Computational Enhancements to Fluence Map Optimization for Total Marrow Irradiation Using IMRT

Velibor V. Mišić

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

BACHELOR OF APPLIED SCIENCE

Supervisor: Professor Dionne M. Aleman

Department of Mechanical and Industrial Engineering
University of Toronto

March 25, 2010

Abstract

Bone marrow transplants are frequently used to treat diseases such as blood and bone marrow cancers. To perform a bone marrow transplant, it is necessary to eliminate the patient's existing bone marrow. In practice, this is most often achieved by irradiating the patient's entire body – a process known as total body irradiation (TBI) – which frequently results in radiation related side effects. A safer alternative to TBI is total marrow irradiation (TMI), which is concerned with irradiating the bone marrow and avoiding healthy tissue as much as possible.

In prior work, we considered the possibility of using intensity modulated radiation therapy (IMRT) for the purpose of TMI and specifically, we developed algorithms to solve a fundamental problem in IMRT treatment planning known as the beam orientation optimization (BOO) problem. In this study, we consider the fluence map optimization (FMO) problem which is at the heart of the BOO problem and consider several methods of improving FMO solution speed and quality. In particular, we consider different line search strategies for the projected gradient algorithm which solves the FMO problem, different warm-start techniques for speeding up FMO evaluation in a BOO setting, and algorithms for parallelized objective function and gradient evaluation to improve the speed of FMO when a large number of beams is used.

We present results from our tests of different line search strategies and different warm start methods. We also report results from using our parallelism-enhanced FMO algorithm to solve the FMO problem with 396 beams. We discuss the quality of the different line search and warm start methods; we also discuss the quality of the 30-beam solutions we studied in this and prior work and the limitations of using IMRT for TMI. We conclude by identifying some interesting questions for future research.

Acknowledgements

I would like to begin by giving my heartfelt thanks to Professor Dionne Aleman, who I have been working with since May of 2008. Her support, energy and passion for research have been incredibly inspiring and I am extremely grateful for the journey we have taken together.

I would also like to thank Dr. Michael Sharpe at Princess Margaret Hospital, who has been collaborating with us on this project, providing us with patient data and providing his input on the quality of the treatment plans we have been obtaining.

From the morLAB, I would like to thank Hamid Ghaffari for many helpful comments and suggestions over the course of this thesis. I would also like to thank Jason Lee for many great conversations and for his help with `netu` (and also for not getting too angry when I accidentally destroyed his jobs).

Lastly, I would like thank Bata, Voja and Jelena for their unremitting love and support. To quote Blackadder, life without you would be like a broken pencil: pointless.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Bone marrow transplantation, or hematopoietic stem cell transplantation, is one method of treatment for a number of diseases of the blood and bone marrow. These include certain forms of cancer (such as leukemia and lymphoma) as well as other diseases (such as aplastic anemia and sickle cell disease). To prepare a patient for a bone marrow transplant, the patient's existing diseased bone marrow must be completely eliminated. This is typically done through a procedure known as total body irradiation (TBI). In TBI, the patient's entire body is irradiated with a single, wide-angle beam to a single prescribed level of dose (e.g. 12Gy).

This type of therapy is ineffective for two main reasons. The first reason is that healthy organs which are not affected by the disease are irradiated unnecessarily, leading to post-treatment complications. The second reason is that for the treatment to be fully effective, *all* of the patient's bone marrow must be eliminated – if any marrow still affected by the disease remains when the transplant is performed, there is a significant chance of the disease recurring. Unfortunately, the higher dose levels needed to achieve a higher elimination of the bone marrow also come with more frequent and more severe toxic effects in healthy tissue. A clinical example illustrating this tradeoff is Clift et al. [1991], where the authors show that the relapse rate for acute myeloid leukemia decreases significantly when the ... dose is increased from 12 Gy to 15.75 Gy; at the same time, "the lower relapse probability in the patients receiving the higher dose ... [does] not result in improved survival because mortality from causes other than relapse [increases]", as patients receiving the higher dose were more likely to develop radiation-related diseases such as cytomegalovirus (CMV) pneumonia and hepatic veno-occlusive disease.

One type of radiation therapy that could be more appropriate for this type of therapy is intensity modulated radiation therapy (IMRT). In conventional radiation therapy, such as the type used in TBI, each beam is of homogeneous intensity: at any point in the cross-section of the beam, the intensity is always the same. In IMRT, each beam that is used in the treatment is made up of several thousand smaller beams or *beamlets*; the intensity of each beamlet can

be controlled individually. As a result, the three-dimensional shape of the dose delivered by the beams can be conformed quite accurately to the shape of the target being irradiated (in this case, the patient's bone marrow). A method of treatment that specifically targets the bone marrow is classified as total marrow irradiation (TMI); in this case, the use of IMRT for TMI is termed intensity modulated total marrow irradiation (IM-TMI).

With regard to alternatives to TBI, there are have been some past studies into alternative modalities for TMI, including IMRT. Aydogan and Roeske [2007] and Aydogan et al. [2007] consider the delivery of TMI using IM-TMI (intensity modulated total marrow irradiation) and show that using standard commercial planning systems, large reductions in dose to organs such as the liver, kidneys and heart can potentially be achieved. Schultheiss et al. [2007] and Wong et al. [2006] consider total marrow irradiation (TMI) using helical tomotherapy, and similarly show that the dose delivered to critical organs can be significantly reduced from conventional TMI levels. In contrast to these studies, we consider the problem of TMI treatment planning within a mathematical framework that we have successfully applied to TMI previously in Mišić et al. [2009] and Mišić et al. [2010]; we also consider non-coplanar beams, which allows issues of uncertainty in the dose deposited to be avoided.

In order to use IMRT for TMI, there are two basic problems that must be solved. The first problem is concerned with how the beamlet intensities, or *fluences*, should be set for a given ensemble of beams to deliver a certain dose to the bone marrow while minimizing the dose delivered to healthy tissue. This problem is known as the *fluence map optimization* (FMO) problem; in this study, we use a convex optimization formulation of FMO that was first proposed in Romeijn et al. [2006] and has been used successfully in previous studies (see Aleman et al. [2008b] and Aleman et al. [2008a]). We solve this optimization problem using the projected gradient algorithm, a standard algorithm for solving such problems. This is the primary problem we will be considering in this study; in particular, we will be studying techniques for improving both the speed with which the FMO problem is solved, and improving the quality of the final beamlet intensities returned.

The second problem is the *beam orientation optimization* (BOO) problem, which is concerned with determining how a set of beams should be oriented for the dose to be delivered optimally. The FMO problem plays a substantial role in our formulation of the BOO problem, because the optimal objective function value of the FMO problem for a set of beams quantifies the quality of that set of beams – that is, how capable the beams are of delivering the prescribed dose to the bone marrow while minimizing the dose delivered to healthy tissue. We have already studied the BOO problem for TMI previously in Mišić et al. [2010]; although we will not directly be dealing with the BOO problem, the second part of this study is concerned with fluence map optimization in the context of BOO.

This study has three main goals. The first goal is to study the effectiveness of three different strategies for performing the line search phase of the projected gradient algorithm. The line search strategy that is used in the projected gradient algorithm is very important because it determines the amount of time

required to solve the problem as well as the quality of the end solution. In a clinical environment, it is important to be able to obtain quality solutions quickly as there is typically a very limited timeframe available for treatment plan optimization.

The second goal is to study three different techniques for warm-starting FMO evaluation in the context of the Add/Drop algorithm. These techniques are of interest because of the fact that two sets of beams which differ by only one beam will typically have very similar optimal beamlet intensities; by exploiting this similarity, it is possible to greatly increase the rate at which the FMO problem is solved in the context of the Add/Drop algorithm.

The third goal of this study is to develop parallelized algorithms for objective function and gradient evaluation in the context of FMO, and to employ these algorithms to solve the FMO problem with all possible beam orientations. The motivation behind this part of the study is to learn about the absolute quality of our solutions in Mišić et al. [2010]: it is relatively straightforward to see whether a treatment is better than conventional TBI, but it is much more difficult to determine how good the treatment is and whether a better treatment is possible. By solving the FMO problem with all possible orientations we obtain a treatment plan that is the best (or close to the best) that can be physically realized for a patient, and we thus gain insight into how good treatments with smaller numbers of beams really are.

Chapter 2

Background

2.1 Beam Orientation Optimization

The beam optimization optimization (BOO) model that we use is identical to the one used in Mišić et al. [2010]. Although the BOO problem is not the focus of this study and we do not explicitly define any new algorithms to solve it, we will need to make reference to some variables from BOO (and Add/Drop) when we discuss warm-start methods in Chapter 4, so we will briefly describe BOO and Add/Drop here.

2.1.1 Model

We use $\boldsymbol{\theta}$ to represent a **single beam orientation**, and Θ to represent a **set of beams** – that is, $\Theta = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n)$, where n is the number of beam orientations in Θ. We consider non-coplanar beam orientations obtained from rotating the gantry of the linear accelerator and translating the couch in the z-axis, so each beam orientation can be fully described by its gantry rotation and couch-z translation, i.e., $\boldsymbol{\theta} = (\theta_G, \theta_z)$, where θ_G is the gantry rotation and θ_z is the couch-z translation.

Although the linear accelerator that is used to deliver IMRT treatments is technically capable of delivering radiation from any gantry angle and any couch-z translation, we discretize the set of gantry angles and the set of couch-z translations so that the gantry angle θ_G and couch-z translation θ_z are restricted to lie in the finite sets $\mathcal{S}_G$ and $\mathcal{S}_z$ respectively. In this study, we discretize the range of gantry angles $[0°, 360°)$ at every $10°$, resulting in the set $\mathcal{S}_G = \{0, 10, 20, \ldots, 350\}$. Similarly, we discretize the range of couch-z translations $[-160$ cm$, -60$ cm$]$ at every 10cm, resulting in the set $\mathcal{S}_z = \{-160, -150, -140, \ \ldots \ , -60\}$.

We define the set of **possible beam orientations** as $\mathcal{B} = \{\boldsymbol{\theta} \mid \theta_G \in \mathcal{S}_G, \theta_z \in \mathcal{S}_z\}$. Consequently, the set of beams Θ is an element of the Cartesian product of n copies of $\mathcal{B}$ ($\Theta \in \prod_{i=1}^{n} \mathcal{B}$).

Finally, we define the function $\mathcal{F} : \prod_{i=1}^{n} \mathcal{B} \rightarrow \mathbb{R}$ as the optimal objective function value obtained from solving the fluence map optimization problem (described in the next section) using the bixels of a set of beams $\Theta \in \prod_{i=1}^{n} \mathcal{B}$. The function $\mathcal{F}$ is used as a measure of the quality of a set of beams, with lower values corresponding to better sets of beams. With these definitions we can define the BOO problem as

$$\begin{aligned} &\text{minimize} && \mathcal{F}(\Theta) \\ &\text{subject to} && \Theta \in \prod_{i=1}^{n} \mathcal{B} \end{aligned}$$

2.1.2 Add/Drop Algorithm

Due to the nonlinearity and nonconvexity of the function $\mathcal{F}$ and the lack of an analytic relationship between Θ and $\mathcal{F}(\Theta)$, this optimization problem is very difficult to solve. We thus turn to the Add/Drop algorithm, which is a type of neighborhood search heuristic for solving the BOO problem. The Add/Drop algorithm was first studied in Kumar [2005] and subsequently studied in Aleman et al. [2008a] to solve the BOO problem for site-specific treatment planning. More recently, it was applied in our previous work (see Mišić et al. [2010]) to the BOO problem for TMI using IMRT. The form of Add/Drop that is used in this study is the form used in Mišić et al. [2010]; we describe it briefly here.

We use D to represent the set of degrees of freedom of each beam. For this study, $D = \{G, z\}$, where G corresponds to the gantry angle and z to the couch-z translation.

The Add/Drop algorithm works in the following way. In iteration i, we have a current iterate, $\Theta^{(i)}$. We establish a neighborhood $\mathcal{N}_b d(\Theta^{(i)})$ around $\Theta^{(i)}$, where $\mathcal{N}_{bd}(\Theta^{(i)})$ is the set of all $\Theta \in \prod_{i=1}^{n} \mathcal{B}$ such that Θ is obtained by modifying beam $\boldsymbol{\theta}_b$ of $\Theta^{(i)}$ with respect to component d. We identify the solution $\bar{\Theta} \in \mathcal{N}_{bd}(\Theta^{(i)})$ which minimizes $\mathcal{F}$ on $\mathcal{N}_{bd}(\Theta^{(i)})$. We check whether $\mathcal{F}(\bar{\Theta}) < \mathcal{F}(\Theta^{(i)})$; if so, the solution $\bar{\Theta}$ improves our objective function value, so we set $\Theta^{(i+1)} = \bar{\Theta}$. Otherwise, we consider a different pair $(b, d) \in \{1, \ldots, n\} \times D$, which furnishes a different type of neighborhood around $\Theta^{(i)}$.

The algorithm terminates when all $\Theta \in \bigcup_{b=1}^{n} \bigcup_{d \in D} \mathcal{N}_{bd}(\Theta^{(i)})$ have been examined without improvement to the current objective function value, $\mathcal{F}(\Theta^{(i)})$. (For a more detailed description, the reader is referred to Mišić et al. [2010].)

2.1.3 Relevant Literature

The BOO problem for IMRT treatment planning has been well studied. Aleman et al. [2008a], Pugachev and Xing [2002], Stein et al. [1997], Rowbottom et al. [2001], Rowbottom et al. [1999a] and Djajaputra et al. [2003] all apply some form of simulated annealing to BOO. Genetic algorithms have similarly been applied to the BOO problem - see Hou et al. [2003], Li et al. [2004], Haas et al. [1999], Ezzell [1996] and Schreibmann and Xing [2005]. Other metaheuristic approaches that have been applied include evolutionary algorithms (see

Schreibmann et al. [2004]), neural networks (see Rowbottom et al. [1999b]) and particle swarm optimization (see Li et al. [2005]). Some studies, such as Haas et al. [1999], Schreibmann et al. [2003] and Potrebko et al. [2008], approach BOO from a geometric standpoint and consider the area/volume of intersection of the beams. Gaede et al. [2004] optimize beam orientations by scrutinizing the relative change in existing beamlet intensities when a new beam is added to a current set of beams. Craft [2007] uses a gradient search algorithm where the gradient is constructed using linear programming (LP) duality theory. Söderström and Brahme [1992] approach beam selection by considering the integral of the low frequency portion of the Fourier transform of the optimal beam profile for each beam, and also by considering the entropy of the optimal beam profiles. A related information theoretic approach which has also been applied to BOO is vector quantization, which is employed by Ehrgott et al. [2008], Acosta et al. [2008] and Reese [2005]. Some studies have taken a more graph-theoretic approach to BOO: Ehrgott et al. [2008] approaching BOO as a set cover problem, while Reese [2005] and Lim et al. [2009] connect BOO to the p-median problem. D'Souza et al. [2008] use the method of nested partitions, where the solution space for the BOO problem is successively divided into smaller subregions iteration by iteration, with promising subregions being explored more extensively than less promising one.

Another heuristic idea that has also been studied for BOO is that of beam's eye view (BEV), which considers what portions of different structures are "seen" by different beams (see Pugachev and Xing [2002] and Pugachev and Xing [2001]). Related to this idea is that of target's eye view (TEV) where, for each critical structure, the beam's eye view from every possible orientation is considered, allowing each beam to be scored differently depending on how much overlap there is between the critical and target structures (see Cho et al. [1999]). Gokhale et al. [1994] consider the attenuation of radiation emanating from a hypothetical source placed at the target and select the paths of least resistance (resulting in the lowest degree of attenuation). In addition to heuristic methods, exhaustive search strategies have also been applied to BOO – see Wang et al. [2004], Liu et al. [2006], Stein et al. [1997] and Meedt et al. [2003]. BOO has also been extensively studied within the framework of integer and mixed-integer programming: see Lee et al. [2006], Ehrgott and Johnston [2003], D'Souza et al. [2004], and Lim et al. [2008]).

2.2 Fluence Map Optimization

2.2.1 Model

Like the BOO model that we use, the fluence map optimization (FMO) model that we use is identical to the one used in Mišić et al. [2010]; we briefly describe it here for later reference.

Suppose we are given a set of beams Θ. The main decision variable of the FMO problem is the variable x_i, which is the intensity or fluence of beamlet i;

the index i belongs to the set B_Θ, which is the set of bixel indices of all beams in Θ.

We let S represent the set of critical structures and T represent the set of target structures. We let z_{js} represent the total dose received by voxel j in structure s. The index j ranges from 1 to v_s, where v_s is the number of voxels in structure s. We let D_{ijs} represent the dose deposition coefficient of beamlet i in voxel j of structure s – that is, the dose delivered to voxel j of structure s by beamlet i at unit intensity.

If we make the assumption that the dose to each voxel in each structure is additive, then we can relate the x_i values to the z_{js} values by defining

$$z_{js} = \sum_{i \in B_\Theta} D_{ijs} x_i, \tag{2.1}$$

i.e. the sum of all of the beamlet intensities of the beams weighted by their corresponding D_{ijs} values.

With the dose defined, we are now interested in assigning a value to how well the actual dose in each voxel matches the dose we desire in that voxel. We let $F_{js}(z_{js})$ be the penalty function associated with voxel j in structure s, and we define it as

$$F_{js}(z_{js}) = \frac{1}{v_s}\left[\underline{w}_s \left(T_s - z_{js}\right)_+^{\underline{p}_s} + \overline{w}_s \left(z_{js} - T_s\right)_+^{\overline{p}_s}\right], \tag{2.2}$$

where T_s is the target dose for structure s, $\underline{w}_s$ and $\overline{w}_s$ are the coefficients for underdosing and overdosing of structure s, $\underline{p}_s$ and $\overline{p}_s$ are the powers for underdosing and overdosing of structure s, and the function $(\cdot)_+$ is defined as $(\cdot)_+ = \max(\cdot, 0)$. Essentially, the value $F_{js}(z_{js})$ represents the weighted deviation of the dose in z_{js} from the target dose of structure s. To ensure the resulting optimization problem is convex, we set $\underline{w}_s, \overline{w}_s > 0$ and $\underline{p}_s, \overline{p}_s > 1$.

Having defined F_{js}, we define our optimization problem as

$$\begin{aligned}
\text{minimize} \quad & F(x) = \sum_{s \in S} \sum_{j=1}^{v_s} F_{js}(z_{js}) \\
\text{subject to} \quad & z_{js} = \sum_{i \in B_\Theta} D_{ijs} x_i, \qquad \forall j \in \{1, \ldots, v_s\},\ \forall s \in S \cup T \\
& x_i \geq 0, \qquad \forall i \in B_\Theta
\end{aligned}$$

We solve this optimization problem using the projected gradient algorithm, which is described more fully in Section 3.2.

2.2.2 Relevant Literature

Many other formulations of the FMO problem exist in the literature. Some studies formulate the problem as a MIP problem (see Preciado-Walters et al.

[2004], Lee et al. [2003, 2006] and Lim et al. [2008]), while others consider FMO as a multi-criteria optimization problem (see Küfer et al. [2003], Hamacher and Küfer [2002] and Lahanas et al. [2003]). With regard to convex optimization formulations, the FMO problem has previously been modelled as an LP (see Craft [2007], Romeijn et al. [2006], Hamacher and Küfer [2002], Küfer et al. [2003] and Lim et al. [2008]), second-order cone program (SOCP) (see Zinchenko et al. [2008]), weighted least-squares problem (see Gaede et al. [2004]) and as a generic convex optimization problem (see Choi and Deasy [2002]).

The projected gradient algorithm which we will discuss in the next chapter has been widely used in radiotherapy treatment planning. For examples of different types of projected gradient techniques that have been applied to radiotherapy treatment planning, see Trofimov et al. [2005], Men et al. [2009], Thieke et al. [2003] and Ólafsson et al. [2005].

2.3 Dose Volume Histograms

The objective function F defined in the previous section gives us the total weighted deviation of the actual dose deposited in the voxels of the patient geometry from the desired overall dose, which constitutes one measure of the quality of a collection of beamlet intensities. However, we cannot directly determine from a given value of F what the resulting dose distribution in the patient geometry will be. A high value, for instance, could mean that the bone marrow is being heavily underdosed, but could just as easily mean that some important organs are being overdosed; furthermore, we do not really know a priori whether a value is "high" or "low" in an absolute sense.

It is possible to represent the dose distribution delivered by a treatment to different structures through what is known as a dose volume histogram (DVH). A DVH is a graph for a particular structure which tells us, for each level of dose, what percentage of the structure's volume receives that amount of dose or higher. In the case of a critical organ, this graph can then be evaluated using medical guidelines and past experience to determine whether the structure will be spared or not; in the case of the bone marrow, the graph can be evaluated to determine whether or not the bone marrow receives enough dose to ensure the patient does not relapse after treatment.

A set of example DVHs for a TMI treatment is shown in Figure 2.1. In this set of DVHs (and all subsequent DVHs) `hemiPTV` represents the patient's bone marrow from the hips up; we can see from its curve that roughly 93-94% of the bone marrow volume receives a dose of 12Gy or more. Our criterion for sufficient dose to the bone marrow is that 95% of the volume of the bone marrow must receive at least the target dose of 12Gy; in this case, the treatment plan falls slightly short of this requirement. We can also see that the spinal cord (represented by `cord`) begins to drop off from 100% volume at roughly the 5Gy point; this indicates that the entire volume of the spinal cord receives at least 5Gy. This meansthat the treatment plan delivers significantly more dose to the spinal cord than it does to the other organs (for any other organ, the percentage

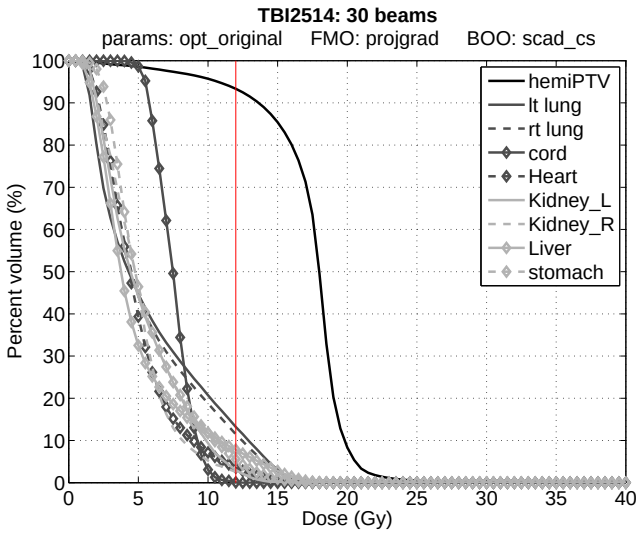


Figure 2.1: Example DVHs.

of the volume that receives at least 5Gy is less than 50%), and thus does not do as good a job of sparing the spinal cord as it does for the other organs.

For TMI, the ideal DVH for a critical structure would be a step function that drops from 100% at 0Gy to 0% immediately after, indicating that the entire organ volume receives no radiation. The ideal DVH for the bone marrow would be a step function that is 100% between 0 and 12Gy, and drops to 0% immediately after 12Gy, indicating that the entire bone marrow volume receives exactly 12Gy. Thus, to perform a quick, heuristic evaluation of a set of DVHs for a TMI treatment plan, one would look for how closely the organ curves resemble a step function at 0Gy (and how close they are in general to the (0Gy, 0%) point), and how closely the bone marrow curve resembles a step function at 12Gy. (In a conventional TBI treatment, all of the patient's structures receive 12Gy, so the DVH curve of each organ would in theory resemble a step function from 100% to 0% at 12Gy.)

Currently, due to the low level of planning required for TBI, there are no universally agreed upon criteria for TMI treatments. We use the following criteria, which we developed together with our collaborators at Princess Margaret Hospital, to evaluate treatment plans:

1. At least 95% of the bone marrow volume must receive at least 12Gy;
2. At most 20% of the bone marrow volume can receive 20Gy or more;
3. None of the bone marrow volume can receive 25Gy or more; and
4. The majority of the volume of each organ should receive below 8Gy.

The first condition ensures that the bone marrow is sufficiently eliminated before the transplant, while the last condition ensures that no organ is significantly overdosed. The second and third conditions ensure that the bone marrow is not significantly overdosed. It is important to control the degree to which the bone marrow is overdosed because at doses of 30Gy and above, fibrosis begins to occur, which prevents the newly transplanted bone marrow from successfully integrating into the patient's bones and the patient's body. We thus limit the volume receiving at least 20Gy and at least 25Gy to protect against this possibility.

Chapter 3

Line Search Strategies

3.1 Introduction

The first form of computational enhancement to fluence map optimization in the context of total marrow irradiation that we will study will be the use of different line search strategies in the projected gradient algorithm. In particular, we will study the standard backtracking line search strategy, a backtracking line search strategy with step length reduction and a forward line search strategy. The type of line search strategy is of direct interest to us in our goal of improving fluence map optimization because the line search strategy determines both the quality of the step (which ultimately affects the final FMO value obtained) and how many different step lengths are tested (and thus how many objective function evaluations occur) in each iteration of projected gradient.

We will begin by describing how the projected gradient algorithm works; we will then describe the three different line search strategies, and present results for all three resulting implementations of projected gradient.

3.2 Projected Gradient

The projected gradient algorithm is a general algorithm that can be used solving convex optimization problems, where the objective function is a convex function and the feasible set is a convex set. Given an initial solution, the algorithm determines an appropriate step length, moves by that amount in the direction of the negative gradient and projects that solution, if it is infeasible, onto the feasible set. The algorithm is formally presented as Algorithm 1.

In Algorithm 1 the function π maps x, which may be infeasible, to $\hat{x}$. For the FMO problem that we are concerned with, the only constraints placed on the variables are that each variable must be greater than or equal to zero, and less than or equal to some upper bound U on the beamlet intensities. As a

Algorithm 1 Generalized Projected Gradient

Require: Percentage change tolerance ϵ

1: Generate an initial solution $x^{(0)}$
2: Set $p = 0$.
3: **while** $p > \epsilon$ **do**
4: Generate a step length λ
5: Set $x^{(i+1)} = \pi(x^{(i)} - \lambda \nabla F(x^{(i)}))$
6: Set $p = (F(x^{(i)}) - F(x^{(i+1)}))/F(x^{(i)})$
7: **end while**

result, the coordinates of $\hat{x} = \pi(x)$ are defined as follows, for all $i \in B_\Theta$:

$$\hat{x}_i = \begin{cases} 0 & \text{if } x_i < 0, \\ x_i & \text{if } 0 \leq x_i \leq U, \\ U & \text{if } x_i > U. \end{cases} \tag{3.1}$$

To ensure that the sequence of iterates $(x^{(i)})_{i=1}^{\infty}$ converges to the global minimum of the optimization problem reasonably quickly, the step lengths should be chosen to satisfy certain conditions. For the line search variations that we study, we require that the the step length λ satisfies the *sufficient decrease* (also known as Armijo) condition,

$$F(\pi(x^{(k)} - \lambda \nabla F(x^{(k)}))) \leq F(x^{(k)}) - \frac{\sigma}{\lambda} \|x^{(k)} - \pi(x^{(k)} - \lambda \nabla F(x^{(k)}))\|^2, \tag{3.2}$$

where $\sigma \in (0, 1)$. (Details on this condition in the context of projected gradient can be found in Bertsekas [1976] and in Nocedal and Wright [2006] for unconstrained problems.)

There is another condition that could potentially also be imposed on the step length λ called the *curvature condition* (see Nocedal and Wright [2006]) to ensure that the algorithm does not take very small steps. However, this condition is undesirable from a computational standpoint because in order to verify that it is satisfied, we would also need to compute the gradient of F at every iterate $\tilde{x}$ corresponding to a potential step length λ. Furthermore, from our prior experience with the backtracking projected gradient algorithm in the context of Add/Drop in Mišić et al. [2010] (which is described in the next section and does not employ such a condition), we know that it is possible to obtain quality solutions even in the absence of such a condition.

3.2.1 Backtracking Line Search

The step length λ in Algorithm 1 can be chosen using a number of different ways. The backtracking method used here is the backtracking line search strategy of Bertsekas [1976], which is shown here as Algorithm 2. This type of line search strategy selects a step length by starting with an initial step length R, checking whether the resulting solution $\tilde{x}$ satisfies the sufficient decrease, and scaling it by a factor β (where $\beta \in (0, 1)$) until it satisfies the condition.

Algorithm 2 Projected Gradient, Backtracking Line Search

Require: Percentage change tolerance $\epsilon < 1$, initial length R_0, scale factor $\beta \in (0,1)$, constant $\sigma \in (0,1)$

Generate an initial solution $x^{(0)}$
Set $R = R_0$
Set $p = 1$
while $p > \epsilon$ **do**
  Set $\lambda = R/\|\nabla F(x^{(i)})\|$
  Set $\tilde{x} = \pi(x^{(i)} - \lambda \nabla F(x^{(i)}))$
  while $F(\tilde{x}) \leq F(x^{(i)}) - \frac{\sigma}{\lambda}\|x^{(i)} - \tilde{x}\|^2$ **do**
    Set $\lambda = \beta\lambda$
    Set $\tilde{x} = \pi(x^{(i)} - \lambda \nabla F(x^{(i)}))$
  end while
  Set $x^{(i+1)} = \pi(x^{(i)} - \lambda \nabla F(x^{(i)}))$
  Set $p = (F(x^{(i)}) - F(x^{(i+1)}))/F(x^{(i)})$
  Set $i = i + 1$
end while

3.2.2 Reduced Step Line Search

The next type of line search strategy that we consider is the reduced step line search. This form of line search is identical to the backtracking line search shown as Algorithm 2, with the slight difference that after m iterations of the outer loop, the initial step length R is set to a new value $\bar{R}$, where $\bar{R} < R_0$. This algorithm grew out of observations that the backtracking projected gradient algorithm in general takes steps as large as R_0 in the early iterations and in later iterations, takes steps that are generally of length βR_0 or smaller. By reducing the initial step length after a few outer loop iterations, we eliminate step lengths that are not likely to satisfy the sufficient decrease condition and thus reduce the overall number of line search iterations (without severely affecting the quality of the end solution).

3.2.3 Forward Line Search

In the backtracking line search implementation of projected gradient, we start from an initial step length R and in each line search iteration that the step length λ does not meet the sufficient decrease condition, it is scaled down by a factor of β. In a similar way we can define a forward line search version of the projected gradient algorithm, where we start from an initial step length R and keep scaling the step length λ up by a factor of γ ($\gamma \in (1, \infty)$) to just before the point where λ no longer satisfies the sufficient decrease condition. The procedure is formally defined as Algorithm 3.

The reasoning behind scaling the step length up to just before the point where it no longer yields a sufficient decrease in F can be explained as follows. To ensure that the algorithm finds a satisfactory step length using a forward line

Algorithm 3 Projected Gradient, Forward Line Search

Require: Percentage change tolerance ϵ, initial length R_0, scale factor γ

Generate an initial solution $x^{(0)}$
Set $R = R_0$
Set $p = 0$
while $p > \epsilon$ **do**
  Set $\lambda = R/\|\nabla F(x^{(i)})\|$
  Set $\tilde{x} = \pi(x^{(i)} - \lambda \nabla F(x^{(i)}))$
  while $F(\tilde{x}) \geq F(x^{(i)}) - \frac{\sigma}{\lambda}\|x^{(i)} - \tilde{x}\|^2$ **do**
    Set $\lambda = \gamma\lambda$
    Set $\tilde{x} = \pi(x^{(i)} - \lambda \nabla F(x^{(i)}))$
  end while
  Set $x^{(i+1)} = \pi(x^{(i)} - \lambda/\gamma \cdot \nabla F(x^{(i)}))$
  Set $p = (F(x^{(i)}) - F(x^{(i+1)}))/F(x^{(i)})$
  Set $i = i + 1$
end while

search method, it makes sense that the initial step length should be sufficiently small so that the line search loop can examine a larger range of step lengths. However, small step lengths typically always meet the sufficient decrease condition, so if the initial length is very small, then the algorithm will in general always accept it, leading to a very slow rate of change in the iterate x and consequently a slow change in the objective function value $F(x)$. By increasing the step length until it no longer yields a sufficient decrease in F, we are able to force the algorithm to take larger steps and increase the rate at which the sequence $(x^{(i)})_{i=1}^n$ converges to the minimizer of F. (In the case that the first step length $\lambda = R/\|\nabla F(x^{(i)})\|$ does not satisfy the initial step length, it is also scaled down by γ. Although we do not check whether the new step length $\gamma \cdot \lambda$ satisfies the sufficient decrease condition, the initial step length of R can be chosen to be small enough that this is never an issue.)

3.3 Computational Results

Each type of projected gradient was executed on a Dell Intel Core 2 Duo laptop with a 2.4GHz CPU and 8GB of RAM on ten different randomly generated 30-beam solutions. The initial bixels of all of the beams were set to 0.3. Each variant also had the same percentage change tolerance of 0.01 and the same σ value of 0.00001. For the reduced step projected gradient implementation, m was set to 3. For both the backtracking and reduced step implementations, $R_0 = 50$ and $\beta = 0.25$. For the forward line search implementation, $R_0 = 3$ and $\gamma = 10$.

Table 3.2 shows the final objective function value attained for each starting point by each implementation of projected gradient. We can see from this table that generally the reduced step projected gradient yields the highest final

objective function values while the standard backtracking projected gradient yields the lowest, which indicates that the reduced step projected gradient yields lower quality solutions than the backtracking projected gradient. We can also see that the forward projected gradient method falls between the backtracking and reduced step methods in terms of final objective function value.

From the perspective of total computation time and number of iterations, the reduced step projected gradient exhibits the best performance. From Table 3.1, we see that on average the reduced step method terminates in fewer iterations than the backtracking and forward methods; similarly, from Table 3.3 we see the reduced step method also takes less actual time to terminate than the other two methods. Also, from Table 3.6 we see that the reduced step projected gradient method requires fewer line search iterations per outer loop iteration than both the backtracking and forward line search implementations, which conforms to what we predicted earlier for the reduced step method.

We can also see from the same table that the forward line search method on average results in a higher number of line search iterations per outer loop iteration. This is not surprising given our definition of the forward line search method; if we execute the backtracking method and the initial step length satisfies the sufficient decrease condition, it will be accepted, and the iteration will terminate with only one line search iteration, corresponding to the initial step length. (This type of scenario occurs in the early iterations, when the bixels are still highly suboptimal.) On the other hand, if we execute the forward method and the initial step length satisfies the sufficient decrease condition (which is typically the case), then the algorithm will perform at least one more line search iteration to check if the current step length can be scaled up without violating the sufficient decrease condition. Thus, the iteration will terminate with at least two line search iterations. It makes sense, therefore, for the forward line search method to (on average) have more line search iterations per outer loop iteration than the backtracking line search method.

It is natural to ask why the reduced step projected gradient method performed significantly worse than the backtracking and forward projected gradient methods in these tests. One reason why could be that the value of m chosen for these tests was too low, and that in some early iterations where the backtracking method picked a larger step length, the reduced step method picked a smaller step length, resulting in a smaller immediate decrease and affecting the decrease achieved in subsequent iterations. It is conceivable that with a higher value of m, the reduced step method would come closer to achieving the same level of final objective function value as the backtracking method (as it would essentially pick the same step lengths for iterations 1 through m). With higher values of m, however, the benefit of reducing the number of line search iterations becomes diminished, as fewer iterations will start from the smaller initial step length.

Another reason why the reduced step method performs worse is that it terminates too early – for starting points 4, 5, 8, 9 and 10, the backtracking projected gradient is able to perform six more iterations before terminating. It is conceivable that with a lower percentage tolerance, the reduced step projected gradient

Starting Point	Num. Iter.s (Backtracking)	Num. Iter.s (Reduced Step)	Num. Iter.s (Forward)
1	17	23	18
2	12	18	13
3	15	21	11
4	17	7	22
5	17	7	8
6	12	15	18
7	17	20	21
8	20	12	18
9	13	7	23
10	20	7	9
Average	16	13.7	16.1

Table 3.1: Total number of (outer loop) iterations for each type of projected gradient and each starting point. (The average indicates the average number of iterations for each type of projected gradient, taken over all starting points.)

method would continue for longer and achieve the same objective function value (or better) by the times that the backtracking projected gradient currently terminates for each of the aforementioned starting points. This is supported by the results of Tables 3.4 and 3.5, which show that for the very same starting points, the reduced step projected gradient achieves a significantly higher average reduction in objective function value per iteration and per line search iteration than either the backtracking or the forward projected gradient implementations.

3.4 Treatment Plan Quality

From the computational results in the previous section, we saw that in terms of final objective function value, the backtracking line search yielded the lowest values on average, while the reduced step line search yielded the highest on average. To attain a sense of how these differences in objective function value map to differences in treatment plan quality, the DVHs of the end solutions of the backtracking, reduced step and forward line search methods for starting point 4 are provided as Figures 3.1, 3.2 and 3.3.

From these DVHs, we can see that the significantly higher average final objective function value associated with the reduced step method translates to a very poor treatment plan; in Figure 3.2, *all* of the critical structures have median doses greater than 5Gy, with some structures (such as the oral cavity, saliva glands, bowel and bladder) receiving extremely high levels of dose. In contrast, the DVHs for the backtracking line search solution in Figure 3.1 show that, with the exception of the left kidney and stomach, the majority of the critical organs have median doses of approximately 5Gy. Furthermore, there is

Starting Point	Final Obj. Fn. Val. (Backtracking)	Final Obj. Fn. Val. (Reduced Step)	Final Obj. Fn. Val. (Forward)
1	12098.9	12230.2	12481.6
2	15275.6	13422.0	15345.9
3	12679.4	12580.5	16087.6
4	12017.1	22455.8	11873.0
5	12829.5	22690.6	18714.6
6	18492.0	17148.7	15552.0
7	12524.1	12806.9	12208.7
8	12479.2	16288.4	13781.7
9	12764.2	21507.1	11724.5
10	11822.9	23452.3	19184.0
Average	13298.3	17458.3	14695.4

Table 3.2: Final objective function value for each type of projected gradient and each starting point. (The average indicates the average final objective function value, taken over all starting points.)

Starting Point	Total Time (Backtracking)	Total Time (Reduced Step)	Total Time (Forward)
1	45.1	51.7	50.4
2	31.7	40.5	38.1
3	42.3	50.9	34.2
4	49.2	15.7	68.1
5	42.0	13.4	22.7
6	32.5	34.0	54.4
7	47.6	46.7	63.7
8	57.3	27.3	53.4
9	34.2	14.5	66.3
10	55.5	14.7	27.2
Average	43.7	30.9	47.8

Table 3.3: Total time required (in minutes) for each type of projected gradient and each starting point. (The average indicates the average total time for each type of projected gradient, taken over all starting points.)

Starting Point	Obj. Fn. Red. per It. (Backtracking)	Obj. Fn. Red. per It. (Reduced Step)	Obj. Fn. Red. per It. (Forward)
1	3197.5	2319.5	2986.9
2	4354.8	2926.8	3986.0
3	3482.7	2442.8	4535.0
4	3005.1	6273.8	2296.5
5	3296.7	7147.8	6694.7
6	4041.9	3271.7	2788.3
7	3017.9	2526.5	2430.1
8	2665.4	4257.5	2902.3
9	4407.2	7357.2	2451.2
10	2668.9	6513.1	5418.4
Average	3413.8	4503.7	3648.9

Table 3.4: Average objective function reduction per iteration for each type of projected gradient and each starting point. (The average indicates the average objective function reduction per iteration for each type of projected gradient, taken over all starting points.)

Starting Point	OF. Red. per LS. It. (Backtracking)	OF. Red. per LS. It. (Reduced Step)	OF. Red. per LS. It. (Forward)
1	1550.3	1646.1	1269.4
2	2177.4	2163.3	1594.4
3	1741.4	1744.9	1814.0
4	1457.0	5377.6	984.2
5	1648.4	6126.7	2466.5
6	2020.9	2544.6	1185.0
7	1509.0	1920.1	1034.1
8	1266.0	3345.2	1233.5
9	2203.6	6306.2	1037.0
10	1300.2	5582.7	2064.1
Average	1687.4	3675.7	1468.2

Table 3.5: Average objective function reduction per line search iteration for each type of projected gradient and each starting point. (The average indicates the average objective function reduction per line search iteration for each type of projected gradient, taken over all starting points.)

Starting Point	Avg. # LS. It. per It. (Backtracking)	Avg. # LS. It. per It. (Reduced Step)	Avg. # LS. It. per It. (Forward)
1	2.06	1.41	2.35
2	2.00	1.35	2.50
3	2.00	1.40	2.50
4	2.06	1.17	2.33
5	2.00	1.17	2.71
6	2.00	1.29	2.35
7	2.00	1.32	2.35
8	2.11	1.27	2.35
9	2.00	1.17	2.36
10	2.05	1.17	2.63
Average	2.03	1.27	2.44

Table 3.6: Average number of line search iterations per projected gradient iteration for each type of projected gradient and each starting point. (The average indicates the average number of line search iterations per projected gradient iteration for each type of projected gradient, taken over all starting points.)

a significant difference in target coverage; the backtracking line search solution guarantees that 95% of the target volume receives a dose of 11.9Gy or more, while the reduced step line search can only guarantee that 95% of the target volume receives 10.1Gy or more.

Comparing the backtracking line search solution and the forward line search solution, we can see that for this particular instance the difference in treatment plan quality is quite small; the critical organs receive roughly the same median dose in both plans, and the forward line search solution also guarantees that 95% of the target volume receives a dose of 11.9Gy or more.

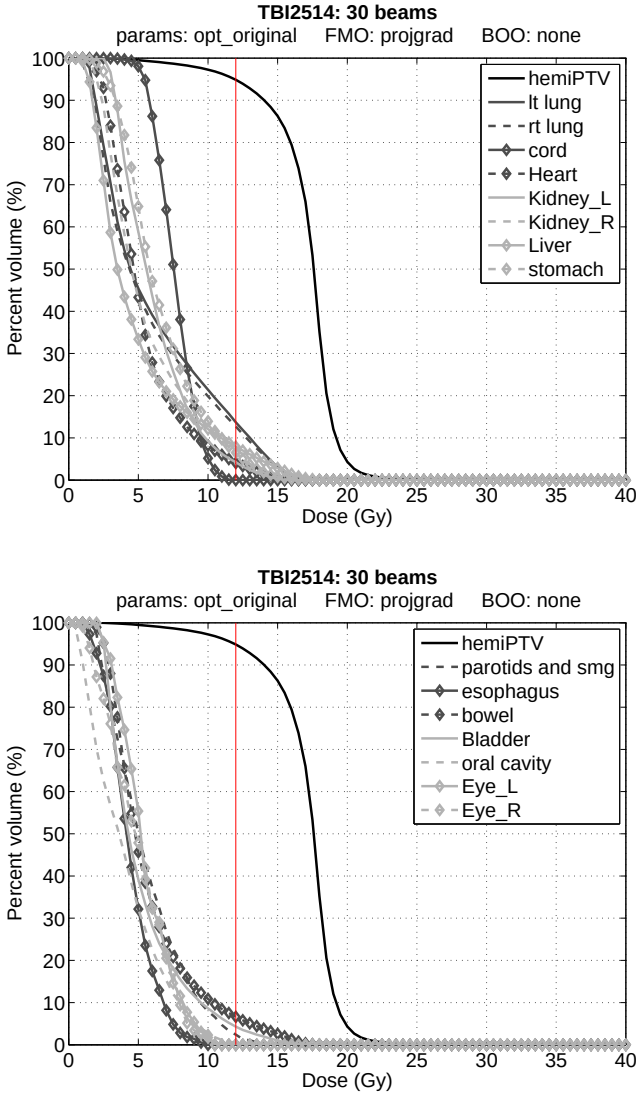
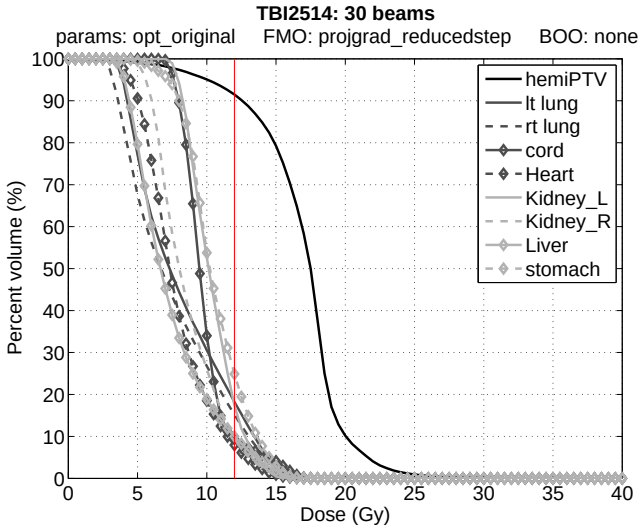


Figure 3.1: DVHs of backtracking line search solution for starting point 4.



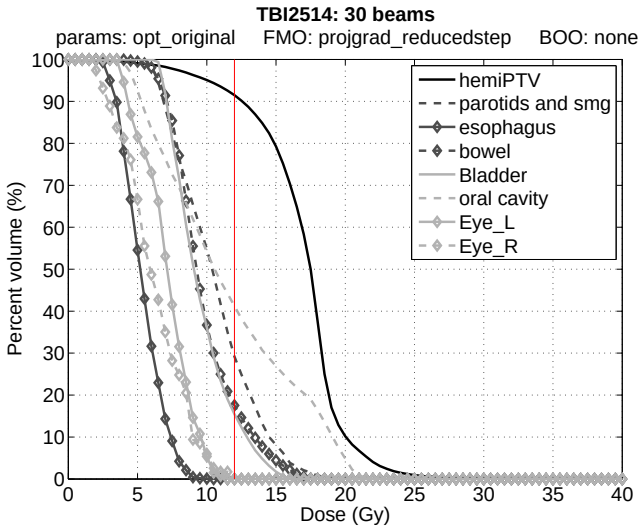


Figure 3.2: DVHs of reduced step line search solution for starting point 4.

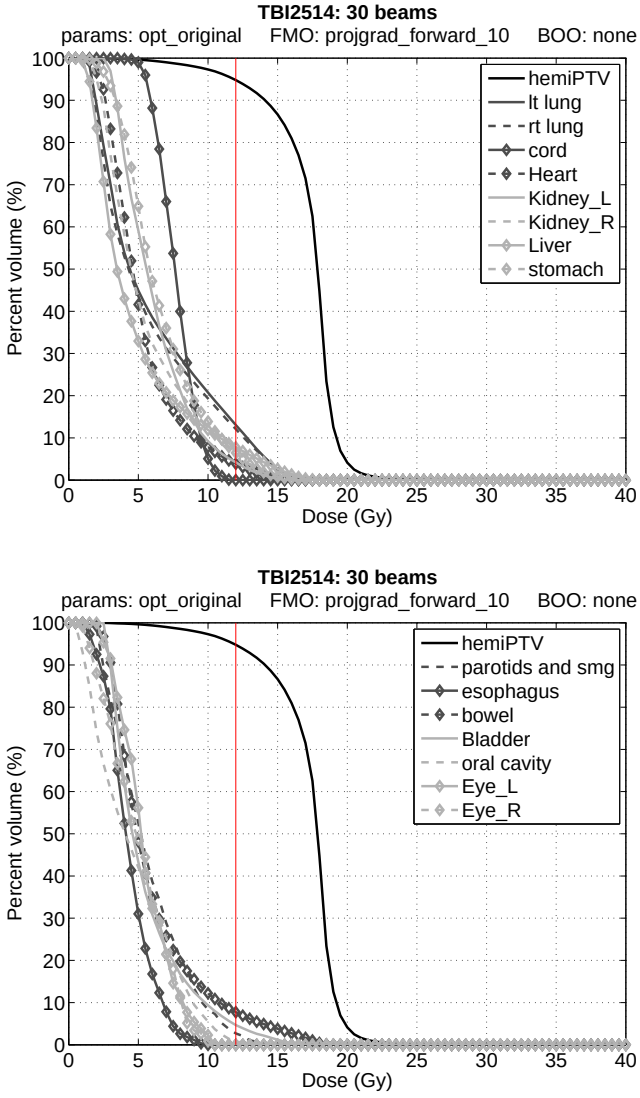


Figure 3.3: DVHs of forward line search solution for starting point 4.

Chapter 4

Warm-start Techniques

4.1 Introduction

The Add/Drop algorithm, as explained earlier, is an iterative algorithm which changes a solution (a set of beams) by modifying (or attempting to modify) a single beam in a single component in each iteration. As a result, when we compare the set of beams $\Theta^{(i)}$ in iteration i to a neighboring set of beams $\tilde{\Theta} \in \mathcal{N}_{bd}(\Theta^{(i)})$ (where $(b, d) \in \{1, \ldots, n\} \times \{G, z\}$), we find that the two solutions differ by only one beam, and the two non-matching beams differ in only one component. Since the two solutions share all but one beam, the two solutions are very similar to one another, and our intuition suggests that the optimal beamlet intensities of the common beams of $\Theta^{(i)}$ and $\tilde{\Theta}$ should also be very close to one another.

This observation about how the beamlet intensities should change within Add/Drop gives rise to a computational enhancement to FMO evaluation in the context of Add/Drop: specifically, the idea of warm-starting the FMO evaluation of a neighboring set of beams $\tilde{\Theta}$ by using the bixels of the current set of beams $\Theta^{(i)}$. We will study three different modes of FMO initialization: cold-start, warm-start using averaging and warm-start using least-squares. In the descriptions that follow we will use

- $\Theta^{(i)}$ to represent the current solution;
- $\tilde{\Theta}$ to represent a neighboring solution of $\Theta^{(i)}$, whose FMO we are interested in evaluating;
- $\boldsymbol{\theta}_k$ to represent the single beam in $\Theta^{(i)}$ which is altered;
- $\tilde{\boldsymbol{\theta}}_k$ to represent the beam in $\tilde{\Theta}$ which corresponds to $\boldsymbol{\theta}_k$ in $\Theta^{(i)}$;
- $B_{\tilde{\boldsymbol{\theta}}_j}$ to represent the set of indices of the bixels which belong to beam $\tilde{\boldsymbol{\theta}}_j$ in $\tilde{\Theta}$;
- $B_{\boldsymbol{\theta}_j}$ to represent the set of bixel indices of beam $\boldsymbol{\theta}_j$ in $\Theta^{(i)}$;

- x_i be the optimal intensity value of beamlet $i \in B_{\Theta^{(i)}}$; and
- $\tilde{x}_i$ be the optimal intensity value of beamlet $i \in B_{\tilde{\Theta}}$.

4.1.1 Cold-start

The most basic form of FMO initialization that we will consider is cold-started FMO evaluation. When an FMO evaluation is cold-started, the bixels of each beam are initialized to a pre-defined constant, and these bixels are fed into the projected gradient algorithm which calculates the FMO value for the beam. This mode of FMO evaluation does not make use of the bixels of the current solution: whenever an FMO is evaluated in this way, it is essentially calculated "from scratch".

Mathematically, if we let κ represent this pre-defined constant, we would then set

$$\tilde{x}_i = \kappa \tag{4.1}$$

for all $i \in B_{\tilde{\Theta}}$. We would then use the resulting set of bixels $\tilde{x}$ as our initial solution for FMO evaluation.

4.1.2 Warm-start using averaging

The first real warm-start method we consider is warm-started FMO evaluation using averaging. Given a neighboring solution whose FMO we are trying to calculate, we initialize the bixels of its beams as follows:

1. For each beam that the neighboring solution $\tilde{\Theta}$ has in common with the current solution $\Theta^{(i)}$, set the initial bixels of the beam to the optimal bixel values of the same beam in $\Theta^{(i)}$.
2. For beam number k where the neighboring solution $\tilde{\Theta}$ and the current solution $\Theta^{(i)}$ differ, calculate $\bar{x}$, the average of the bixels of the altered beam $\boldsymbol{\theta}_k$ in the current solution $\Theta^{(i)}$, and initialize all of the bixels of the corresponding beam $\tilde{\boldsymbol{\theta}}_k$ in $\tilde{\Theta}$ to $\bar{x}$.

As mentioned earlier, the neighboring solution and the current solution differ by only one component of one beam, so we intuitively expect the optimal bixel values of the common beams of the neighboring solution and the current solution to be very close to one another. By allowing FMO evaluation to begin from a set of bixels that should be quite close to the actual optimal set of bixels, the projected gradient algorithm we employ to solve the FMO problem should be able to converge more quickly to the optimal set of bixels. This improvement in the speed of FMO evaluation should result in a higher quality solution being found by Add/Drop, as the algorithm will be able to go through a higher number of iterations.

The step which involves averaging the bixels of the beam that is altered in the current solution and uniformly setting the bixels of the corresponding beam in the neighboring solution to this average can be justified by considering the

energy of the beams. The sum of the bixels of the beam can be thought of as a measure of the total energy of the beam; the higher the sum of the bixels, the more radiation is being delivered, and the greater the effect the beam has on the patient. (If the sum of the bixels is zero, all of the bixels are zero, and the beam is essentially not delivering any radiation in the treatment.) By setting the bixels of the corresponding beam in the neighboring solution to the average of the altered beam in the current solution, the individual bixel sums of the two beams will be very close to one another, and so the effects of the two beams on the patient should be somewhat similar. (There will certainly be some differences in their effect due to the differences in the directions of the beams.)

Mathematically, we set

$$\tilde{x}_i = x_i \tag{4.2}$$

for every $i \in B_{\boldsymbol{\theta}_j}$ and all $\boldsymbol{\theta}_j \in \Theta^{(i)} \setminus \{\boldsymbol{\theta}_k\}$. For those $i \in B_{\tilde{\boldsymbol{\theta}}_k}$, we first calculate

$$\bar{x} = \frac{1}{|B_{\boldsymbol{\theta}_k}|} \sum_{i' \in B_{\boldsymbol{\theta}_k}} x_{i'}, \tag{4.3}$$

and then set

$$\tilde{x}_i = \bar{x} \tag{4.4}$$

for all $i \in B_{\tilde{\boldsymbol{\theta}}_k}$. The vector of bixels $\tilde{x}$ is then our initial solution when the projected gradient algorithm is executed for $\tilde{\Theta}$.

4.1.3 Warm-start using least-squares

In the method of warm-starting FMO evaluation using averaging, the motivation behind averaging the bixels of the altered beam $\boldsymbol{\theta}_k$ in the current solution $\Theta^{(i)}$ and setting the bixels of the corresponding beam $\tilde{\boldsymbol{\theta}}_k$ in the neighboring solution $\tilde{\Theta}$ to this average was to attempt to get $\tilde{\boldsymbol{\theta}}_k$ to have a similar effect to $\boldsymbol{\theta}_k$. In this second warm-start method, we take this notion further by selecting the bixels of $\tilde{\boldsymbol{\theta}}_k$ so that the dose contributed by $\tilde{\boldsymbol{\theta}}_k$ is as close as possible to the contribution of $\boldsymbol{\theta}_k$ in a least-squares sense.

To make this notion rigorous, we define $z_{js}^{(k)}$ to be the dose contributed to voxel j in structure s by the bixels of $\boldsymbol{\theta}_k$; it is given by

$$z_{js}^{(k)} = \sum_{i \in B_{\boldsymbol{\theta}_k}} D_{ijs} x_i. \tag{4.5}$$

To obtain the initial bixel values of $\tilde{\boldsymbol{\theta}}_k$, that is, the values of $\tilde{x}$, we solve the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & \mathcal{Z} = \sum_{s \in S} \sum_{j=1}^{v_s} \left(\tilde{z}_{js}^{(k)} - z_{js}^{(k)} \right)^2 \\ \text{subject to} \quad & \tilde{z}_{js}^{(k)} = \sum_{i \in B_{\tilde{\boldsymbol{\theta}}_k}} D_{ijs} \tilde{x}_i, \qquad \forall s \in S, j \in \{1, \ldots, v_s\} \\ & \tilde{x}_i \geq 0, \qquad \forall i \in B_{\tilde{\boldsymbol{\theta}}_k} \end{aligned} \tag{4.6}$$

For the other beams in $\tilde{\Theta}$, we set the bixels to the corresponding bixels in $\Theta^{(i)}$ - i.e. we set

$$\tilde{x}_i = x_i \tag{4.7}$$

for every $i \in B_{\boldsymbol{\theta}_j}$ and all $\boldsymbol{\theta}_j \in \Theta^{(i)} \setminus \{\boldsymbol{\theta}_k\}$.

To see why initializing the bixels of $\tilde{\boldsymbol{\theta}}_k$ in this manner might be desirable, consider the following illustrative example. Suppose that the beamlets of $\tilde{\boldsymbol{\theta}}_k$ are able to attain a dose contribution that is very close to (or the same as) the dose contribution of the beamlets of the old beam $\boldsymbol{\theta}_k$. The two solutions $\tilde{\Theta}$ and $\Theta^{(i)}$ will then result in very similar doses in the patient (i.e. the values of z_{js} will be very close to each other). This in turn means that the objective function value associated with $\tilde{x}$ (the initial bixels of $\tilde{\Theta}$) will be very similar to, if not the same as, the objective function value associated with x (the optimal bixel values of $\Theta^{(i)}$). This places us in an advantageous position because if we begin evaluating the FMO value of $\tilde{\Theta}$ starting from $\tilde{x}$, our FMO value will be approximately as good as the FMO value of $\Theta^{(i)}$; this means that the projected gradient phase will either be short if the starting solution is already sufficiently optimal, or will yield a significantly better solution.

On the other hand, if the beamlets of the $\tilde{\boldsymbol{\theta}}_k$ are unable to attain a contribution in dose that is at all similar to the beamlets of the old beam, then that could mean two things. One scenario is that it may still be possible for the beam to improve the solution (e.g. by delivering dose to target voxels that are already covered by other beams, but doing so at a lower penalty to the objective). In this case, the projected gradient algorithm still has to iterate as usual, and we do not gain any improvement in the time needed to calculate the FMO. In contrast, if the beam is "bad" (i.e. taking out the old beam prevents the solution from hitting a lot of target voxels and forces overdosing in critical structure voxels), then once again, the projected gradient algorithm still has to iterate normally.

4.2 Computational Results

To test the effectiveness of these methods, we implemented the sequential cycling Add/Drop (SCAD) algorithm, described in Mišić et al. [2010], with each mode of FMO initialization. Both δ_G and δ_z were set to 20. For the cold-start mode of FMO initialization, κ was set to 0.3; the optimization problem (4.6) for the warm-start using least-squares mode was solved using the MATLAB command

Starting Point	Num. Iter.s (Cold-start)	Num. Iter.s (Warm-start, averaging)	Num. Iter.s (Warm-start, least-squares)
1	15	55	52
2	14	47	45
3	9	43	43
4	9	35	41
5	10	38	41
6	9	39	40
7	7	37	41
8	13	38	39
9	10	37	38
10	10	38	36
11	9	34	41
12	7	31	36
13	10	48	47
Average	10.2	40.0	41.5

Table 4.1: Number of Add/Drop iterations attained for each starting point and using each mode of FMO initialization. (The average indicates the average number of iterations for each mode over all of the starting points.)

`lsqlin` using the default parameters. These three implementations were then executed on 13 randomly chosen starting points and were allowed to run for 12 hours.

Table 4.2 shows the average time per Add/Drop iteration for each starting point and using each mode of FMO initialization, while Table 4.1 shows the number of Add/Drop iterations for each starting point. As we can see from these results, the two warm-start techniques lead to significantly lower FMO evaluation times and thus lower Add/Drop iteration times than the cold-start method; this leads to a significantly higher number of iterations in each execution of the Add/Drop algorithm with the two warm-start techniques than in each execution with the cold-start technique. The difference between the warm-start using averaging and the warm-start using least-squares modes is less significant: the two methods attain comparable rates of FMO evaluation, with the warm-start using least-squares method perhaps exhibiting slightly lower Add/Drop iteration time (as shown by the average time per iteration and average number of iterations over all of the starting points).

In addition to these results on the rate of FMO evaluation in all three Add/Drop implementations, Table 4.3 shows the final FMO value attained for each starting point, while Table 4.4 provides the difference in final FMO value for each starting point between each pair of the three modes of FMO initialization (cold-start versus warm-start using averaging, cold-start versus warm-start using least-squares, warm-start using averaging versus warm-start using least-squares). From these tables we can see that for all of the starting points tested, the cold-start method of FMO initialization leads to final FMO values that are

Starting Point	Avg. Time / Iter. (Cold-start)	Avg. Time / Iter. (Warm-start, averaging)	Avg. Time / Iter. (Warm-start, least-squares)
1	47.7	13.0	13.7
2	50.3	15.2	15.9
3	75.3	17.1	16.7
4	73.5	20.2	17.3
5	70.8	18.8	17.1
6	77.8	18.2	17.7
7	95.3	19.2	17.2
8	56.0	18.7	18.1
9	69.3	19.0	18.9
10	66.0	18.6	19.7
11	75.0	20.7	17.2
12	93.0	22.6	19.6
13	71.5	14.9	15.6
Average	70.9	18.2	17.3

Table 4.2: Average time per Add/Drop iteration in minutes for each starting point and using each mode of FMO initialization. (The average indicates the average time per iteration for each mode over all of the starting points.)

higher (and therefore worse) than the two warm-start methods.

We can also see, from Table 4.4, that the cold-start method leads to final FMO values that are on average approximately 2900 units greater than those of the warm-start using averaging method and approximately 3300 units greater than those of the warm-start using least-squares method. These differences are significant because they are quite large relative to the average of the final FMO values attained using the cold-started Add/Drop: the average cold-start final FMO value is 14906.1, which means that the warm-start using averaging and warm-start using least-square methods are able to reduce the average final FMO value obtained using the cold-start method by roughly 20 %. As we will see in the next section, this difference in numerical FMO value does translate to a significant difference in overall treatment plan quality.

When we compare the warm-start using averaging and the warm-start using least-squares methods, the results from Table 4.4 and Table 4.3 become less striking. In particular, the overall difference in final FMO value is lower: the final FMO values obtained by the warm-start using averaging A/D is on average 443.8 units higher than the final FMO values obtained by the warm-start using least-squares A/D. (Relative to the average final FMO value obtained using the warm-start using averaging method, the difference of 443.8 units translates to only a 4 % reduction in final FMO value.) Furthermore, for one starting point the final FMO value obtained using the warm-start using averaging method is actually better than the value obtained using the warm-start using least-squares method. Overall we can see that, while the warm-start using least-squares method is generally able to attain a better FMO value than the warm-start using

Starting Point	Final FMO Value (Cold-start)	Final FMO Value (Warm-start, averaging)	Final FMO Value (Warm-start, least-squares)
1	19109.3	18355.8	17302.5
2	21613.2	11756.9	11675.7
3	18416.9	13093.0	12414.4
4	12262.7	10959.1	10396.2
5	14457.3	11177.4	10738.1
6	14743.8	10668.1	10601.0
7	11184.0	10473.4	10198.5
8	11849.7	10310.6	10056.4
9	13212.4	11581.5	10868.0
10	12075.8	10730.7	10234.6
11	12587.9	10562.7	10600.8
12	11458.5	10560.7	9990.0
13	20807.5	15987.5	15371.8
Average	14906.1	12016.7	11572.9

Table 4.3: Final FMO value for each starting point and using each mode of FMO initialization.

averaging method, the improvement is not as significant as the improvement that is realized when we move from the cold-start method to either of the warm-start methods.

4.3 Treatment Plan Quality

We now compare the three modes of FMO initialization with respect to treatment plan quality. Figures 4.1, 4.2 and 4.3 show dose volume histograms (DVHs) for the final beam solution of starting point 6 using the cold-start, the warm-start using averaging and the warm-start using least-squares methods respectively. From these DVHs, we can see that the two warm-start methods are able to achieve better plan quality than the cold-start method. In particular, we can see that the two warm-start methods lead to lower median doses in the critical organs while achieving slightly higher dose in the bone marrow (indicated as "hemiPTV" in the DVHs). We can also see that the tails of a number of organ DVH curves drop off at a slower rate in the cold-start plan – for example, the curve for the liver indicates that approximately 33% of the liver volume receives 5Gy or more in the cold-start plan, while approximately only 27% receives 5Gy or more in the warm-start using averaging plan. Furthermore, we can see that a larger volume of the bone marrow receives more than 20Gy in the cold-start plan than in both the warm-start plans (which is an important consideration as extensive overdose to the bone marrow can lead to fibrosis, which can detrimentally affect the success of the subsequent bone marrow transplant). Another important difference is that both of the warm-start plans meet treatment plan criteria; the warm-start using averaging and warm-start using

Starting Point	Final FMO Diff. (CS - WS avg)	Final FMO Diff. (CS - WS lsq)	Final FMO Diff. (WS avg - WS lsq)
1	753.4	1806.7	1053.3
2	9856.3	9937.5	81.2
3	5323.9	6002.5	678.6
4	1303.6	1866.6	562.9
5	3280.0	3719.3	439.3
6	4075.7	4142.8	67.1
7	710.6	985.5	274.9
8	1539.1	1793.3	254.3
9	1630.9	2344.4	713.5
10	1345.1	1841.2	496.0
11	2025.3	1987.2	-38.1
12	897.8	1468.5	570.7
13	4820.0	5435.7	615.7
Average	2889.4	3333.2	443.8

Table 4.4: Difference in final FMO value between each pair of the three modes of FMO initialization for each starting point. ("CS" stands for "cold-start", "WS avg" stands for "warm-start using averaging" and "WS lsq" stands for "warm-start using least-squares"; the average at the bottom indicates the average difference in final FMO value for each pair of the three modes over all of the starting points.)

least-squares plans ensure that 95% of the bone marrow receives at least 12.4 and 12.5Gy respectively, while the cold-start plan can only ensure that 95% of the bone marrow receives at least 10.7Gy.

Comparing the two warm-start plans with one another, we can see that there is very little qualitative difference between them. The volume of the bone marrow that receives 12Gy or more is approximately the same for both plans, and the volume that receives 20Gy or more is also very similar for both plans. The dose distributions in the organs in both plans are very similar: the organs in the top panels of both Figures 4.2 and 4.3 (left and right lungs, spinal cord, heart, left and right kidneys, liver and stomach) all achieve median doses of 4Gy or less, and the volumes of each critical organ receiving 12Gy or more are extremely similar in both plans.

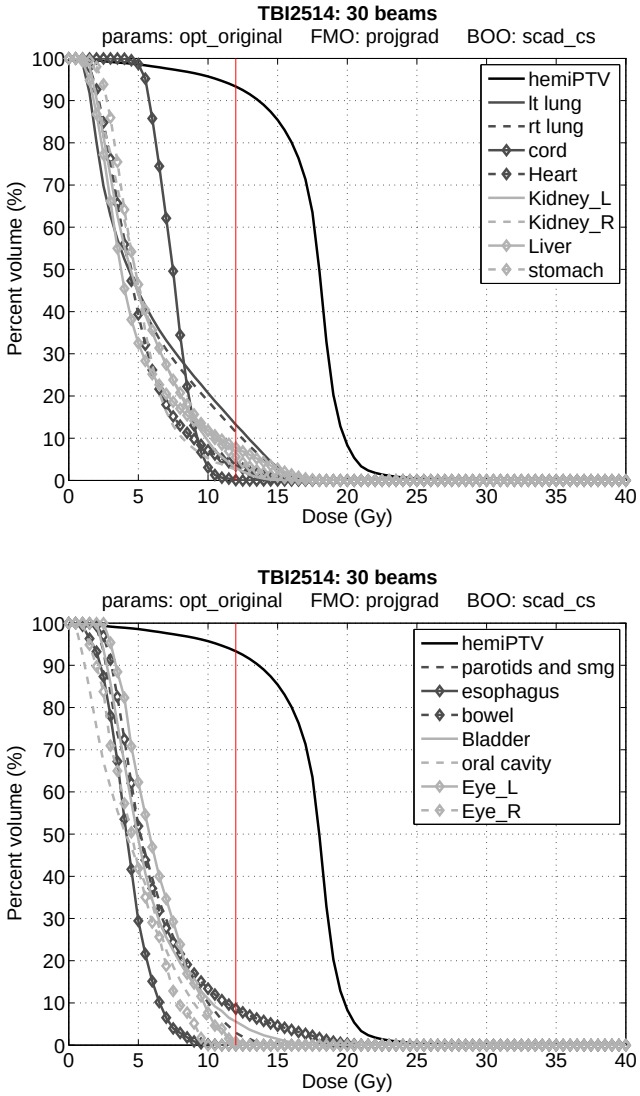


Figure 4.1: DVHs of final beam solution for starting point 6, obtained by the cold-started A/D.

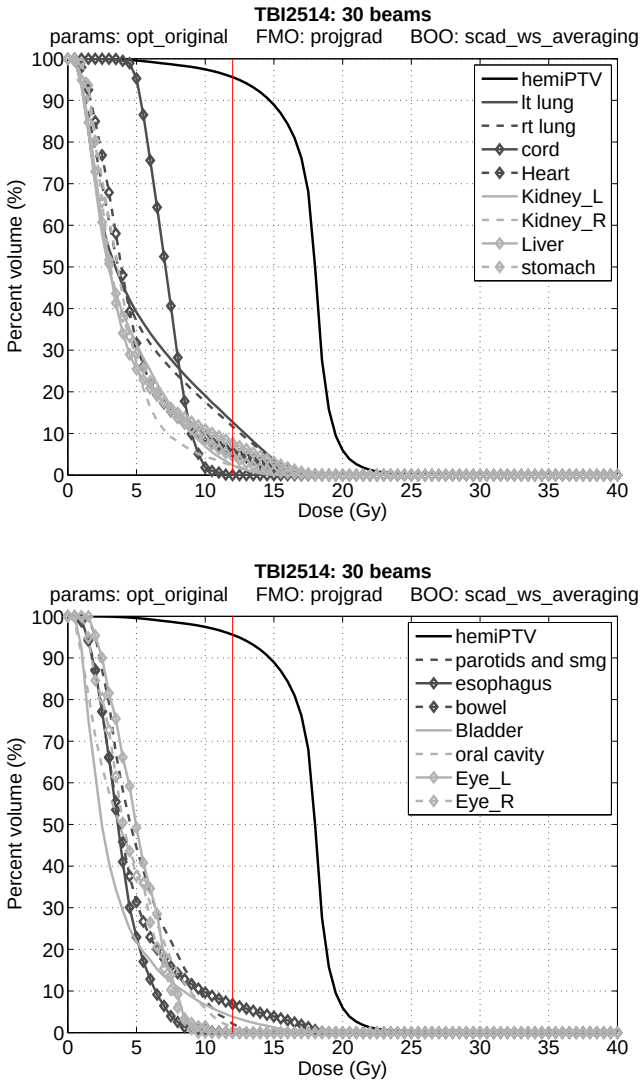


Figure 4.2: DVHs of final beam solution for starting point 6, obtained by the warm-started using averaging A/D.

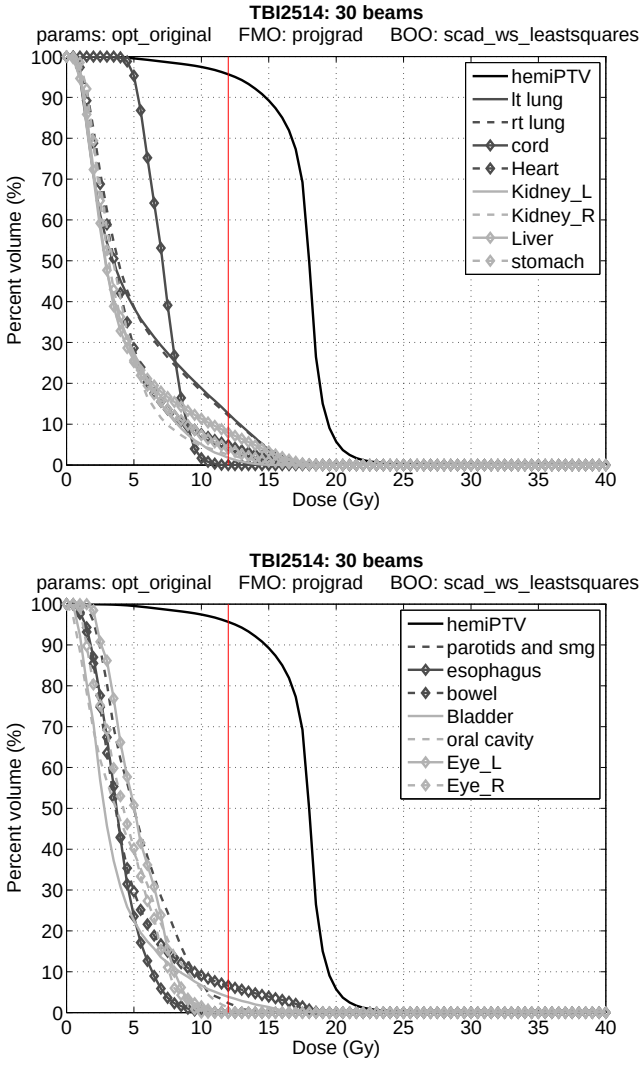


Figure 4.3: DVHs of final beam solution for starting point 6, obtained by the warm-started using least-squares A/D.

Chapter 5

Parallelized Objective Function and Gradient Evaluation

5.1 Introduction

The process of fluence map optimization is, in general, a highly computationally intensive process. The reason for this is that any gradient-based algorithm for fluence map optimization will require the calculation of the objective function F and the calculation of the gradient ∇F for a given set of bixels, which are themselves computationally expensive processes. To calculate the objective function, we must first calculate the dose to every voxel of every structure, which requires loading the dose deposition coefficients for each beam and each structure, and adding up the effects of each beamlet of each beam to every voxel in every structure. After calculating the dose, we must then go through every voxel of every structure, calculate the associated penalty contribution of the voxel, and add up these penalty contributions to get the final value of F for the bixels. To calculate the gradient, we must go through every beam and every structure, load the corresponding dose deposition coefficients and then loop through every beamlet of the selected beam and every voxel of the selected structure to add up the contribution of the voxel to the component of the gradient for each beamlet.

These computations become particularly difficult in the context of TMI for a couple of reasons. First of all, to appropriately plan for TMI a large number of structures must be contoured, which results in a large number of voxels to be considered. Second of all, to design a satisfactory TMI treatment, a large number of beams should be used, which results in more computations to calculate the dose to every voxel of the patient geometry and more computations to calculate the correspondingly larger gradient vector.

At the same time, there is also a great amount of interest in generating

plans for TMI which utilize a large number of beams. One reason why solving the FMO problem with a large number of beams (or all beams for which dose deposition coefficients are available) is interesting is that it allows us to validate our earlier plans: by solving the FMO problem with a large number of beams, we will essentially find out what the best possible treatment plan for TMI is. By knowing what the best possible treatment plan for TMI is, we will of course know whether our earlier plans are better than conventional TBI plans, but we will also know how good they are in an absolute sense.

Another reason why solutions using a large number of beams are interesting is that it is still not clear what type of radiation therapy is best suited for TMI. The plans that are considered here are all "point-and-shoot" IMRT: the gantry is rotated and the couch shifted for one beam direction, the patient is irradiated for some period of time from that direction, the beam is turned off, and the gantry is rotated and the couch shifted for the next beam direction. In contrast, another form of radiotherapy that may be more effective may be arc therapy, where the beam is continuously on as the gantry and couch are shifted (sweeping out an "arc" around the patient). This type of radiotherapy may be better suited for TMI because instead of irradiating the patient from a finite number of beam orientations, the patient is being irradiated from a continuum of beam orientations. A first approach to arc therapy could potentially involve first developing a plan with a large number of fixed beams, and then optimally determining how the gantry and couch will be shifted to transition from one beam to the next, and how the beamlets will change from one beam to the next.

In this chapter, we outline how the serial algorithms for objective function and gradient evaluation work. We then present an algorithm which can utilize multiple processors to compute the dose to every voxel in parallel and to compute the gradient in parallel. We show and discuss results for an implementation of the standard backtracking projected gradient which uses these parallelized algorithms to develop TMI plans using 396 beams.

5.2 Serial Objective Function and Gradient Evaluation

5.2.1 Notation

In the descriptions that follow, we will use

- $\mathbf{z}$ to represent the vector of voxel doses ($\mathbf{z} = [z_{js}]_{s \in S, j \in \{1,\ldots,v_s\}}$);
- $\mathbf{D}^{(k)}$ to represent the matrix of D_{ijs} values corresponding to a single beam and all of the voxels in the patient geometry (i.e. $\mathbf{D}^{(k)}$ has $\sum_{s \in S} v_s$ rows and $|B_{\boldsymbol{\theta}_k}|$ columns); and
- $\mathbf{D}_s^{(k)}$ to represent the matrix of D_{ijs} values corresponding to a single beam

and all of the voxels in a single organ s (i.e. $\mathbf{D}_s^{(k)}$ has v_s rows and $|B_{\boldsymbol{\theta}_k}|$ columns).

5.2.2 The Operations of FMO

For a column vector of beamlet intensities x corresponding to a set of n beams Θ, we are interested in three different operations:

- *The calculation of the voxel doses*: for each voxel $j \in \{1, \ldots, v_s\}$, and each structure $s \in S$, we calculate

$$z_{js} = \sum_{i \in B_\Theta} D_{ijs} x_i. \tag{5.1}$$

- *The penalization of the voxel doses*: we calculate the total penalty associated with the bixels x as

$$F(x) = \sum_{s \in S} \sum_{j=1}^{v_s} \frac{1}{v_s} \left[\underline{w}_s \left(T_s - z_{js} \right)_+^{\underline{p}_s} + \overline{w}_s \left(z_{js} - T_s \right)_+^{\overline{p}_s} \right]. \tag{5.2}$$

- *The calculation of the gradient*: we calculate

$$\nabla F = \left[\frac{\partial F}{\partial x_i} \right] \tag{5.3}$$

where

$$\begin{aligned} \frac{\partial F}{\partial x_i} = \sum_{s \in S} \sum_{j=1}^{v_s} \frac{1}{v_s} \Big[& \underline{w}_s \cdot \underline{p}_s \cdot \left(T_s - z_{js} \right)_+^{\underline{p}_s - 1} \cdot (-D_{ijs}) \\ & + \overline{w}_s \cdot \overline{p}_s \cdot \left(z_{js} - T_s \right)_+^{\overline{p}_s - 1} \cdot (+D_{ijs}) \Big] \end{aligned} \tag{5.4}$$

for each $i \in B_\Theta$.

5.2.3 Algorithm Description

If the objective function and gradient are to be evaluated serially, we use Algorithms 4, 5 and 6, which are defined below.

5.2.4 Number of Operations

When computed serially, we can see that $\mathbf{z}$, F and ∇F require different numbers of smaller operations:

- $\mathbf{z}$: To calculate one element of $\mathbf{z}$, we need to sum through all the beamlets, which constitutes $|B_\Theta|$ operations. We must do this for each voxel, so the total number of operations is on the order of $N_v \cdot |B_\Theta|$, where N_v is the total number of voxels ($N_v = \sum_{s \in S} v_s$).

Algorithm 4 Calculate voxel dose $\mathbf{z}$ (serial)

Initialize $z_{js} = 0$ for all $s \in S$, $j \in \{1, \ldots, v_s\}$
for $k = 1$ to n **do**
  for all $s \in S$ **do**
    Load $\mathbf{D}_s^{(k)}$ from hard disk
    for all $i \in B_{\boldsymbol{\theta}_k}$ **do**
      for $j = 1$ to v_s **do**
        Set $z_{js} = z_{js} + D_{ijs} x_i$
      end for
    end for
  end for
end for

Algorithm 5 Calculate $F(x)$ from voxel dose $\mathbf{z}$ (serial)

Initialize $F(x) = 0$
for all $s \in S$ **do**
  for $j = 1$ to v_s **do**
    if $z_{js} < T_s$ **then**
      Set $F(x) = F(x) + \frac{1}{v_s} \underline{w}_s \left(T_s - z_{js}\right)^{\underline{p}_s}$
    else
      Set $F(x) = F(x) + \frac{1}{v_s} \overline{w}_s \left(z_{js} - T_s\right)^{\overline{p}_s}$
    end if
  end for
end for

Algorithm 6 Calculate gradient ∇F from voxel dose $\mathbf{z}$ (serial)

Initialize $\frac{\partial F}{\partial x_i} = 0$ for each $i \in B_\Theta$
for $k = 1$ to n **do**
  for all $s \in S$ **do**
    Load $\mathbf{D}_s^{(k)}$ from hard disk
    for all $i \in B_{\boldsymbol{\theta}_k}$ **do**
      for $j = 1$ to v_s **do**
        if $z_{js} > T_s$ **then**
          Set $\frac{\partial F}{\partial x_i} = \frac{\partial F}{\partial x_i} - \frac{1}{v_s} D_{ijs} \underline{w}_s \underline{p}_s \left(T_s - z_{js}\right)^{\underline{p}_s - 1}$
        else
          Set $\frac{\partial F}{\partial x_i} = \frac{\partial F}{\partial x_i} + \frac{1}{v_s} D_{ijs} \overline{w}_s \overline{p}_s \left(z_{js} - T_s\right)^{\overline{p}_s - 1}$
        end if
      end for
    end for
  end for
end for

- F: To calculate the objective function F given $\mathbf{z}$, we must sum through every voxel. This results in a number of operations on the order of N_v.
- ∇F: To calculate one element of the gradient ∇F, we must sum through each voxels, resulting in N_v operations. We must do this for each beamlet, so the total number of operations is on the order of $N_v \cdot |B_\Theta|$.

In other words, as the number of beamlets $|B_\Theta|$ grows, the number of smaller operations required to calculate $\mathbf{z}$ and ∇F grows linearly in $|B_\Theta|$. Each beam that is considered for TMI has approximately the same number of beamlets, so $|B_\Theta|$ grows linearly with the number of beams n. On the other hand, calculating F requires N_v operations always, regardless of the number of beams that are used for the treatment. These observations suggest that the calculation of $\mathbf{z}$ and the calculation of ∇F would most benefit from parallelization, and in Section 5.3 we present an algorithms for objective function and gradient evaluation which parallelize exactly those calculations.

5.3 Parallel Objective Function and Gradient Evaluation

As we observed in the previous section, the time required to compute the objective function and gradient for a given set of bixels grows with the number of beams. In this section, we describe a parallelized algorithm for calculating the objective function and the gradient which allows us to mitigate the effect of this growth.

The general idea behind these algorithms is as follows. Suppose that we desire to calculate the objective function for a set of bixels corresponding to n beams. Suppose also that Q processors are available to us, where Q divides evenly into n. Then to calculate the total dose delivered, we can divide up the set of n beams into Q subsets, each consisting of n/Q beams. We can then, in parallel, calculate the contribution of each subset to the total dose, collect the contributions and add them up to obtain the overall dose to each voxel. (This is possible since the dose is a linear function of the bixel intensities.) To calculate the actual objective function, we penalize the overall voxel dose using Algorithm 5 as before.

To calculate the gradient for a set of bixels we proceed in a similar way. Given n and Q as described above together with the overall voxel dose, we can divide up the set of n beams into Q subsets. We can then, in parallel, calculate the piece of the gradient that corresponds to each subset (that is, calculate the partial derivative of F for each bixel in each subset of beams). After these pieces are calculated, we can collect them and concatenate them to obtain the gradient.

5.3.1 Notation

In addition to the notation defined earlier, we will use

- $x^{(k)}$ to represent the part of the x vector associated with beam $\boldsymbol{\theta}_k \in \Theta$;
- Q to represent the total number of processors available for use;
- q to refer to a particular processor in $\{1, \ldots Q\}$, and to the subset of beams of Θ assigned to processor q;
- $\mathbf{z}^{(q)}$ to represent the voxel dose contribution of the qth beam subset;
- $\mathbf{g}^{(q)}$ to represent the piece of the gradient associated with the qth beam subset;
- $\mathbf{h}^{(k)}$ to represent the piece of the gradient associated with beam $\boldsymbol{\theta}_k \in \Theta$;
- y_{js} to represent an auxilliary scalar associated with voxel j in structure s; and
- $\mathbf{y}$ to represent the column vector of y_{js} values for all $s \in S$, $j \in \{1, \ldots, v_s\}$.

5.3.2 Parallel Operations

Suppose that subset number q of Θ consists of beams $\boldsymbol{\theta}_{L_q}$, $\boldsymbol{\theta}_{L_q+1}, \ldots,$ $\boldsymbol{\theta}_{U_q-1}$, $\boldsymbol{\theta}_{U_q}$. We can then define the voxel contribution of the qth beam subset as

$$\mathbf{z}^{(q)} = \sum_{k=L_q}^{U_q} \mathbf{D}^{(k)} x^{(k)}. \tag{5.5}$$

If the subsets numbered 1 to Q comprise the entire set of beams Θ, we can calculate the overall voxel dose as

$$\mathbf{z} = \sum_{q=1}^{Q} \mathbf{z}^{(q)}, \tag{5.6}$$

i.e. summing up the voxel dose contributions of each beam subset to obtain the overall voxel doses associated with Θ.

We can express the operation of calculating the gradient ∇F in a similar way. We let $\mathbf{g}^{(q)}$ be the part of ∇F corresponding to beam subset number q, so that we can express ∇F as

$$\nabla F = \left[\mathbf{g}^{(1)} \; \mathbf{g}^{(2)} \; \ldots \; \mathbf{g}^{(Q)} \right]. \tag{5.7}$$

To define $\mathbf{g}^{(q)}$, we first let $\mathbf{y}$ be an auxilliary column vector, defined as $\mathbf{y} = [y_{js}]_{s \in S, j \in \{1,\ldots,v_s\}}$, where

$$y_{js} = \frac{1}{v_s} \left[-\underline{w}_s \underline{p}_s \left(T_s - z_{js} \right)_+^{\underline{p}_s - 1} + \overline{w}_s \overline{p}_s \left(z_{js} - T_s \right)_+^{\overline{p}_s - 1} \right]. \tag{5.8}$$

If subset number q of Θ consists beams $\boldsymbol{\theta}_{L_q}$, $\boldsymbol{\theta}_{L_q+1}, \ldots,$ $\boldsymbol{\theta}_{U_q-1}$, $\boldsymbol{\theta}_{U_q}$, we can then define

$$\mathbf{g}^{(q)} = \left[\mathbf{h}^{(L_q)} \; \mathbf{h}^{(L_q+1)} \; \ldots \; \mathbf{h}^{(U_q-1)} \; \mathbf{h}^{(U_q)} \right], \tag{5.9}$$

where

$$\mathbf{h}^{(k)} = \mathbf{y}^T \mathbf{D}^{(k)}. \tag{5.10}$$

5.3.3 Algorithm Descriptions

Given a number Q which divides evenly into the number of beams n, the objective function and gradient can be calculated in parallel by the following set of algorithms. Algorithm 7 describes how to calculate the voxel dose of the qth beam subset ($\mathbf{z}^{(q)}$), which is then used by Algorithm 8 to calculate the complete voxel dose $\mathbf{z}$. Similarly, Algorithm 9 describes how to calculate the piece of the gradient associated with the qth beam subset ($\mathbf{g}^{(q)}$), which is used by Algorithm 10 to obtain the complete gradient ∇F.

Algorithm 7 Calculate voxel dose $\mathbf{z}^{(q)}$ of beam subset q on processor q

Require: Parameter Q (total number of processors), parameter n (total number of beams), processor number q

Initialize $z_{js}^{(q)} = 0$ for all $s \in S$, $j \in \{1, \dots v_s\}$
Set $L = n(q-1)/Q + 1$, $U = nq/Q$
for $k = L$ to U **do**
    Load $\mathbf{D}^{(k)}$ from hard disk
    Set $\mathbf{z}^{(q)} = \mathbf{z}^{(q)} + \mathbf{D}^{(k)} x^{(k)}$
end for

Algorithm 8 Calculate complete voxel dose $\mathbf{z}$ (parallel)

Require: Parameter Q (total number of processors), parameter n (total number of beams)

Initialize $z_{js} = 0$ for all $s \in S$, $j \in \{1, \dots, v_s\}$
for $q = 1$ to Q **do**
    Execute Algorithm 7 on processor q with parameters Q, n
end for
Wait for Algorithm 7 to finish on processors $1, 2, \dots, Q$
for $q = 1$ to Q **do**
    Set $\mathbf{z} = \mathbf{z} + \mathbf{z}^{(q)}$
end for

5.4 Computational Results

The parallelized algorithms for dose and gradient calculation described were used in a standard implementation of the projected gradient with a backtracking line search. The resulting projected gradient algorithm was used to solve the FMO problem using all 396 beam orientations in $\mathcal{B}$ with different sets of FMO parameters. To give a sense of the size of this optimization problem, the total number of bixels that could be used to treat the patient was 977,601 while the total number of voxels was 651,410. In each case, 11 processors were used (i.e. Q was set to 11) and the percentage change tolerance was set to 10^{-6}. Each case was allowed to execute for just over 24 hours.

Algorithm 9 Calculate partial gradient $\mathbf{g}^{(q)}$ associated with beam subset q on processor q

Require: Parameter Q (total number of processors), parameter n (total number of beams), parameter $\mathbf{z}$ (total voxel dose), processor number q

Initialize $y_{js} = 0$ for all $s \in S$, $j \in \{1, \ldots, v_s\}$
for $k = 1$ to n **do**
  for all $s \in S$ **do**
    for $j = 1$ to v_s **do**
      if $z_{js} < T_s$ **then**
        Set $y_{js} = -\frac{1}{v_s}\underline{w}_s\underline{p}_s \left(T_s - z_{js}\right)^{\underline{p}_s - 1}$
      else
        Set $y_{js} = \frac{1}{v_s}\overline{w}_s\bar{p}_s \left(z_{js} - T_s\right)^{\bar{p}_s - 1}$
      end if
    end for
  end for
end for

Set $L = n(q-1)/Q + 1$, $U = nq/Q$
Initialize $g_i^{(q)} = 0$ for all $i \in \bigcup_{k=L}^{U} B_{\boldsymbol{\theta}_k}$
for $k = L$ to U **do**
  Load $\mathbf{D}^{(k)}$ from hard disk
  Set $\mathbf{h}^{(k)} = \mathbf{y}^T\mathbf{D}^{(k)}$
end for
Set $\mathbf{g}^{(q)} = \left[\mathbf{h}^{(L)}\ \mathbf{h}^{(L+1)}\ \ldots\ \mathbf{h}^{(U-1)}\ \mathbf{h}^{(U)}\right]$

Algorithm 10 Calculate complete gradient ∇F (parallel)

Require: Parameter Q (total number of processors), parameter n (total number of beams)

for all $q = 1$ to Q **do**
  Execute Algorithm 9 with parameters q, n on processor q
end for
Wait for Algorithm 9 to finish on processors $1, 2, \ldots, Q$
Set $\nabla F = \left[\mathbf{g}^{(1)}\ \mathbf{g}^{(2)}\ \ldots\ \mathbf{g}^{(Q-1)}\ \mathbf{g}^{(Q)}\right]$

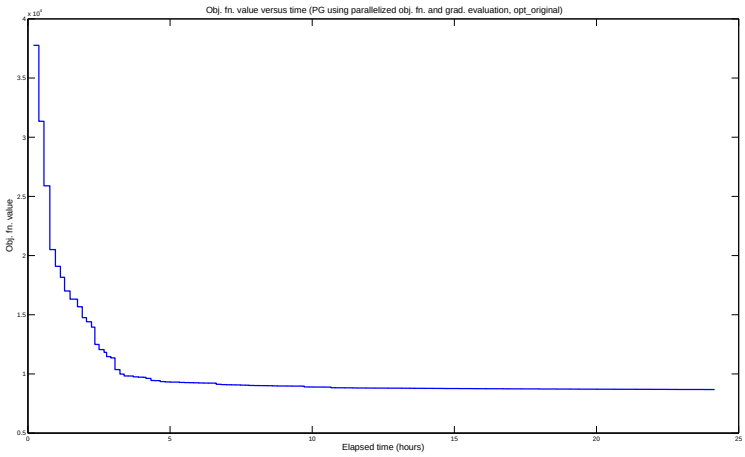


Figure 5.1: Plot of objective function value versus time for projected gradient using parallelized objective function and gradient evaluation and parameter set `opt_original`.

Three different sets of optimization parameters were used. The parameter set `opt_original` is the main set of parameters used in Mišić et al. [2010], which attempts to account for all of the critical organs. The parameter set `opt_ooc_3` is another set of parameters used in Mišić et al. [2010], which accounts for all of the organs except for the spinal cord. The parameter set `opt_atlasair_2` is equivalent to `opt_original` with the exception of the threshold dose of the bone marrow, which is reduced from its value in `opt_original`.

Figures 5.1, 5.3 and 5.5 show objective function value versus time for `opt_original`, `opt_ooc_3` and `opt_atlasair_2` respectively. We can see that the greatest amount of change in the objective function occurs at the beginning, within the first five to six hours. After the six hour mark, the rate at which the objective function value changes becomes much slower.

Similarly, Figures 5.2, 5.4 and 5.6 show the percentage change in objective function value versus time for `opt_original`, `opt_ooc_3` and `opt_atlasair_2` respectively: from these plots, we can see that the greatest amount percentage change in the objective function occurs within the first five to six hours. We can also see that for all three sets of parameters, there is a slight spike in the percentage change in objective function value shortly before the ten hour mark.

5.5 Treatment Plan Quality

To compare the treatment plan quality of the three parameter sets, three sets of DVHs were prepared:

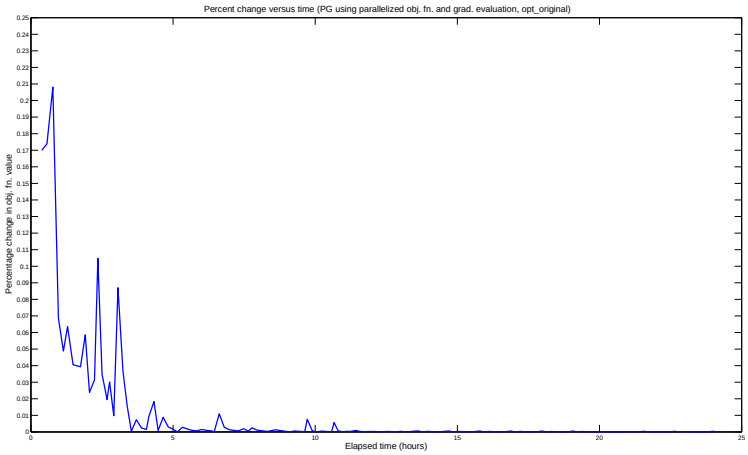


Figure 5.2: Plot of percentage change in objective function value versus time for projected gradient using parallelized objective function and gradient evaluation and parameter set `opt_original`.

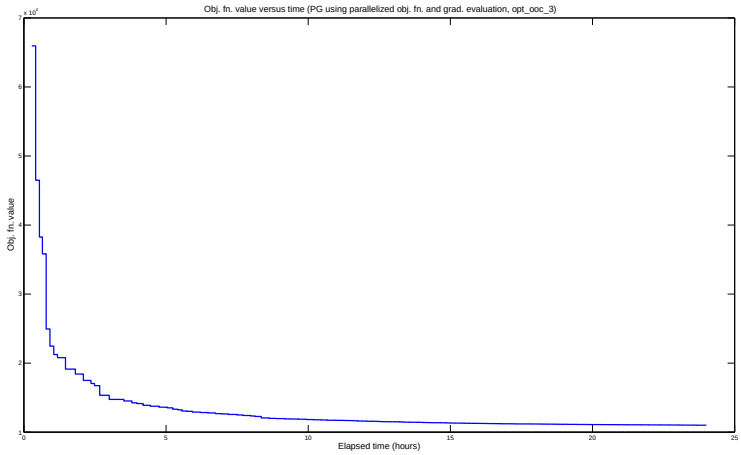


Figure 5.3: Plot of objective function value versus time for projected gradient using parallelized objective function and gradient evaluation and parameter set `opt_ooc_3`.

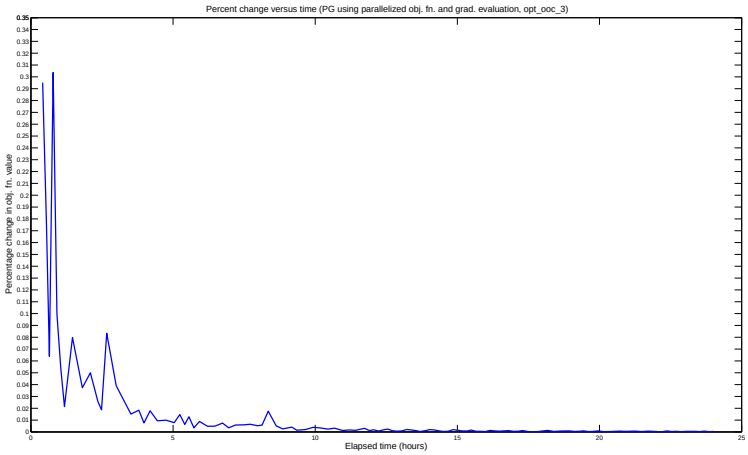


Figure 5.4: Plot of percentage change in objective function value versus time for projected gradient using parallelized objective function and gradient evaluation and parameter set `opt_ooc_3`.

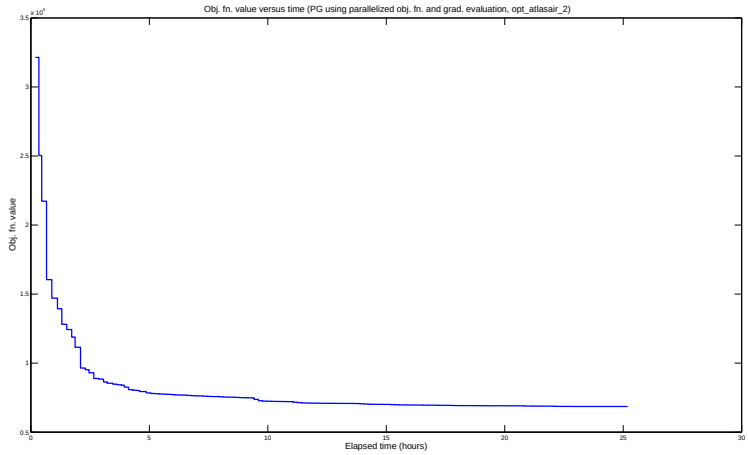


Figure 5.5: Plot of objective function value versus time for projected gradient using parallelized objective function and gradient evaluation and parameter set `opt_atlasair_2`.

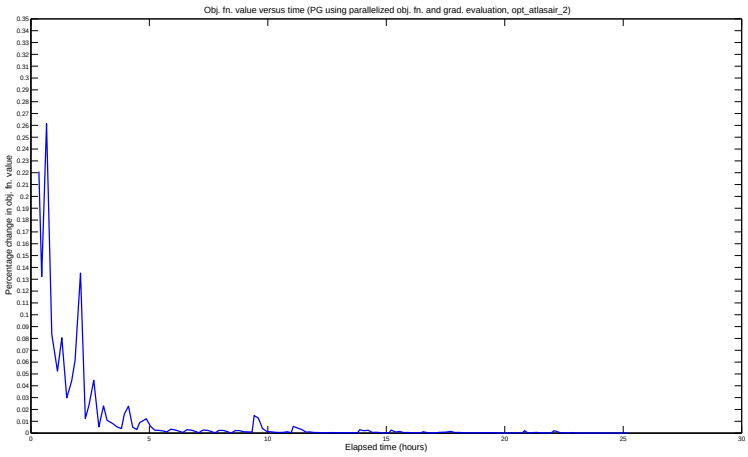


Figure 5.6: Plot of percentage change in objective function value versus time for projected gradient using parallelized objective function and gradient evaluation and parameter set `opt_atlasair_2`.

- Figures 5.8, 5.9 and 5.10 shows the 396-beam solution at six hours, twelve hours and shortly after 24 hours using the `opt_original` parameter set;
- Figures 5.11, 5.12 and 5.13 shows the 396-beam solution at six hours, twelve hours and shortly after 24 hours using the `opt_ooc_3` parameter set; and
- Figures 5.14, 5.15 and 5.16 shows the 396-beam solution at six hours, twelve hours and shortly after 24 hours using the `opt_atlasair_2` parameter set.

In addition to these DVHs corresponding to 396-beam solutions, Figure 5.7 shows a 30 beam solution obtained using the warm-start using least-squares Add/Drop and using the `opt_original` parameter set.

From these DVHs, a few things become evident. First of all, comparing the 30-beam `opt_original` solution in Figure 5.7 to the final `opt_original` solution in Figure 5.10, we can see that although the 396-beam solution is better, the DVHs of the 396-beam solution are qualitatively very similar to those of the 30 beam solution. In particular, of all of the critical organs, the spinal cord receives the most dose in both Figures 5.7 and 5.10; after the spinal cord, the lungs receive the next highest amount of dose, with just under 20% of the volumes of the left and right lungs receiving more than 10Gy. With regard to improvement, we can see that all of the critical organs (except the spinal cord) in the 30-beam plan receive a median dose of at most 5Gy; in the final 396-beam

plan using the opt_original parameter set, the same organs receive a median dose of at most 4Gy. Furthermore, there is an increase in dose to the target: in the 30 beam plan, 95% of the target volume receives more than 12.9Gy, while the same percentage of the target volume in the 396-beam plan receives more than 13.5Gy.

Second of all, we can see that the change in DVH quality for all three parameter sets is relatively small from 6 hours to 12 hours, and from 12 hours to the final plan. For opt_original, the DVH curves for the stomach, liver and spinal cord drop down approximately three to four percentage points from 6 hours to 12 hours, and approximately one percentage point from 12 hours to the final plan. The dose to the target also changes by a small amount; 95% of the target volume receives 13.1Gy or more at 6 hours, 13.4Gy or more at 12 hours and 13.5Gy in the final plan. For opt_ooc_3, we can see that from 6 hours to 12 hours and from 12 hours to the final plan, there is a drop in the median dose of the majority of the critical organs by approximately 0.5Gy. The change to the target dose is slightly more significant from 6 hours to 12 hours, however; at 6 hours 95% of the target volume receives only 11.9Gy or more, while at 12 hours 95% of the target volume receives 12.5Gy or more. For opt_atlasair_2, the change in critical organ dose is slightly greater from 6 to 12 hours than it is for the other parameter sets, with the majority of the critical organs curves being shifted down by approximately 5%; from 12 hours to the final plan, the change is not as significant. These small changes match the behaviour shown in the plots of the previous section, which show that for all three sets of parameters, most of the change in the objective function value occurs in the first five to six hours, and that the rate of change in the objective function is greatly reduced after the six hour mark.

The most important knowledge that can be taken away from these results concerns the limitations of using IMRT (and external beam radiotherapy) for the purpose of total marrow irradiation. Although we have not exhaustively searched through all possible sets of parameters, the parameter sets that are reported were the ones which were best able to satisfy our treatment plan criteria. As such, they strongly suggest the existence of several limitations of using IMRT for TMI:

1. It is not possible to achieve a median dose of under 2Gy for a majority of the critical organs while achieving adequate dose in the target;

2. It is not possible to achieve a median dose in the spinal cord below 6Gy; and

3. It is not possible to achieve a low median dose in the spinal cord while simultaneously reducing overdosing in the lungs (the volume that receives 10Gy or more) and achieving the target dose in the bone marrow.

The first and second points are clear from the DVHs of all three sets of parameters. The third fact we can infer from comparing the DVHs of opt_ooc_3 (Figure 5.13) with the DVHs of opt_original and opt_atlasair_2 (Figures 5.10

and 5.16 respectively). With opt_ooc_3, we can see that the overdose in the lungs is fairly tame (with only roughly 10% of the volume of each lung receiving more than 10Gy), but the spinal cord receives a very high amount of dose (median dose is higher than the target dose of 12Gy). On the other hand, with opt_atlasair_2, the spinal cord receives a lower amount of dose (median dose of approximately 6Gy), but a higher volume of the lungs receives 10Gy or more, and the corresponding DVH curves for the lungs drop off at a much slower rate than they do for opt_ooc_3.

It is interesting to observe that using the opt_original parameter set, the 396-beam solution obtained after 6 hours (Figure 5.8) is just as good, if not slightly better, than the benchmark 30 beam solution in Figure 5.7 with respect to both target dose and organ sparing. (Similarly, the opt_atlasair_2 solution obtained after 6 hours also compares very favorably to the benchmark 30 beam solution, although it is only able to guarantee that 95% of the bone marrow volume receives more than 11.5Gy, and so falls slightly short of meeting all of our treatment plan criteria.) This suggests that this type of parallelized projected gradient algorithm has strong potential to be used to generate fixed beam solutions that could then be used to develop arc therapy treatment plans in a clinically realistic timeframe.

Of the three sets of parameters that we tested, opt_atlasair_2 gives the best results. Comparing the opt_atlasair_2 set to the opt_ooc_3 set, we see that the opt_ooc_3 set allows for a lower overall dose to the lungs (just under 10% of the volume of the lungs receiving 10Gy or more for opt_ooc_3 versus roughly 15% of the volume of the lungs receiving 10Gy or more for opt_atlasair_2). However, the opt_ooc_3 set allows for an unacceptable level of dose to the spinal cord, which is not the case for opt_atlasair_2 (spinal cord median dose is just over 12Gy for opt_ooc_3 versus roughly 6Gy for opt_atlasair_2). Comparing the opt_atlasair_2 set to the opt_original set, we see that the final opt_atlasair_2 plan still achieves the required target dose, but does so with every organ receiving less dose than in opt_original (the median dose of each organ decreases by an amount between roughly 0.5Gy and 1Gy from opt_original to opt_atlasair_2). More interestingly, the opt_atlasair_2 set performs better at controlling the level of overdose in the bone marrow (10% of the target volume receives roughly 19Gy or more for opt_original, versus almost 0% of the volume for opt_original). This may be desirable because as we have mentioned before, high levels of overdose can lead to fibrosis in the bones and detrimentally affect the success of the subsequent bone marrow transplant.

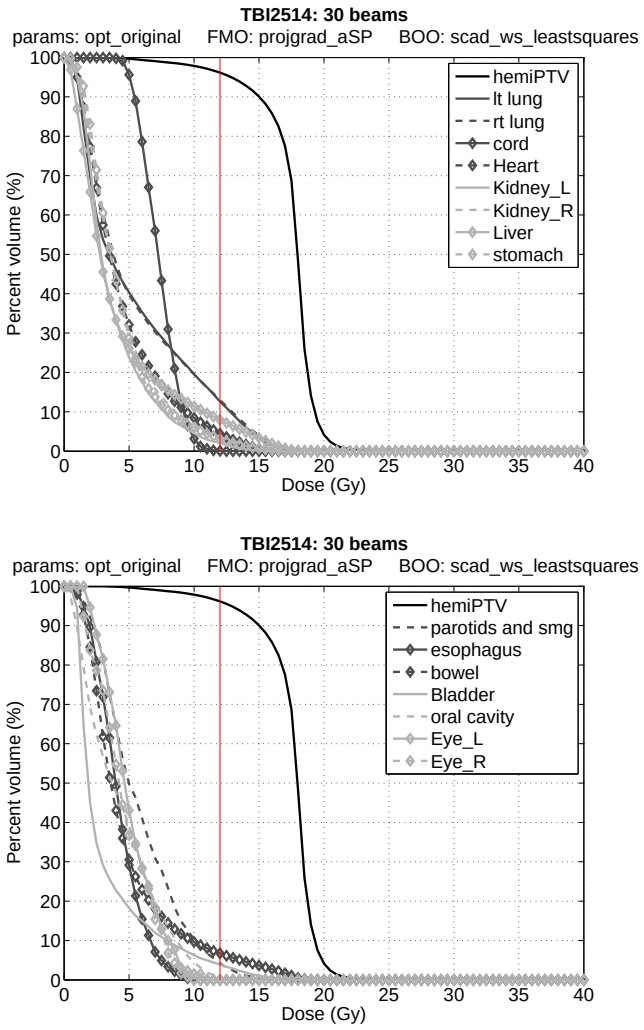


Figure 5.7: DVHs of a 30 beam solution obtained shortly after 12 hours, with `opt_original` parameter set.

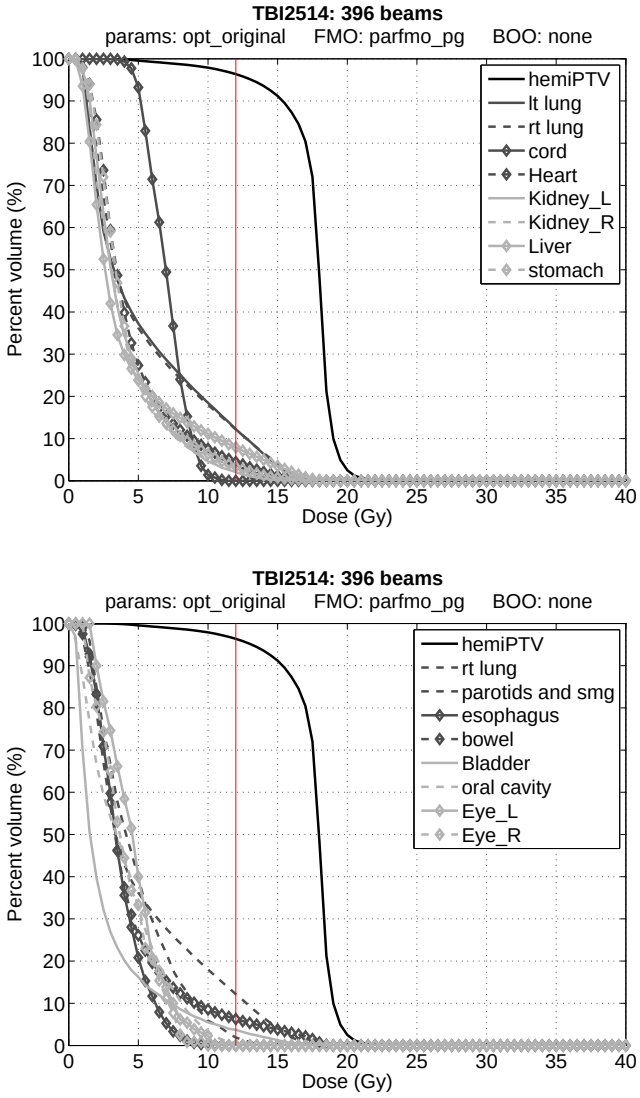


Figure 5.8: DVHs of 396-beam solution at six hours of projected gradient execution time, with `opt_original` parameter set.

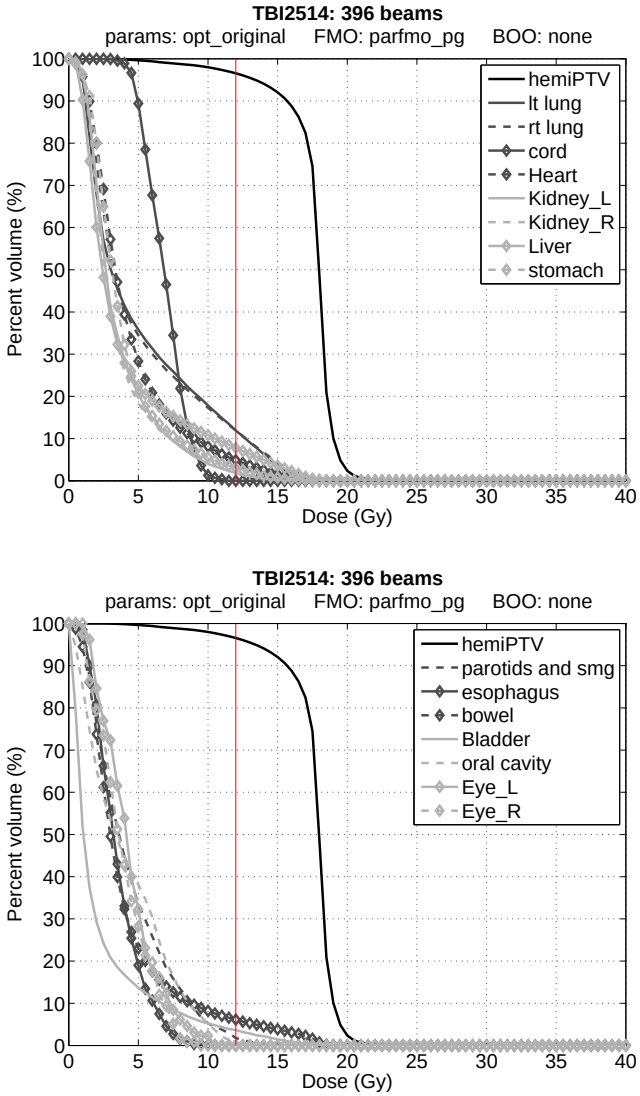


Figure 5.9: DVHs of 396-beam solution at twelve hours of projected gradient execution time, with `opt_original` parameter set.

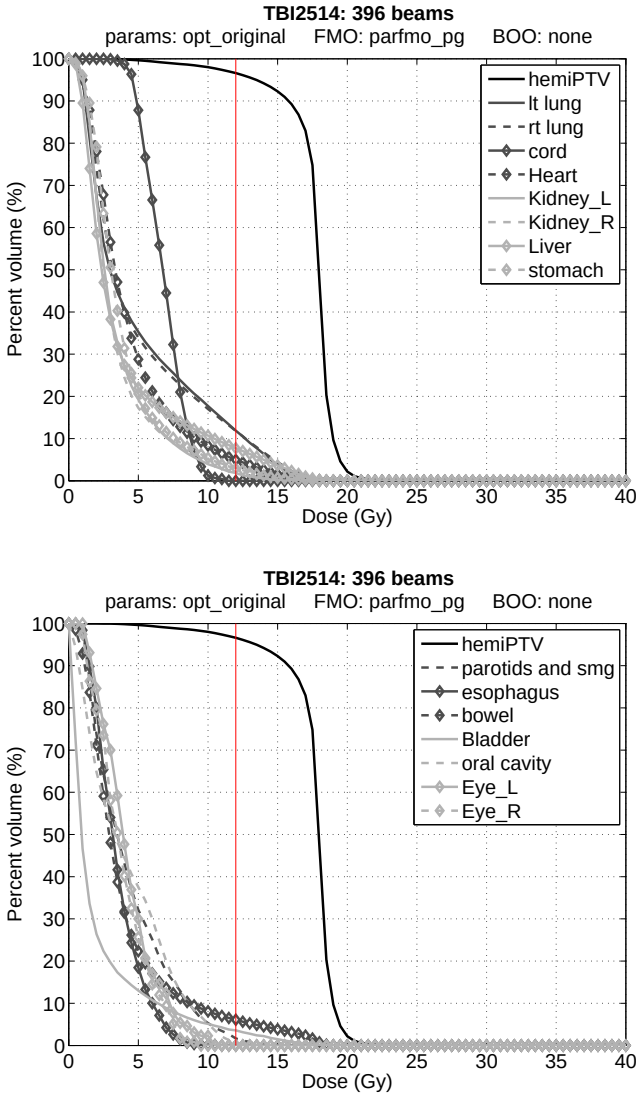


Figure 5.10: DVHs of 396-beam solution shortly after 24 hours of projected gradient execution time, with `opt_original` parameter set.

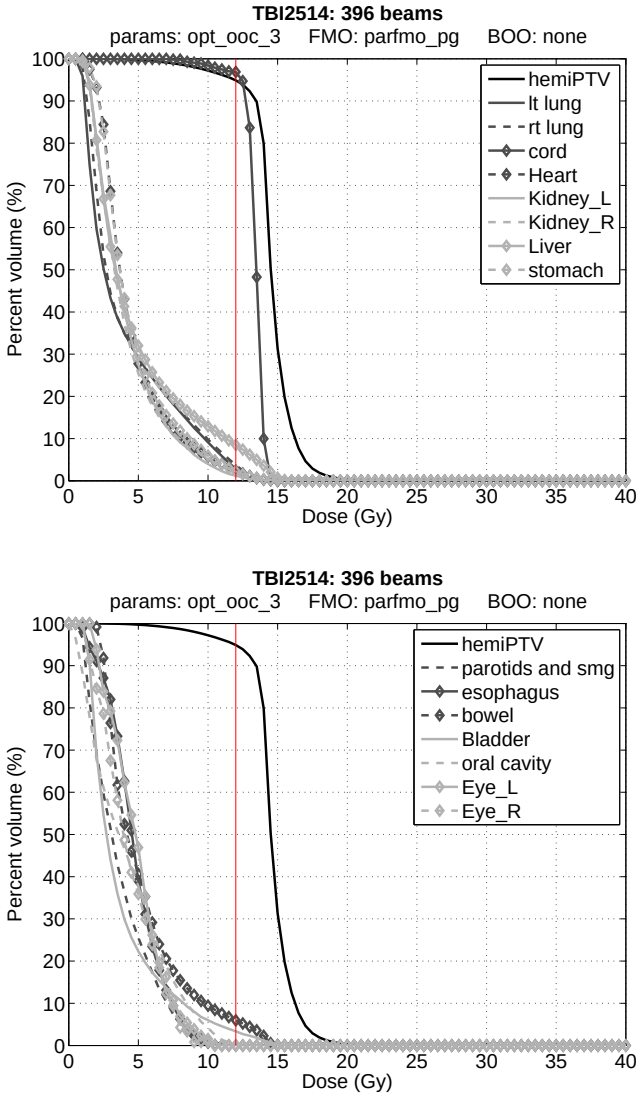


Figure 5.11: DVHs of 396-beam solution at six hours of projected gradient execution time, with `opt_ooc_3` parameter set.

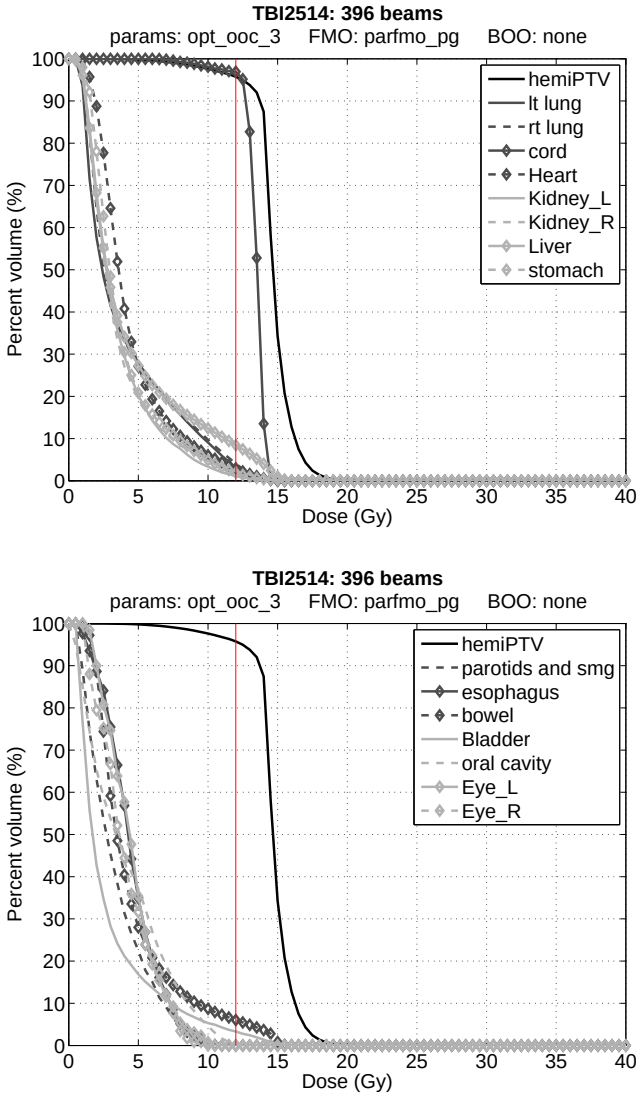


Figure 5.12: DVHs of 396-beam solution at twelve hours of projected gradient execution time, with `opt_ooc_3` parameter set.

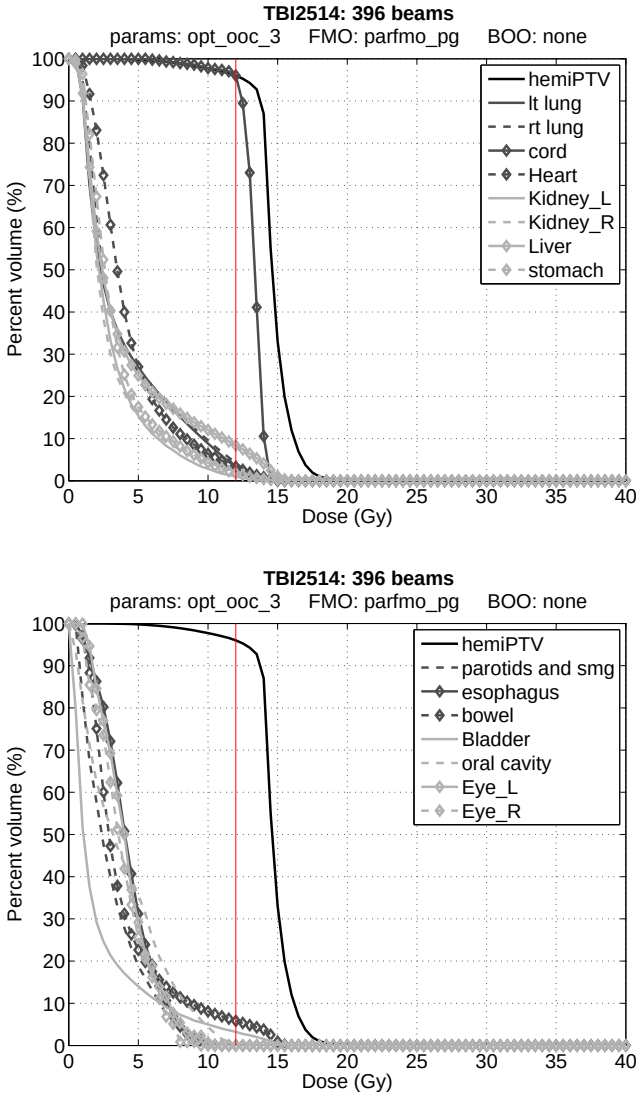


Figure 5.13: DVHs of 396 beam solution shortly after 24 hours of projected gradient execution time, with `opt_ooc_3` parameter set.

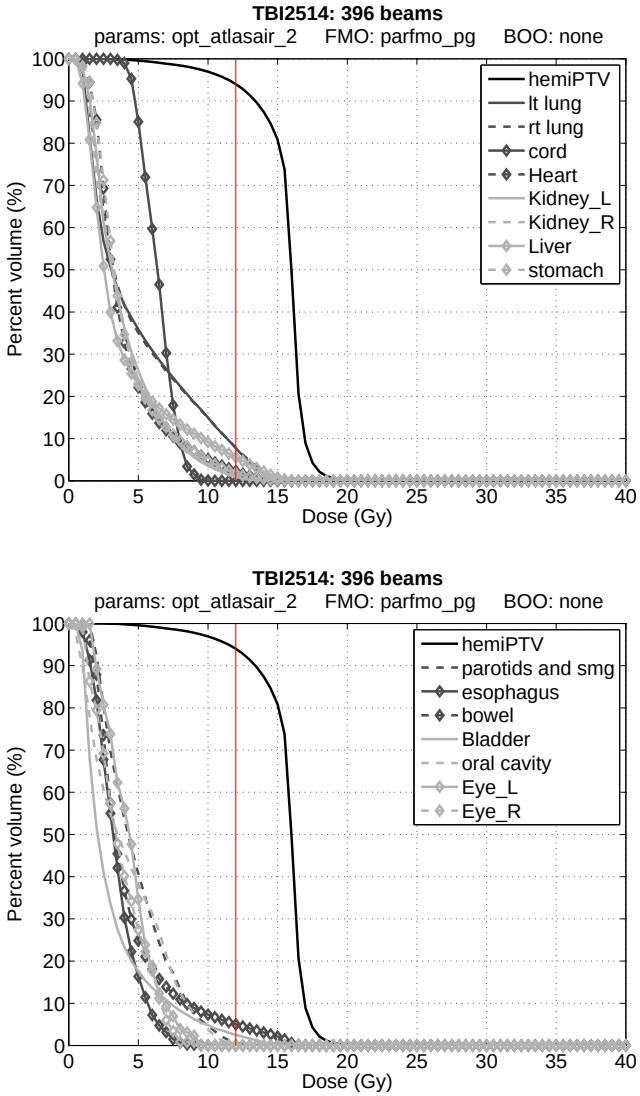


Figure 5.14: DVHs of 396 beam solution at six hours of projected gradient execution time, with `opt_atlasair_2` parameter set.

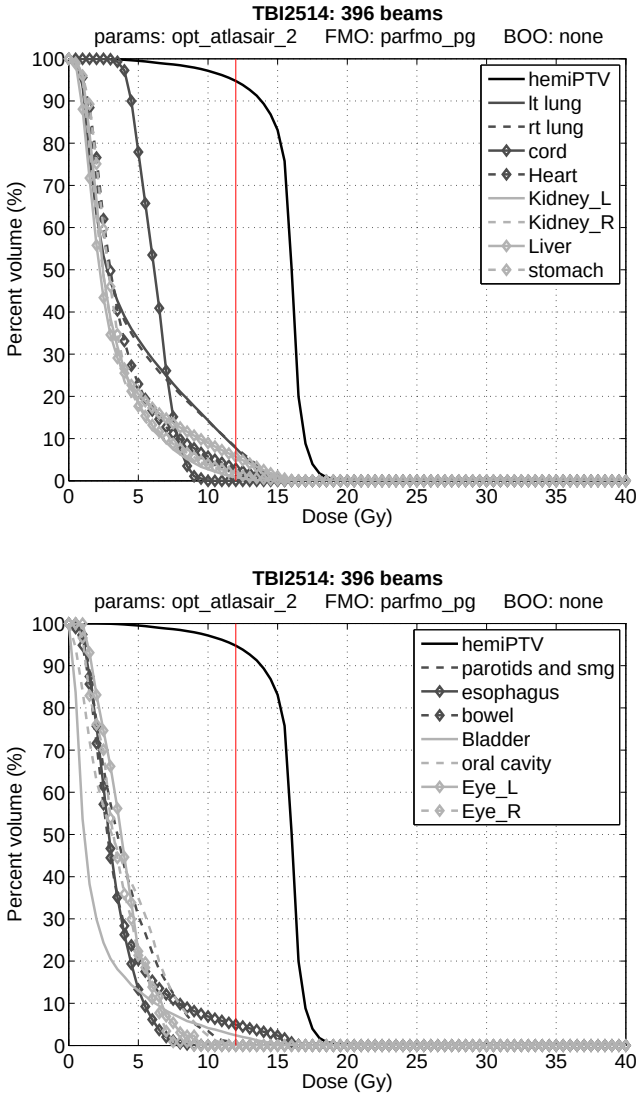


Figure 5.15: DVHs of 396 beam solution at twelve hours of projected gradient execution time, with `opt_atlasair_2` parameter set.

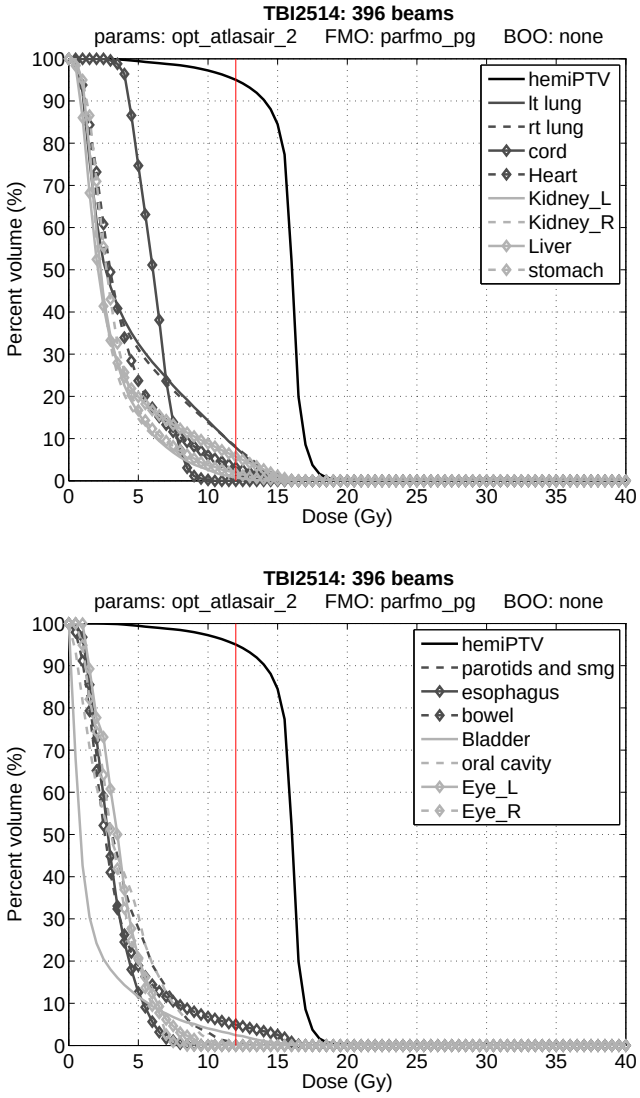


Figure 5.16: DVHs of 396 beam solution shortly after 24 hours of projected gradient execution time, with `opt_atlasair_2` parameter set.

Chapter 6

Conclusions and Future Work

From this study we can reach a number of important conclusions regarding methods to improve fluence map optimization for TMI. From our discussion of line search strategies in Chapter 3, we see that at the moment, the backtracking line search strategy seems to provide the best results of all three line search methods, both in terms of computation time and quality of the end solution. However, more testing is required to see whether the reduced step and forward line search can attain an improvement over the normal backtracking line search implementation of projected gradient.

In Chapter 4, we studied three different techniques for warm-starting FMO evaluation in the framework of the Add/Drop algorithm for BOO. We saw that both warm-start techniques (warm-start using averaging and warm-start using least-squares) led to a significant reduction in computation time over the standard cold-start method, and also led to significantly lower final FMO values. Both of the warm-start methods which we studied could be readily implemented in MATLAB and could be applied to FMO for other treatment planning problems, not just TMI. Also, these methods help to highlight the versatility of the FMO formulation that we use and why it may be desirable to use the exact FMO value (as opposed to an approximation or a different figure of merit) as the quality of a set of beams.

Finally, in Chapter 5 we developed parallelized algorithms for objective function and gradient evaluation, and used these algorithms to obtain high-quality treatment plans using all available beams. By solving the FMO problem with all possible beams, we have discovered what appear to be basic limitations of IM-TMI: in particular, that it is not possible to achieve a median dose below 2Gy in the majority of the organs simultaneously, while meeting the target dose requirement; that the lungs and spinal cord are the most difficult critical organs to spare; and that it is not possible to drive the dose in the lungs below a certain level without greatly increasing the dose in the spinal cord. We have also

discovered that the 30-beam solutions which we proposed earlier in Mišić et al. [2010], although not as good as the corresponding `opt_original` solutions with 396 beams, are not significantly worse than the 396 beam solutions.

In future work, we could consider other kinds of warm-start procedures to speed up FMO evaluation in Add/Drop – in particular, we could consider warm-start schemes where instead of directly reusing the bixels of $\Theta \setminus \{\boldsymbol{\theta}_k\}$, we instead modify those bixels in some way in response to the dose deposition coefficients of the new beam, $\tilde{\boldsymbol{\theta}}_k$. It would also be interesting to more closely examine the structure of our FMO problem and the structure of the D_{ijs} values to determine if it is possible to place some kind of guarantee on the quality of the initial set of bixels obtained by the warm-start using averaging and warm-start using least-squares techniques (e.g. if the D_{ijs} of $\boldsymbol{\theta}_k$ and $\tilde{\boldsymbol{\theta}}_k$ "differ" by some amount, then what kind of objective function value can we expect for the initial bixel values $\tilde{x}$?).

The implementation of projected gradient using the parallelized objective function and gradient algorithms has yielded some interesting results about what the best physically possible IMRT treatment looks like and where our previous 30-beam solutions stand. More parameter sets should be tested to confirm that the parameter sets shown here do represent the best possible solutions. An intriguing observation that we made about this algorithm was that for both `opt_atlasair_2` and `opt_original` parameters sets, the solution obtained after 6 hours generally compared quite favorably to the 30-beam solution obtained after 12 hours. It would be interesting to explore whether these solutions could in some way be used to identify which beams in the set of all beams are better than others and guide the design of plans using smaller numbers of beams.

Finally, as stated earlier, a promising alternative to conventional IMRT for TMI may be arc therapy. The parallelism-enhanced projected gradient algorithm that we studied in Chapter 5 constitutes the first step in designing an arc therapy plan; the next step would involve designing an algorithm to determine an appropriate "route" to take through the beams, and to fill in the bixels along the path in some optimal way.

Bibliography

R. Acosta, M. Ehrgott, A. Holder, D. Nevin, J. Reese, and B. Salter. *The influence of dose grid resolution on beam selection strategies in radiotherapy treatment design*, volume 12 of *Springer Optimization and Its Applications*, pages 1–23. Springer New York, 2008.

D. M. Aleman, A. Kumar, R. K. Ahuja, H. E. Romeijn, and J. F. Dempsey. Neighborhood search approaches to beam orientation optimization in intensity modulated radiation therapy treatment planning. *Journal of Global Optimization*, 2008a.

D.M. Aleman, H.E. Romeijn, and J.F. Dempsey. A response surface approach to beam orientation optimization in intensity modulated radiation therapy treatment planning. *INFORMS Journal on Computing: Computational Biology and Medical Applications*, 20(3):1–15, 2008b.

B. Aydogan, A. J. Mundt, and J. C. Roeske. Linac-based total marrow irradiation (IM-TMI). *Technology in Cancer Research and Treatment*, 5(5):513–519, 2007.

B. Aydogan and J. C. Roeske. Feasibility study of linac-modulated total marrow irradiation (IM-TMI). *Medical Physics*, 34(6):2455–2455, 2007.

D. P. Bertsekas. On the Goldstein-Levitin-Polyak Gradient Projection Method. *IEEE Transactions on Automatic Control*, 21(2):174–184, 1976.

B. C. J. Cho, W. H. Roa, D. Robinson, and B. Murray. The development of target-eye-view maps for selection of coplanar or noncoplanar beams in conformal radiotherapy treatment planning. *Medical Physics*, 26(11):2367–2372, 1999.

B. Choi and J. O. Deasy. The generalized equivalent uniform dose function asa basis for intensity-modulated treatment planning. *Physics in Medicine and Biology*, 47:3579–3589, 2002.

R. A. Clift, C. D. Buckner, F. R. Appelbaum, E. Bryant, S. I. Bearman, F. B. Petersen, L. D. Fisher, C. Anasetti, P. Beatty, W. I. Bensinger, K. Doney, R. S. Hill, G. B. McDonald, P. Martin, J. Meyers, J. Sanders, J. Singer,

P. Stewart, K. M. Sullivan, R. Witherspoon, R. Storb, J. A. Hansen, and E. D. Thomas. Allogeneic marrow transplantation in patients with chronic myeloid leukemia in the chronic phase: a randomized trial of two irradiation regimens. *Blood*, 77:1660–1665, 1991.

D. Craft. Local beam angle optimization with linear programming and gradient search. *Physics in Medicine and Biology*, 52:N127–N135, 2007.

D. Djajaputra, Q. Wu, Yan Wu, and R. Mohan. Algoirthm and performance of a clinical IMRT beam-angle optimization system. *Physics in Medicine and Biology*, 48:3191–3212, 2003.

W. D. D'Souza, R. R. Meyer, and L. Shi. Selection of beam orientations in intensity-modulated radiation therapy using single-beam indices and integer programming. *Physics in Medicine and Biology*, 49:3465–3481, 2004.

W. D. D'Souza, H. H. Zhang, D. P. Nazareth, L. Shi, and R. R. Meyer. A nested partitions framework for beam angle optimization in intensity-modulated radiation therapy. *Physics in Medicine and Biology*, 53:3293–3307, 2008.

M. Ehrgott, A. Holder, and J. Reese. Beam selection in radiotherapy design. *Linear Algebra and its Applications*, 428:1272–1312, 2008.

M. Ehrgott and R. Johnston. Optimisation of beam directions in intensity modulated radiation therapy planning. *OR Spectrum*, 25:251–264, 2003.

G. A. Ezzell. Genetic and geometric optimization of three-dimensional radiation therapy treatment planning. *Medical Physics*, 23(3):293–305, 1996.

S. Gaede, E. Wong, and H. Rasmussen. An algorithm for systematic selection of beam directions for IMRT. *Medical Physics*, 31:376–388, 2004.

P. Gokhale, E. M. A. Hussein, and N. Kulkarni. Determination of beam orientation in radiotherapy planning. *Medical Physics*, 21(3):393–400, 1994.

O. C. L. Haas, K. J. Burnham, and J. A. Mills. Optimization of beam orientation in radiotherapy using planar geometry. *Physics in Medicine and Biology*, 43: 2179–2193, 1999.

H. W. Hamacher and K.-H. Küfer. Inverse radiation therapy planning – a multiple objective optimization approach. *Discrete Applied Mathematics*, 118: 145–161, 2002.

Q. Hou, J. Wang, Y. Chen, and J. M. Galvin. Beam orientation optimization for IMRT by a hybrid method of the genetic algorithm and the simulated dynamics. *Medical Physics*, 30:2360–2367, 2003.

K.-H. Küfer, A. Scherrer, M. Monz, F. Alonso, H. Trinkaus, and T. Bortfeld. Intensity-modulated radiotherapy – a large scale multi-criteria programming problem. *OR Spectrum*, 25:223–249, 2003.

A. Kumar. *Novel methods for intensity-modulated radiation therapy treatment planning*. PhD thesis, University of Florida, 2005.

M. Lahanas, E. Schreibmann, and D. Baltas. Multiobjective inverse planning for intensity modulated radiotherapy with constraint-free gradient-based optimization algorithms. *Physics in Medicine and Biology*, 48:2843–2871, 2003.

E. K. Lee, T. Fox, and I. Crocker. Integer programming applied to intensity-modulated radiation therapy treatment planning. *Annals of Operations Research*, 119:165–181, 2003.

E. K. Lee, T. Fox, and I. Crocker. Simultaneous beam geometry and intensity map optimization in intensity-modulated radiation therapy. *International Journal of Radiation Oncology Biology Physics*, 64(1):301–320, 2006.

Y. Li, D. Yao, J. Yao, and W. Chen. A particle swarm optimization algorithm for beam angle selection in intensity-modulated radiotherapy planning. *Physics in Medicine and Biology*, 50:3491–3514, 2005.

Y. Li, J. Yao, and D. Yao. Automatic beam angle selection in IMRT planning using genetic algorithm. *Physics in Medicine and Biology*, 49:1915–1932, 2004.

G. J. Lim, J. Choi, and R. Mohan. Iterative solution methods for beam angle and fluence map optimization in intensity modulated radiation therapy planning. *OR Spectrum*, 30:289–309, 2008.

G. J. Lim, A. Holder, and J. Reese. A clustering approach for optimizing beam angles in imrt planning. In *Proceedings of the IIE Annual Conference*. IERC, 2009.

H. H. Liu, M. Jauregui, X. Zhang, X. Wang, L. Dong, and R. Mohan. Beam angle optimization and reduction for intensity-modulated radiation therapy of non-small-cell lung cancer. *International Journal of Radiation Oncology Biology Physics*, 65(2):561–572, 2006.

G. Meedt, M. Alber, and F. Nüsslin. Non-coplanar beam direction optimization for intensity-modulated radiotherapy. *Physics in Medicine and Biology*, 48: 2999–3019, 2003.

C. Men, X. Gu, D. Choi, A. Majumdar, Z. Zhen, K. Mueller, and S. B. Jian. Gpu-based ultrafast IMRT plan optimization. *Physics in Medicine and Biology*, 54:6565–6573, 2009.

V. V. Mišić, D. M. Aleman, and M. B. Sharpe. Total marrow irradiation using intensity modulated radiation therapy optimization. In *Proceedings of the IIE Annual Conference*. IERC, 2009.

V. V. Mišić, D. M. Aleman, and M. B. Sharpe. Neighborhood search approaches to non-coplanar beam orientation optimization for total marrow irradiation using IMRT. *European Journal of Operational Research*, 205(3), 2010.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Science, 2006.

A. Ólafsson, R. Jeraj, and S. J. Wright. Optimization of intensity-modulated radiation therapy with biological objectives. *Physics in Medicine and Biology*, 50:5357–5379, 2005.

J. Osterman, W. Weaver, J. Slater, and P. Glass. Failure-proof method for instrinsic field subtraction. *Manhattan Journal of Physics*, 1(4):7–10, 1959.

P. S. Potrebko, B. M. C. McCurdy, J. B. Butler, and A. S. El-Gubtan. Improving intensity-modulated radiation therapy using the anatomic beam orientation optimization algorithm. *Medical Physics*, 35:2170–2179, 2008.

F. Preciado-Walters, R. Rardin, M. Langer, and V. Thai. A coupled column generation, mixed integer approach to optimal planning of intensity modulated radiation therapy for cancer. *Mathematical Programming*, 101(2):319–338, 2004.

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(5):1361–1370, 2001.

A. Pugachev and L. Xing. Incorporating prior knowledge into beam orientation optimization in IMRT. *International Journal of Radiation Oncology Biology Physics*, 54(5):1565–1574, 2002.

J. G. Reese. Vector quantization, graph theory and IMRT design, 2005. Senior Project, Trinity University.

H. E. Romeijn, R. K. Ahuja, J. F. Dempsey, and A. Kumar. A new linear programming approach to radiation therapy treatment planning problems. *Operations Research*, 54(2):201–216, 2006.

C. G. Rowbottom, C. M. Nutting, and S. Webb. Beam-orientation optimization of intensity-modulated radiotherapy: clinical application to parotid gland tumours. *Radiotherapy and Oncology*, 59:169–177, 2001.

C. G. Rowbottom, M. Oldham, and S. Webb. Constrained customization of non-coplanar beam orientations in radiotherapy of brain tumours. *Physics in Medicine and Biology*, 44:383–399, 1999a.

C. G. Rowbottom, S. Webb, and M. Oldham. Beam-orientation customization using an artificial neural network. *Physics in Medicine and Biology*, 44:2251–2262, 1999b.

E. Schreibmann, M. Lahanas, R. Uricchio, K. Theodorou, C. Kappas, and D. Baltas. A geometry based optimization algorithm for conformal external beam radiotherapy. *Physics in Medicine and Biology*, 48:1825–1841, 2003.

E. Schreibmann, M. Lahanas, L. Xing, and D. Baltas. Multiobjective evolutionary optimization of the number of beams, their orientations and weights for intensity-modulated radiation therapy. *Physics in Medicine and Biology*, 49: 747–770, 2004.

E. Schreibmann and L. Xing. Dose-volume based ranking of incident beam direction and its utility in facilitating IMRT beam placement. *International Journal of Radiation Oncology Biology Physics*, 63(2):584–593, 2005.

T. E. Schultheiss, J. Wong, A. Liu, G. Olivera, and G. Somlo. Image-guided total marrow and total lymphatic irradiation using helical tomotherapy. *International Journal of Radiation Oncology, Biology, Physics*, 67:1259–1267, 2007.

S. Söderström and A. Brahme. Selection of suitable beam orientations in radiation therapy using entropy and fourier transform measures. *Physics in Medicine and Biology*, 37(4):911–924, 1992.

J. Stein, R. Mohan, X.-H. Wang, T. Bortfeld, Q. Wu, K. Preiser, C. C. Ling, and W. Schlegel. Number and orientations of beams in intensity-modulated radiation treatments. *Medical Physics*, 24(2):149–160, 1997.

C. Thieke, T. Bortfeld, A. Niemierko, and S. Nill. From physical dose constraints to equivalent uniform dose constraints in inverse radiotherapy planning. *Medical Physics*, 30(9), 2003.

A. Trofimov, E. Rietzel, H.-M. Lu, B. Martin, S. Jiang, G. T. Y. Chen, and T. Bortfeld. Temporo-spatial IMRT optimization: concepts, implementation and initial results. *Physics in Medicine and Biology*, 50:2779–2798, 2005.

X. Wang, X. Zhang, L. Dong, H. Liu, Q. Wu, and R. Mohan. Development of methods for beam angle optimization for IMRT using an accelerated exhausitve search strategy. *International Journal of Radiation Oncology Biology Physics*, 60(4):1325–1337, 2004.

J. Y. C. Wong, A. Liu, T. Schultheiss, L. Popplewell, A. Stein, J. Rosenthal, M. Essensten, S. Forman, and G. Somlo. Targeted total marrow irradiation using three-dimensional image-guided tomographic intensity-modulated radiation therapy: An alternative to standard total body irradiation. *Biology of Blood and Marrow Transplantation*, 12(3):306–315, 2006.

Y. Zinchenko, T. Craig, H. Keller, T. Terlaky, and M. Sharpe. Controlling the dose distribution with gEUD-type constraints within the convex radiotherapy optimization. *Physics in Medicine and Biology*, 53:3231–3250, 2008.