu, R. T. K. Ho, and C. P. Yu, *Monte carlo calculated output factors of a leksell gamma knife unit*, Physics, Medicine and Biology, 44 (1999), pp. N247–N249.

[19] B. C. J. Cho, W. H. Roa, D. Robinson, and B. Murray, *The development of target-eye-view maps for selection of coplanar or noncoplanr beams in conformal radiotherapy treatment planning*, Medical Physics, 26 (1999), pp. 2367–2372.

[20] P. S. Cho, H. G. Kuterdem, and R. J. Marks, *A spherical dose model*

for radiosurgery treatment planning, Physics in Medicine and Biology, 43 (1998), pp. 3145–3148.

[21] R. E. M. Cooper, *A gradient method of optimizing external-beam radiotherapy treatment plans*, Radiology, 128 (1978), pp. 235–243.

[22] S. M. Crooks, A. Pugachev, C. King, and L. Xing, *Examination of the effect of increasing the number of radiation beams on a radiation treatment plan*, Physics in Medicine and Biology, 47 (2002), pp. 3485–3501.

[23] J. Dai and Y. Zhu, *Selecting beam weight and wedge filter on the basis of dose gradient analysis*, Medical Physics, 27 (2001), pp. 1746–1752.

[24] J. Dai, Y. Zhu, and Q. Ji, *Optimizing beam weights and wedge filters with the concept of the super-omni wedge*, Medical Physics, 27 (2000), pp. 2757–2762.

[25] A. Drud, *CONOPT: A GRG code for large sparse dynamic nonlinear optimization problems*, Mathematical Programming, 31 (1985), pp. 153–191.

[26] M. C. Ferris, *MATLAB and GAMS interfacing optimization and visualization soft ware*, Technical Report Mathematical Programming Technical Report 98-19, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, 1998.

[27] M. C. Ferris, J. Lim, and D. M. Shepard, *An optimization approach for radiosurgery treatment planning*, SIAM Journal On Optimization, Forthcoming, (2002).

[28] ——, *Radiosurgery optimization via nonlinear programming*, Annals of Operations Research, Forthcoming, (2002).

[29] M. C. Ferris, R. R. Meyer, and W. D'Souza, *Radiation treatment planning: Mixed integer programming formulations and approaches*, Optimization Technical Report 02-08, University of Wisconsin, (2002).

[30] M. C. Ferris and D. M. Shepard, *Optimization of Gamma Knife radiosurgery*, in Discrete Mathematical Problems with Medical Applications, D.-Z. Du, P. Pardalos, and J. Wang, eds., vol. 55 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, 2000, pp. 27–44.

[31] GAMS, *GAMS/CPLEX 7.5 User Notes*, GAMS Development Corporation: http://www.gams.com/docs/, 2002.

[32] J. C. Ganz, *Gamma Knife Surgery*, Springer-Verlag Wien, Austria, 1997.

[33] Y. Ge and J. M. Fitzpatrick, *On the generation of skeletons from discrete euclidean distance maps*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 18 (1996), pp. 1055–1066.

[34] M. Goitein, M. Abrams, S. Rowell, H. Pollari, and J. Wiles, *Multi-dimensional treatment planning: Ii. beam's eye-view, back projection, and projection through ct sections*, International Journal of Radiation Oncology: Biology, Physics, 9 (1983), pp. 789–797.

[35] P. Gokhale, E. Hussein, and N. Kulkarni, *Determination of beam*

orientation in radiotherapy planning, Medical Physics, 21 (1994), pp. 393–400.

[36] O. Hass, K. J. Burnham, and J. A. Mills, *Optimization of beam orientation in radiotherapy using planar geometry*, Physics in Medicine and Biology, 43 (1998), pp. 2179–2193.

[37] L. Hong, A. Kaufman, Y. Wei, A. Viswambharn, M. Wax, and Z. Liang, *3D virtual colonoscopy*, Proceedings of 1995 Symposium on Biomedical Visualization Atlanta Ga., (1995), pp. 26–32.

[38] ILOG CPLEX Division, *CPLEX Optimizer*, 889 Alder Avenue, Incline Village, Nevada. `http://www.cplex.com/`.

[39] Intensity Modulated Radiation Therapy Collaborative Working Group, *Intensity-modulated radiotherapy: Current status and issues of interest*, International Journal of Radiation Oncology: Biology, Physics, 51 (2001), pp. 880–914.

[40] T. J. Jordan and P. C. Williams, *The design and performance characteristics of a multileaf collimator*, Physics in Medicine and Biology., 39 (1994), pp. 231–251.

[41] H. M. Kooy, L. A. Nedzi, J. S. Loeffler, E. Alexander, C. Cheng, E. Mannarino, E. Holupka, and R. Siddon, *Treatment planning for streotactic radiosurgery of intra-cranial lesions*, International Journal of Radiation Oncology, Biology and Physics, 21 (1991), pp. 683–693.

[42] M. Langer, R. Brown, M. Urie, J. Leong, M. Stracher, and J. Shapiro, *Large-scale optimization of beam-weights under dose-volume restrictions*, International Journal of Radiation Oncology: Biology, Physics, 18 (1990), pp. 887–893.

[43] M. Langer and J. Leong, *Optimization of beam weights under dose-volume restriction*, International Journal of Radiation Oncology: Biology, Physics, 13 (1987), pp. 1255–1260.

[44] E. K. Lee, T. Fox, and I. Crocker, *Optimization of radiosurgery treatment planning via mixed integer programming*, Medical Physics, 27 (2000), pp. 995–1004.

[45] J. Legras, B. Legras, and J. P. Lambert, *Software for linear and non-linear optimization in external radiotherapy*, Computer Programs in Biomedicine, 15 (1982), pp. 233–242.

[46] F. Leymarie and M. D. Levine, *Fast raster scan distance propagation on the discrete rectangular lattice*, Computer Vision Graphics Image Processing, 55 (1992), pp. 84–94.

[47] J. G. Li, A. L. Boyer, and L. Xing, *Clinical implementation of wedge filter optimization in three-dimensional radiotherapy treatment planning*, Radiotherapy and Oncology, 53 (1999), pp. 257–264.

[48] J. Lim, M. C. Ferris, S. J. Wright, D. M. Shepard, and M. A. Earl, *An optimization framework for conformation radiation therapy*, Working Paper, University of Wisconsin - Madison, (October 2002).

[49] L. Luo, H. Shu, W. Yu, Y. Yan, X. Bao, and Y. Fu, *Optimizing computerized treatment planning for the Gamma Knife by source culling*, International Journal of Radiation Oncology, Biology and Physics, 45 (1999), pp. 1339–1346.

[50] O. L. Mangasarian, *Nonlinear Programming*, SIAM, 1994.

[51] C. M. Mao and M. Sonka, *A fully parallel 3D thinning algorithm and its applications*, Computer Vision Image Understanding, 64 (1996), pp. 420–433.

[52] D. L. McShan, B. A. Fraass, and A. S. Lichter, *Full integration of the beam's eye view concept into computerized treatment planning*, International Journal of Radiation Oncology: Biology, Physics, 18 (1989), pp. 1485–1494.

[53] S. Morrill, R. Lane, J. Wong, and I. I. Rosen, *Dose-volume considerations with linear programming*, Medical Physics, 6 (1991), pp. 1201–1210.

[54] S. M. Morrill, R. G. Lane, G. Jacobson, and I. Rosen, *Treatment planning optimization using constrained simulated annealing*, Physics in Medicine and Bilology, 36 (1991), pp. 1341–1361.

[55] K. G. Murty, *Linear Programming*, John Wiley & Sons, New York, 1983.

[56] L. Myrianthopoulos, G. Chen, S. Vijayakumar, H. Halpern, D. R. Spelbring, and C. Pelizzari, *Beams eye view volumetrics - an aid in rapid treatment plan development and evaluation*, International Journal of Radiation Oncology: Biology, Physics, 23 (1992).

[57] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*, John Wiley & Sons, 1988.

[58] C. W. Niblack, P. B. Gibbons, and D. W. Capson, *Generating skeletons and centerlines from the sitance transform*, CVGIP:Graph Models Image Processing, 54 (1992), pp. 420–437.

[59] A. Pugachev and L. Xing, *Pseudo beam's-eye-view as applied to beam orientation selection in intensity-modulated radiation therapy*, International Journal of Radiation Oncology: Biology, Physics, 51 (2001), pp. 1361–1370.

[60] A. T. Redpath, B. L. Vickery, and D. H. Wright, *A new technique for radiotherapy planning using quadratic programming*, Physics in Medicine and Biology, 21 (1976), pp. 781–791.

[61] I. I. Rosen, R. Lane, S. Morrill, and J. Belli, *Treatment plan optimization using linear programming*, Medical Physics, 18 (1990), pp. 141–152.

[62] C. Rowbottom, V. Khoo, and S. Webb, *Simultaneous optimization of beam orientations and beam weights in conformal radiotherapy*, Medical Physics, 28 (2001), pp. 1696–1702.

[63] C. Rowbottom, S. Webb, and M. Oldham, *Improvements in prostate radiotherapy from the customization of beam directions*, Medical Physics, 25 (1998).

[64] ——, *Beam-orientation customization using an artificial neural network*, Physics in Medicine and Biology, 44 (1999).

[65] J. S. Rustagi, *Optimization Techniques in Statistics*, Academic Press, 1994.

[66] S. Shalev, D. Viggars, M. Carey, and P. Hahn, *The objective evaluation of alternative treatment plans 2 score functions*, International Journal of Radiation Oncology: Biology, Physics, 20 (1991), pp. 1067–1073.

[67] D. M. Shepard, M. C. Ferris, G. Olivera, and T. R. Mackie, *Optimizing the delivery of radiation to cancer patients*, SIAM Review, 41 (1999), pp. 721–744.

[68] D. M. Shepard, M. C. Ferris, R. Ove, and L. Ma, *Inverse treatment planning for Gamma Knife radiosurgery*, Medical Physics, 27 (2000), p. 12.

[69] G. W. Sherouse, *A mathematical basis for selection of wedge angle and orientation*, Medical Physics, 20 (1993), pp. 1211–1218.

[70] H. Shu, Y. Yan, L. Luo, and X. Bao, *Three dimensional optimization of treatment planning for gamma unit treatment system*, Medical Physics, 25 (1998), pp. 2352–2357.

[71] H. Z. Shu, Y. L. Yan, X. D. Bao, Y. Fu, and L. M. Luo, *Treatment planning optimization by quasi-newton and simulated annealing methods for gamma unit treatment system*, Physics, Medicine and Biology, 43 (1998), pp. 2795–2805.

[72] S. Soderstrom, A. Gustafsson, and A. Brahme, *Few-field radiation-therapy optimization in the phase-space of complication-free tumor central,*

International Journal of imaging systems and technology, 6 (1995), pp. 91–103.

[73] G. Starkschall, *A constrained least-squares optimization method for external beam radiation therapy treatment planning*, Medical Physics, 11 (1984), pp. 659–665.

[74] R. A. Stone, V. Smith, and L. Verhey, *Inverse planning for the Gamma Knife*, Medical Physics, 20 (1993), p. 865.

[75] J. Tervo and P. Kolmonen, *A model for the control of a multileaf collimator in radiation therapy treatment planning*, Inverse Problems, 16 (2000), pp. 1875–1895.

[76] P. van der Eijk, *GDX facilities in GAMS*, GAMS Contributed Software, http://www.gams.com/contrib/GDXUtils.pdf, 2002.

[77] P. J. M. van Laarhoven and E. H. L. Aarts, *Simulated Annealing: Theory and Applications*, Kluwer Academic Publishers, 1989.

[78] V.Chvátal, *Linear Programming*, W.H. Freeman and Company, New York, 1983.

[79] L. Walton, C. K. Bomford, and D. Ramsden, *The sheffield streotactic radiosurgeyr unit: Physical characteristics and principles of operation*, The British Journal of Radiology, 3 (1987), pp. 897–960.

[80] J. Wang, *Packing of unequal spheres and automated radiosurgical treatment planning*, Journal of Combinatorial Optimization, 3 (1999), pp. 453–463.

[81] S. Webb, *Optimization of conformal radiotherapy dose distributions by simulated annealing*, Physics in Medicine and Biology, 34 (1989), pp. 1349–1369.

[82] ——, *Optimization by simulated annealing of three-dimensional, conformal treatment planning for radiation fields defined by a multileaf collimator*, Physics in Medicine and Biology, 36 (1991), pp. 1201–1226.

[83] ——, *Optimization by simulated annealing of three-dimensional, conformal treatment planning for radiation fields defined by a multileaf collimator: Ii. inclusion of the two-dimensional modulation of the x-ray intensity*, Physics in Medicine and Biology, 37 (1992), p. 1992.

[84] ——, *The Physics of Conformal Radiotherapy*, Institute of Physics Publishing, 1997.

[85] ——, *Configuration options for intensity-modulated radiation therapy using multiple static fields shaped by a multileaf collimator*, Physics in Medicine and Biology., 43 (1998), pp. 241–260.

[86] A. Wu, *Physics and dosimetry of the gamma knife*, Neurosurgery Clinics of North America, 3 (1992), pp. 35–50.

[87] Q. J. Wu and J. D. Bourland, *Morphology-guided radiosurgery treatment planning and optimization for multiple isocenters*, Medical Physics, 26 (1999), pp. 2151–2160.

[88] Q. R. Wu, *Treatment planning optimization for gamma unit radiosurgery,PhD thesis*, The Mayo Graduate School, 1996.

[89] X. Wu and Y. Zhu, *A global optimization method for three-dimensional conformal radiotherapy treatment planning*, Physics in Medicine and Biology, 46 (2001), pp. 109–119.

[90] L. Xiaowei and Z. Chunxiang, *Simulation of dose distribution by the leksell gamma unit*, Physics, Medicine and Biology, 44 (1999), pp. 441–445.

[91] L. Xing, R. Hamilton, C. Pelizzari, and G. Chen, *A three-dimensional algorithm for optimizing beam weights and wedge filters*, Medical Physics, 25 (1998), pp. 1858–1865.

[92] L. Xing, C. Pelizzari, F. Kuchnir, and G. Chen, *Optimization of relative weights and wedge angles in treatment planning*, Medical Physics, 24 (1997), pp. 215–221.

[93] Y. Yan, H. Shu, and X. Bao, *Clinical treatment planning optimization by Powell's method for Gamma unit treatmen system*, International Journal of Radiation Oncology, Biology and Physics, 39 (1997), pp. 247–254.

[94] Y. Yan, H. Shu, X. Bao, L. Luo, and Y. Bai, *Clinical treatment planning optimization by powell's method for gamma unit treatment system*, International Journal of Radiation Oncology, Biology and Physics, 39 (1997), pp. 247–254.

[95] B. S. Yandell, *Practical Data Analysis for Designed Experiments*, Chapman and Hall, London, 1997.

[96] Y. Zhou, A. Kaufman, and A. W.Toga, *Three dimensional skelton and*

centerline generation based on an approximate minimum distance field, Visual Computers, 14 (1998), pp. 303–314.

[97] Y. Zhou and A. W. Toga, *Efficient skeletonization of volumetric objects*, IEEE Transaction on Visualization and Computer Graphics, 5 (1999), pp. 196–209.