Operations Research Methods for Optimization in Radiation Oncology

Review Article

Matthias Ehrgott[1]*, Allen Holder[2]†

1 Department of Management Science, Lancaster University, Lancaster, United Kingdom

2 Department of Mathematics, Rose-Hulman Institute of Technology, Terre Haute, IN, USA

Abstract: Operations Research has a successful tradition of applying mathematical analysis to a wide range of applications, and problems in Medical Physics have been popular over the last couple of decades. The original application was in the optimal design of the fluence map for a radiotherapy treatment, a problem that has continued to receive attention. However, Operations Research has been applied to other clinical problems like patient scheduling, vault design, and image alignment. The overriding theme of this article is to present how techniques in Operations Research apply to clinical problems, which we accomplish in three parts. First, we present the perspective from which an operations researcher addresses a clinical problem. Second, we succinctly introduce the underlying methods that are used to optimize a system, and third, we demonstrate how modern software facilitates problem design. Our discussion is supported by several publications to foster continued study. With numerous clinical, medical, and managerial decisions associated with a clinic, operations research has a promising future at improving how radiotherapy treatments are designed and delivered.

1. Introduction

Operations Research (OR) is an area of study that focuses on the analysis of decision making. The field was born out of the military need to improve the efficiency of "operations" during the second world war, but the associated methods of modeling, optimizing, and analyzing real world phenomenon have found a spectrum of applications across a range of disciplines. A recent area of interest is the application of OR to medicine, and most notably,

---

* *E-mail: m.ehrgott@lancaster.ac.uk*

† *E-mail: holder@rose-hulman.edu, p: +1 812 877 8682*

to the optimal design of a radiotherapy treatment. The original suggestion to optimize the design process was made in 1968 by Bahr et al [4]. This early publication shows that the medical physics community recognized the use of OR long before the OR community recognized the applications in Medical Physics. However, in the 1990s the operations research community became aware of the array of important applications in Medical Physics, and today there is a devoted collection of OR experts whose primary interests lie in the study of applying OR to problems in Medical Physics, e.g. see the recent works of [2, 6, 9, 35].

Through the authors' professional interactions, we have observed that the OR and Medical Physics communities approach problems from different perspectives, and one of the goals of this article is to address the questions of how and why OR is beneficial to problems in Medical Physics? One of the main differences in research methodologies is that the OR community uses a rich taxonomy of problem classes that are individually well studied in their mathematical abstraction, and when a problem in medical physics is considered by an OR expert, one of the initial objectives is to model the problem so that it becomes a member of one of these classes. As such, we are not immediately concerned with solving the problem but are instead interested in modeling and classifying it as a member of a known class. Once a problem is classified, we address how to use (or solve) an appropriate model, but since it is a member of a specified class, we can likely use or adapt the solution methods developed for the class. We often continue by analyzing the results to better understand their value to the application. So from an OR perspective, the general research process is to model, solve, and analyze against an underlying taxonomy of well known problems. Of course, not every problem fits nicely within the taxonomy, and in this case, the OR expert has to either combine known methods or invent new ones, and in either case, the field of OR benefits from the advancement. This type of work similarly follows the central process of modeling, solving and analyzing, and it is this methodology that we convey.

One of the advantages of addressing a problem through the lens of OR is that we can harness the immense professional expertise of prior research, which is often embodied in state-of-the-art software. This routinely alleviates the OR expert from the tedious development of software to address a particular problem. Instead, a problem can be modeled with professional software that is linked to advanced algorithms. This streamlines problem solving and facilitates investigation since altering and solving a model are seamless. Examples of this solution approach are presented throughout.

Consistent with the OR perspective, we have organized the remainder of the paper as follows. Section 2 considers convex problems in Medical Physics. We continue with a discussion of discrete problems in Section 3, which is followed by a collection of other problem formats in Section 4. Section 5 discusses the important idea of considering multiple, and often competing, objectives. Each section points to problems in Medical Physics that have been modeled as a member of the associated problem class. Moreover, each section explains how to

recognize whether or not the abstraction of a real-world phenomena can be appropriately modeled as a member of a class. Information about software and solution analysis is also included.

We mention that several reviews of Medical Physics applications already exist in the OR literature, and we point readers to [20] and [52] as modern and sweeping reviews. Our exposition is different because it focuses more on the OR methods for medical physicists instead of focusing on the applications for the OR community. Any reader who would like to learn more about OR and optimization is pointed to the classic text of [27].

2. Convex Problems

The study of convex optimization is arguably at the foundation of the field of optimization. There are two reasons, 1) many real-world problems naturally fall into this category, and 2) convexity is a mathematical property that allows us to prove the optimality of a solution. The general form is

$$\min\{f(x) : x \in X\}, \tag{1}$$

where f is a convex function and X is a convex set of n-vectors. The problem asks us to find the smallest value of f, called the *objective function*, over the collection of vectors in X. The set X is convex if for all x and y in X, we have that the line segment between them is also in X, i.e.

$$(1 - \alpha)x + \alpha y \in X,$$

provided that $0 \leq \alpha \leq 1$. The set X is called the *feasible region* and is commonly defined functionally as $X = \{x : g(x) \leq 0\}$. In this case the set X is convex if g is a convex function, and (1) becomes

$$\min\{f(x) : g(x) \leq 0\}. \tag{2}$$

Beginning calculus students learn an intuitive concept of convexity, which promotes that a function is convex if it's second derivative is positive. Such intuition is overly restrictive since it only considers real-valued, twice differentiable functions of a single variable. Instead, we let x be a vector of length n and use a definition that weakens the differentiability condition. A real-valued function is *convex* if

$$f((1 - \alpha)x + \alpha y) \leq (1 - \alpha)f(x) + \alpha f(y), \tag{3}$$

for any x and y in the function's domain and any α satisfying $0 \leq \alpha \leq 1$. This condition essentially states that the line segment between $(x, f(x))$ and $(y, f(y))$ is above or identical to the function. The definition requires no differentiability, although it is worth mentioning that it does imply continuity. If f is twice differentiable,

this condition is the same as the eigenvalues of the Hessian being non-negative, which reduces to the intuitive concept from calculus. The function is *strictly convex* if the inequality in (3) holds strictly for $0 < \alpha < 1$ and $x \neq y$. The function g need not be single valued, and in general, we assume that g maps the n-vector x to the t-vector $(g_1(x), g_2(x), \ldots, g_t(x))^T$. In this case, g is convex if each component function $g_i(x)$ is convex.

Joseph-Louis Lagrange considered optimization problems like (2) in the middle of the 18th century, and his investigation provided mathematical constructs to solve such problems. The function underlying the theoretical development bears his name, and the Lagrangian is

$$(x, \lambda) = f(x) - \lambda^T g(x).$$

After years of study by great minds like von Neumann, the core theory to solve optimization problems with digital computation was complete. Without a foray into the theoretical development, the theme is that under the assumption that f and g have suitably nice analytic conditions, we have that if x^* is an optimal solution to (2), then there is a $\lambda^* \leq 0$ so that

$$g(x^*) \leq 0, \ \nabla f(x^*) - (\lambda^*)^T \nabla g(x^*) = 0, \text{ and } (\lambda^*)^T g(x^*) = 0. \tag{4}$$

These are often called the first order (necessary) Lagrange conditions for optimality. Alone, these equations do not classify optimality since they may have solutions that are not optimal. However, if f and g are convex, then these equations are satisfied if and only if x^* is optimal, which is the quintessential reason why convex problems are important. This means solving a convex optimization problem is the same as solving (4).

Convex problems often arise in Medical Physics as deviation problems. A prominent example is the optimal design of a fluence map in intensity modulated radiotherapy, and we first consider such problems. Let x represent the fluence over a collection of angles and $D(x)$ be the linear operator that maps fluence into dose. A naïve but illustrative treatment design problem is

$$\min \left\{ \|D(x) - T\|_p : x \geq 0 \right\}. \tag{5}$$

A bit of book keeping is needed to make sense of this model. The vector $D(x)$ is usually segmented into sub-vectors that give the dose to the target, organs-at-risk, and normal tissues, and the corresponding sub-vectors of T represent the target amounts for each of the tissues, which are normally zero except for the target. A solution to the model is a fluence pattern x that minimizes the deviation from the desired dose T. We mention that deviations for different tissues are often considered individually and weighted to form an objective. The problem is then an instance of a multiple objective problem, a topic that is developed in Section 5.

The parameter p defines the norm used to measure the deviation, and the 3 cases of p being 1, 2 and ∞ are common. For $p = 1$ and $p = 2$, we have that $\|D(x) - T\|_p = (\sum_{i=1}^{m} |D_i(x) - T_i|^p)^{1/p}$.

In particular, if $p = 1$, the problem is linear, and if $p = 2$, the problem is quadratic and asks us to find a least squares solution. For $p = \infty$, we have $\|D(x) - T\|_p = \max\{|D_i(x) - T_i| : i = 1, 2, \ldots, m\}$, which means the problem minimizes the maximum deviation. This is again a linear problem.

All three models are convex, and all three can be solved by satisfying (4). We detail the case for $p = \infty$. If we let A be the matrix so that $D(x) = Ax$, then (5) can be re-written as,

$$\min\{\|Ax - T\|_\infty : x \geq 0\} \Leftrightarrow \min\{z : -ze \leq Ax - T \leq ze, \ x \geq 0, \ z \geq 0\},$$

where e is a vector of ones of necessary length and z is an added variable that measures the maximum deviation through the constraint $-ze \leq Ax - T \leq ze$. The model on the right is linear, and in this case, the necessary and sufficient conditions in (4) become,

$$Ax + ze \ \geq \ T \tag{6}$$

$$-Ax + ze \ \geq \ -T \tag{7}$$

$$A^T \lambda' - A^T \lambda'' \ \leq \ 0 \tag{8}$$

$$e^T \lambda' + e^T \lambda'' \ \leq \ 1 \tag{9}$$

$$(Ax - ze - T)^T \lambda' + (-Ax + ze + T)^T \lambda'' \ = \ 0 \tag{10}$$

$$x, z, \lambda', \lambda'' \ \geq \ 0. \tag{11}$$

We reiterate that any solution $(x, z, \lambda', \lambda'')$ gives an optimal solution of z because the problem is convex. Similar systems exist for the $p = 1$ and $p = 2$ cases.

If the original problem is strictly convex, then we can prove that systems like (6) through (11) have a unique solution. This is the case for $p = 2$ (least-squares), which is one of the advantages of the 2-norm. However, this is not generally a good reason to use the 2-norm. Indeed, the norm should be selected to best approach the situation. For example, we know the maximum deviation from our prescribed dose if we solve the problem with the infinity-norm. If the deviation is minuscule, then we have likely achieved a favorable fluence pattern. Such a guarantee is not available with other norms. The optimal value of the problem with the 1-norm, divided by the number of voxels, is the minimum average deviation, which may include a few large deviations balanced against several small deviations. The 2-norm has the same behavior, although it places greater emphasis on decreasing large deviations. In general, larger values of p reduce large deviations, with the ultimate value of

$p = \infty$ minimizing the largest deviation. From a modeling perspective, the value of p should be selected to fit the desired emphasis on large deviations. For a fluence problem, it might make sense to minimize the infinity-norm first, and if the value is small, then the treatment could be accepted. If the value is large, then we have gained the knowledge that large deviations from the prescribed dose are necessary, and we might proceed with a subsequent solve with the 1 or 2-norm to find an optimal fluence pattern that counters large deviations with a preponderance of small deviations. Of course, a treatment planner would need to inspect the spatial position of the deviations to ensure a standard of care. Restricting dose volumetrically and spatially can be achieved with additional constraints, some of which are addressed in the next section.

The special case of linear programming is important to many areas of optimization since we often use successive linear approximations or relaxations to solve a problem even if it is not linear. For this reason a few words about linear programming are important. Linear programs are not strictly convex, which means that we can not generally guarantee a unique solution to a system like (6) through (11). In particular, an important but often overlooked fact is that different algorithms often terminate with different solutions. The source of this issue can be succinctly described in reference to the necessary and sufficient Lagrange conditions. Almost all solution procedures divide the system into the three categories of primal feasibility, (6) & (7) with the nonnegativity of x and z; dual feasibility, (8) & (9) with the nonnegativity of λ' and λ''; and complementarity, (10). The general attack is to satisfy two of the three categories and search for the third. A primal method, such as the primal simplex method, satisfies primal feasibility and complementarity and searches for dual feasibility, at which point it stops. A dual method instead satisfies dual feasibility and complementarity and searches for primal feasibility. Interior point methods satisfy primal and dual feasibility and search for complementarity. Since different methods solve the system differently, it is understandable that different algorithms could result in different optimal solutions to the same problem.

Many users of optimization are interested in an algorithm's speed, but the solution's characteristics should also be considered. For example, in [22] it is shown that the dual simplex method, which is commonly the fastest option, tends to group fluence so that a few angles deliver large amounts of unacceptable dose. The interior point methods, which are provably more efficient in the worst-case, tend to distribute fluence over many angles. Although each algorithm is efficient on real problems, the characteristics of the optimal solutions vary significantly. We suggest selecting a solution method and model that fits the desired outcome.

Linear programming has also appeared opaquely in medical physics [28, 51]. As an example, the biological objective of maximizing the probability of tumor control is used in [49], but this objective is the exponential of a linear function. Since the exponential is strictly monotonic, optimizing the biological objective gives the same solution as optimizing the linear function. Similar biological objectives also equate to linear programs.

We turn our discussion to how modeling and solving are separate but linked entities in OR. We demonstrate some of what we have discussed above by adapting the simple fluence model in (5). The data needed for the problem is the dose matrix A and the prescription T. Since the same problem is used for illustrative purposes in the more challenging problems of later sections, we consider a much simplified version from what would be clinically meaningful. However, the point of the example is to show how an OR expert models and solves a problem. Our examples are based on the Modeling software $AMPL^{\copyright}$, which links to a suite of different numerical solvers. If not stated, we used $CPLEX^{\copyright}$ as our solver. All examples may be downloaded from http://holderfamily.dot5hosting.com/aholder/research/papers/Supplements/pmb.tar.gz. We consider the acoustic neuroma depicted in Figure 1. A small pencil beam model was used to calculate dose [38], and the prescription, which comprised the parameter T, was to deliver 60 units of dose to the target and none to the remaining tissues. The image was divided into a grid of 50×50 voxels (3mm thickness). Upon relabeling, the first 9 voxels were the target, the following 14 voxels were the left eye socket, the next 33 voxels were the brain stem, and the last 998 voxels were the remaining normal tissue. We consider 6 equispaced angles, each with 10 pencils, of which only those delivering significant dose to the target were used. This left 51 pencils whose fluence was to be decided.

The deviation model in (5) is too simplistic to interpret clinically since it treats all voxel deviations the same. For example, if $p = 1$, the objective is to minimize the sum of deviations, and hence, large anatomical structures dominate the design process. Since the normal tissue has the preponderance of voxels, the optimal solution to (5) is $x = 0$ if $p = 1$, i.e. the optimal treatment is no treatment. Similar issues arise if $p = 2$ or $p = \infty$. To impart a minor clinical interpretation, we alter (5) to

$$\min \{ \omega \|D(x) - T\|_p + \omega \|D(x) - T\|_p$$
$$+ \omega \|D(x) - T\|_p + \omega \|D(x) - T\|_p : x \geq 0 \} . \tag{12}$$

The subscripts PTV, STM, EYE, and $NRML$ indicate the dose and prescription levels for the target (PTV), the brain stem (STM), the left eye socket (EYE), and the remaining normal tissue ($NRML$). This model remains convex but distinguishes between deviations in different tissues. The ω scalars allow us to weight the importance of the different tissues. This model is a scalarization of a multiple objective problem, a problem class covered in Section 5.

The code in Figure 1 illustrates the simplicity of creating a model with modeling software. These 22 lines are all that is needed to generate a 2-norm version of (12). Importantly, this model statement is independent of problem size, which is dictated by the size of the data and not the mathematical relationships of the model. The data is located in another file, and although this would change for different patients, different prescriptions, and different model parameters, this same model statement works as long as the goal is to solve the associated 2-norm

problem. The first 10 lines of code define the index sets on which the problem's data relies. The set ANGLES indexes the angles, BEAMS indexes the sub-beams (sometimes called bixels) in each angle, and PENCILS is a collection of angle, sub-beam pairs. The VOXEL sets are similar. The `param` commands inform AMPL to expect a matrix A, whose rows are indexed by VOXELS and whose columns are indexed by PENCILS. A target dose T is also expected. Each pencil has an associated variable that represents its fluence. The vector of nonnegative variables is labeled x in the model statement. The objective is named "Deviation" and is the square of the 2-norm. The ω scalars multiply the deviations for the target and the brain stem by 10 per voxel - notice that we divide by the number of voxels in each structure, i.e. we divide by the cardinality (`card`) of each voxel set. Similarly, deviations in the eye socket are multiplied by 1 per voxel and by 1/10 per voxel in the remaining tissue. Figures 2 and 3 show similar code for the 1- and infinity-norms, and readers should notice their similarity.

The solution to the infinity-norm problem delivered 497.81 monitor units along angle $60°$, which gave a maximum deviation of 1.06 Gy from the desired 60 Gy to the target. The brain stem received as high as 57.54 Gy and the normal as much as 62.86 Gy. The eye socket received no significant radiation. The 2-norm problem similarly used only the angle at $60°$, but at the lower amount of 336.28 monitor units. This gave an under treatment of 30.60 Gy on the target but a maximum dose of 48 Gy for the brain stem. The 1-norm instead delivered 309.54 monitor units along angle $60°$ and 114.17 monitor units along angle $240°$, which were opposing angles. This gave a maximum deviation in the target of 12.04 Gy and maximum dose to the brain stem of 59.68 Gy. Each of these models could be altered to represent a myriad of clinical desires, such as dose-volume constraints, hard prescription bounds that must be enforced, the restriction to non-opposing angles, etc.... The point we wish to emphasize is that the fundamental models along with their more meaningful extensions are easily created within a modeling environment, and we hope that readers will consider such systems as they continue their research. The benefits are threefold: 1) models are built with a common language that facilitates dissemination, 2) natural research questions are easily posed and answered by varying the model statement, and 3) several different solvers can be used on the same model. This allows a user to experiment with different model and solver combinations to see how they effect the treatment.

2-Norm

```
set TISSUES;
set ANGLES;
set BEAMS;
set PENCILS within ANGLES, BEAMS;
set VOXELS;
set EYEVOXELS within VOXELS;
set BRNSTMVOXELS within VOXELS;
set TARGETVOXELS within VOXELS;
set OARVOXELS within VOXELS;
set NORMALVOXELS within VOXELS;

param A VOXELS, PENCILS;
param T VOXELS;

var x PENCILS >= 0;

minimize Deviation:
    (10 / card(TARGETVOXELS)) * sum {v in TARGETVOXELS}
        ((sum{(a,i) in PENCILS} A[v,a,i]*x[a,i]) - T[v])^2 +
    (10 / card(BRNSTMVOXELS)) * sum {v in BRNSTMVOXELS}
        ((sum{(a,i) in PENCILS} A[v,a,i]*x[a,i]) - T[v])^2 +
    (1 / card(EYEVOXELS)) * sum {v in EYEVOXELS}
        ((sum{(a,i) in PENCILS} A[v,a,i]*x[a,i]) - T[v])^2 +
    (0.1 / card(NORMALVOXELS)) * sum {v in NORMALVOXELS}
        ((sum{(a,i) in PENCILS} A[v,a,i]*x[a,i]) - T[v])^2;
```

Figure 1. The figure on the left depicts the acoustic neuroma used for our examples. On the right, AMPL code for the 2-norm deviation model in (12).

1-Norm

```
set TISSUES;
set ANGLES;
set BEAMS;
set PENCILS within {ANGLES, BEAMS};
set VOXELS;
set EYEVOXELS within VOXELS;
set BRNSTMVOXELS within VOXELS;
set TARGETVOXELS within VOXELS;
set OARVOXELS within VOXELS;
set NORMALVOXELS within VOXELS;

param A {VOXELS, PENCILS};
param T {VOXELS};

var z {VOXELS} >= 0;
var x {PENCILS} >= 0;

minimize Deviation:
    (10 / card(TARGETVOXELS)) * sum {v in TARGETVOXELS} z[v] +
    (10 / card(BRNSTMVOXELS))* sum {v in BRNSTMVOXELS} z[v] +
    (1 / card(EYEVOXELS)) * sum {v in EYEVOXELS} z[v] +
    (0.1 / card(NORMALVOXELS)) * sum {v in NORMALVOXELS} z[v];

subject to TrgtDeviationUpBound {v in TARGETVOXELS}:
    sum {(a,i) in PENCILS} A[v,a,i] * x[a,i] - T[v] <= z[v];

subject to EyeDeviationUpBound {v in EYEVOXELS}:
    sum {(a,i) in PENCILS} A[v,a,i] * x[a,i] - T[v] <= z[v];

subject to BrnStmDeviationUpBound {v in BRNSTMVOXELS}:
    sum {(a,i) in PENCILS} A[v,a,i] * x[a,i] - T[v] <= z[v];

subject to NrmlDeviationUpBound {v in NORMALVOXELS}:
    sum {(a,i) in PENCILS} A[v,a,i] * x[a,i] - T[v] <= z[v];

subject to TrgtDeviationLowBound {v in TARGETVOXELS}:
    sum {(a,i) in PENCILS} A[v,a,i] * x[a,i] - T[v] >= -z[v];

subject to EyeDeviationLowBound {v in EYEVOXELS}:
    sum {(a,i) in PENCILS} A[v,a,i] * x[a,i] - T[v] >= -z[v];

subject to BrnStmDeviationLowBound {v in BRNSTMVOXELS}:
    sum {(a,i) in PENCILS} A[v,a,i] * x[a,i] - T[v] >= -z[v];

subject to NrmlDeviationLowBound {v in NORMALVOXELS}:
    sum {(a,i) in PENCILS} A[v,a,i] * x[a,i] - T[v] >= -z[v];
```

Figure 2. AMPL code for the 1-norm deviation problem in (12).

Infinity-Norm

```
set TISSUES;
set ANGLES;
set BEAMS;
set PENCILS within {ANGLES, BEAMS};
set VOXELS;  set EYEVOXELS within VOXELS;
set BRNSTMVOXELS within VOXELS;
set TARGETVOXELS within VOXELS;
set OARVOXELS within VOXELS;
set NORMALVOXELS within VOXELS;

param A {VOXELS, PENCILS};
param T {VOXELS};

var z {TISSUES} >= 0;
var x {PENCILS} >= 0;

minimize Deviation:
      10*z["TARGET"] + 10*z["BRNSTM"] + 1*z["EYE"] + 0.1*z["NORMAL"];

subject to TrgtDeviationUpBound {v in TARGETVOXELS}:
      sum {(a,i) in PENCILS} A[v,a,i] * x[a,i] - T[v] <= z["TARGET"];

subject to EyeDeviationUpBound {v in EYEVOXELS}:
      sum {(a,i) in PENCILS} A[v,a,i] * x[a,i] - T[v] <= z["EYE"];

subject to BrnStmDeviationUpBound {v in BRNSTMVOXELS}:
      sum {(a,i) in PENCILS} A[v,a,i] * x[a,i] - T[v] <= z["BRNSTM"];

subject to NrmlDeviationUpBound {v in NORMALVOXELS}:
      sum {(a,i) in PENCILS} A[v,a,i] * x[a,i] - T[v] <= z["NORMAL"];

subject to TrgtDeviationLowBound {v in TARGETVOXELS}:
      sum {(a,i) in PENCILS} A[v,a,i] * x[a,i] - T[v] >= -z["TARGET"];

subject to EyeDeviationLowBound {v in EYEVOXELS}:
      sum {(a,i) in PENCILS} A[v,a,i] * x[a,i] - T[v] >= -z["EYE"];

subject to BrnStmDeviationLowBound {v in BRNSTMVOXELS}:
      sum {(a,i) in PENCILS} A[v,a,i] * x[a,i] - T[v] >= -z["BRNSTM"];

subject to NrmlDeviationLowBound {v in NORMALVOXELS}:
      sum {(a,i) in PENCILS} A[v,a,i] * x[a,i] - T[v] >= -z["NORMAL"];
```

Figure 3.   **AMPL code for the infinity-norm deviation problem in** (12)**.**

As a final example of the use of convex optimization (in this case linear programming) we consider the evaluation of the acceptability of treatment plans. We have illustrated the use of convex optimization problems in radiation oncology with the example of the treatment design problem. Radiotherapy treatment design as outlined above involves a planner entering a number of plan objectives into the treatment planning system, where the chosen plan objectives may not be the same as those used to evaluate the plan's acceptability by an oncologist, but rather are selected based on the personal experience of the planner [43]. In view of this discrepancy, can Operations Research help assess the acceptability of a treatment design for clinical use, knowing what plans have been accepted in the past? The answer is yes. Data Envelopment Analysis [14] is a linear programming technique that allows the comparison of different "units" based on multiple factors. We consider a treatment plan to be a unit, and we want to compare different plans based on factors related to tumor control (the so-called outputs) and sparing of organs at risk (inputs). Let us consider nine prostate treatments for similar patients (whose anatomy has a similar overlap of around 5%between prostate and rectum). We would like to evaluate which of the plans is most efficient in terms of achieving high tumor control (measured as the dose received by 95% of the tumor volume or D95) with low dose delivered to the rectum (measured as generalized equivalent uniform dose or gEUD). The nine plans are listed in Table 1 with data points comprising their D95 and gEUD values plotted in Figure 4.

Table 1. **Data for nine prostate treatment plans.**

Plan	Tumor D95	Rectal gEUD	Efficiency score θ_i^*
A	71.625	61.428	0.996
B	71.775	61.593	0.998
C	72.025	61.853	1.000
D	71.225	61.369	0.989
E	71.575	61.730	0.990
F	71.515	61.610	0.991
G	71.625	61.934	0.988
H	71.724	62.562	0.980
I	71.325	60.710	1.000

We mention that the use of DEA does not depend on the calculation method of the metrics, in this case D95 and gEUD, as long as the calculations are consistent, which is an assumption we make. From this perspective, the application of DEA can be used on any appropriate pre-computed information that a treatment planner might find valuable.

We notice that point (60.710, 71.000) and point (63.000, 72.025) corresponding to plans C and I are connected by straight line segments that form a frontier enveloping all data points (hence the name). The plot provides evidence that, e.g. plan H is not efficient, as it should be possible to reduce gEUD without impairing D95. An Operations Research approach to compute this efficiency is to formulate and solve the linear program (13) − (17) for each plan i.
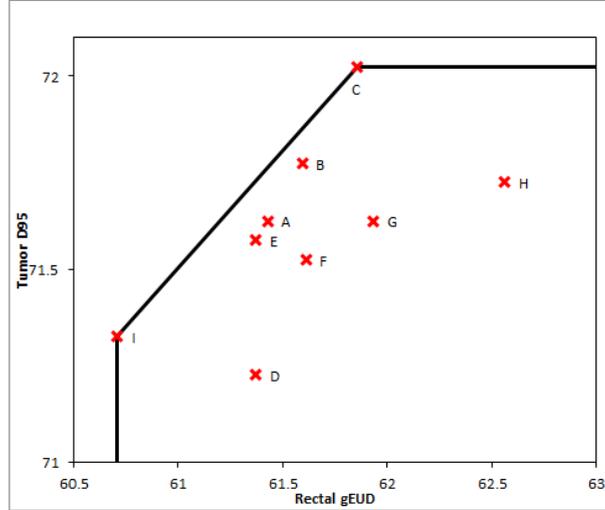
Figure 4.   Plot of the data for nine prostate plans.

$$\min \theta_i \tag{13}$$

$$\text{subject to } \theta_i x^i - X\lambda \ \geq \ 0 \tag{14}$$

$$-z^i + Z\lambda \ \geq \ 0 \tag{15}$$

$$e^T\lambda \ = \ 1 \tag{16}$$

$$\lambda \ \geq \ 0. \tag{17}$$

$$\tag{18}$$

Here, x_i is the vector of input data for unit i, in our case just the single number $gEUD^i$, and X is the matrix whose columns are the vectors x^i. Similarly, z^i is a vector of output values of unit i, here $D95^i$, with the matrix of columns z^i being denoted Z. The linear programming model calculates the smallest value θ_i such that there is a convex combination λ (see constraint (16) with (17)) of input-output vectors from (X, Y) such that the combined inputs are less than or equal than the inputs of unit i (14), with the combined outputs being at least as large as those of unit i (15). In other words, it looks for treatment plans that indicates the possibility of a reduction of rectal gEUD without a reduction of tumor D95.

If the optimal value θ_i^* is equal to 1, no simultaneous improvement is possible and the unit is deemed efficient, indicating that there is no evidence within the nine plans that it could be further improved, whereas otherwise, i.e. if the optimal value $\theta_i^* < 1$, it is not efficient, indicating that it should be possible to find a plan that delivers

less dose to the rectum without reducing D95. This suggestion is derived from the existence of plans C and I: A convex combination of plans C and I on the frontier at level D95 = 71.725 would have a lower 2% lower rectal gEUD than plan H. Thus the method also suggests which plans to consult when trying to improve plan H. A more detailed description of using DEA to evaluate treatment plans is found in [35].

We close this section with a brief discussion of recent uses of convex optimization in the literature. Both linear and quadratic models have been suggested to optimize fluence, see [20] as a review. Most of these models are extended versions of (5). Additional techniques are foundin [12, 39, 50, 61], which adapt probabilistic measures to control dose. These models are discussed in Section 4. Deviation problems are also used for image alignment and comparison [48], and have also been used for vault design [37].

3.   Discrete Problems

Discrete optimization problems arise if the feasible region X in (2) is discrete, i.e. finite or countable. Typically, this means the decision variables are restricted to take only integer values. More precisely, optimization problems with only 0-1 variables and only integer variables are called binary and integer optimization problems, respectively. Some optimization problems contain both continuous and integer variables and are called mixed integer optimization problems. Integer variables are used if modeling entities can only occur in discrete amounts, and binary variables model yes/no decisions and are particularly versatile for modeling logical statements. For example, the statement "if option A is selected then option B must be selected" translates to $x_A - x_B \leq 0$ for binary decision variables x_A and x_B that take a value of one if option A, respectively B, is selected and zero otherwise. Binary variables also allow counting by summation and can be used as "master" variables to control others.

Discrete optimization problems are harder to solve than convex optimization problems because the tools of convex optimization are no longer available due to the fact that the feasible region is no longer convex. This drawback is severe if the objective function f or the constraints g are nonlinear, however, many problems that appear in applications have linear objectives and constraints. Hence, we introduce this important problem class by considering problems with linear constraints and objective functions. We let $f(x) = c^T x$, where c is a n-vector called the cost vector, and $g(x) = Ax - b$, where A is a $m \times n$-matrix and b is a m-vector. The optimization problem

$$\min\{c^T x : Ax \leq b, \ x \text{ integer}\}$$

is called an integer program (IP).

There are two main strategies to solve integer programming problems, namely branch and bound algorithms and cutting plane algorithms. Branch and bound algorithms follow a "divide-and-conquer" strategy. A division

of the feasible set X is a set $\{X_1, \ldots X_s\}$ of subsets of X such that $X = X_1 \cup X_2 \cup \ldots, \cup X_s$. The optimal solution of the original problem must be the best of the optimal solutions of the subproblems $\min\{c^T x : x \in X_i\}$. This subdivision scheme is applied recursively and constitutes the branching part of the algorithm. It can be visualized in a branch and bound tree, with nodes representing subproblems and branches representing the division of a problem into subproblems. The recursion stops whenever a) a subproblem is infeasible, i.e. $X_i = \emptyset$, b) an optimal (integer) solution for the subproblem is known, or c) the optimal (integer) solution of the subproblem is guaranteed to be worse than the optimal solution of the original problem. The branching part and the bounding in c) can be implemented in many ways and are very often designed specifically for a particular problem.

The most common type of branch and bound is linear programming based branch and bound. In this strategy, to solve an integer program $\min\{c^T x : Ax \le b, x \in Z^n\}$, the integrality constraints are initially omitted (relaxed). The resulting linear program is solved, which gives a lower bound on the value of the IP. Next, a variable that has a fractional value in the optimal solution is chosen and two subproblems are created, one with the added constraint that the variable has to be less than or equal to its current value rounded down to the next lower integer, one with the added constraint that the variable has to be greater than or equal to its current value rounded up to the next higher integer. This branching partitions the feasible region of the IP into two disjoint subsets. Imposing such bounds recursively eventually results in the discovery of integer feasible solutions. Once a feasible solution to the original IP is known, it is possible to check condition c) by comparing its value with that of any optimal solution of a subproblem. If the latter is bigger, then no further subdivision of that subproblem needs consideration because an optimal solution cannot be found in the subproblem. Since the LP relaxation of any subproblem is a lower bound on its optimal value, it is sufficient to check whether the optimal value of the LP relaxation is larger than the value of the best known feasible solution (the incumbent).

Cutting plane algorithms also start by solving the LP relaxation of the IP. If this leads to a fractional optimal solution, new constraints are added to the problem, which is re-solved. At least one of the added constraints must be violated by the current optimal solution (which is therefore "cut off" from the feasible set, hence the name cutting plane), yet all integer solutions remain feasible. This process continues until an integer optimal solution is found. In order to be effective, the selection of constraints to be added is crucial. Modern solvers for IPs can automatically generate many types of constraints, but for many integer programming problems, specific classes of constraints are derived to successfully solve problems of relevant size.

It is possible, and often necessary, to combine branch and bound algorithms with cutting plane algorithms, which yields branch and cut algorithms. Here cutting planes are introduced at every node of the branch and bound tree, i.e. in every subproblem.

For very large integer programs it may be impossible to include all variables in the model from the start. Linear programming relaxations can then be solved with subsets of variables, and new variables can be generated when necessary. Whether an additional variable contributes to the improvement of the objective function can be determined by calculating its reduced cost, which measures the rate at which it will contribute to the objective function. This step is called "pricing" variables. The inclusion of procedures to generate columns in branch and bound or branch and cut algorithms is called branch and price, and respectively, branch and cut and price. We refer readers to [36, 60] for a thorough introduction to integer programming.

In Radiation Oncology (mixed) integer optimization models appear mainly in three areas: 1) the beam selection problem, 2) the fluence map optimization problem in order to model dose volume constraints, and 3) the segmentation problem. We explain each of these in some detail below. A fourth problem related to Radiation Oncology where integer programming can be used is the scheduling of treatments and the handling of patient wait lists. This is a more traditional management type of application of Operations Research that has received relatively little attention [13, 30, 42]. The work in [13] presents a model to schedule treatments over a period of six days with the objective to maximize the number of new patients starting treatment (and thus reduce the waiting list) and the sum of booked appointments.

A thorough discussion and survey of existing literature on the beam optimization problem can be found in [22]. Here, we show how to incorporate beam angle optimization in fluence map optimization. In the beam selection problem we assume that the gantry's possible locations are listed in a set $\mathcal{A}$ of positions/angles (e.g. in 360 stops on a full circle around the isocenter of the PTV in a coplanar treatment), where at most R positions are to be selected for treatment. We index the possible angles by a and define binary variables w_a as follows:

$$w_a = \begin{cases} 1 & \text{if angle } a \text{ is used} \\ 0 & \text{otherwise.} \end{cases}$$

The number of beams to be selected can be controlled by the constraint $\sum_{a \in \mathcal{A}} w_a \leq R$. It is important to note that if the quality of a set of beam angles is measured by the optimal value of a fluence map optimization problem it can be shown mathematically that, under very general and reasonable conditions, using more angles can never degrade a treatment [22].

Other constraints that can easily be modeled are the avoidance of opposing beams a_1 and a_2 by $w_{a_1} + w_{a_2} \leq 1$, minimal spacing of beams by $\sum_{a' \in U(a)} w_a \leq 1$, where $U(a)$ is the set of beams in a neighborhood of a from which only one should be selected. Moreover, if we consider continuous fluence variables, say x, constraints of the form $x_{ai} \leq M w_a$ ensure that the fluence of any bixel (a, i) of beam a is 0 whenever that beam is not selected to be used for treatment, i.e. $w_a = 0$. The optimization of beam angles can be easily included in any

such fluence problem, and as an example, the AMPL models of Figures 1, 2 and 3 only need the few additional lines shown in Figure 5. We remark that the solutions of our fluence map optimization example in Section 2 only used one or two angles anyway, so that the addition of the beam selection constraints would have had no effect.

Beam Angle Optimization

```
var w {ANGLES} binary;

param R;
param M;

subject to Numangles: sum {a in ANGLES} w[a] <= R;
subject to Switch-off {(a,i) in PENCILS}: x[a, i] <= M * w[a];
```

Figure 5. **AMPL code for including beam angle optimization in fluence map optimization models.**

Dose volume constraints typically impose that at most $q\%$ of the volume of an organ at risk may receive a dose greater than p Gy. From an OR perspective such a constraint is naturally modeled using integer variables. The discretization of the patient volume into voxels (assumed to be of equal size) and the fact that dose is only calculated in a finite number of points, one per voxel, makes it possible to determine a volume by counting the dose points within that volume. Hence if we define

$$y_v = \begin{cases} 1 & \text{if voxel } v \text{ receives more than } p \text{ Gy} \\ 0 & \text{otherwise,} \end{cases}$$

then we can model the dose volume constraint by enforcing the constraints

$$\sum_{v \in OAR} y_v \leq \frac{q}{100}|OAR| \quad \text{and}$$
$$(Ax)_v \leq p + U_v y_v \quad \text{for all } v \in OAR.$$

Here OAR is the set of voxels in the organ at risk and $|OAR|$ is the number of voxels in the OAR. The first constraint counts the voxels in the organ at risk having $y_v = 1$ and ensures that these are at most $q\%$ of the voxels in the organ at risk. The second set of constraints ensures that indeed only voxels receiving more than p Gy are counted in the first constraint. Here U_v is the largest dose any voxel in the OAR can receive. Once again, these constraints are easily incorporated in a fluence map optimization model. The AMPL code for including a dose volume constraint on the brain stem in the example of Section 2 is given in Figure 6. The literature on dose volume constraint models is discussed in more detail in Section 2.4 of [20].

Observing that the optimal solution found by the infinity norm model of Figure 3 in our example treats 13 of the 33 brain stem voxels with more than 50 Gy, we have added a constraint that at most 20% of the brain stem is

Dose Volume Constraint

```
var y {BRNSTMVOXELS} binary;

param U {VOXELS};

param P;
param Q;

subject to VolumePercent:
      sum {v in BRNSTMVOXELS} y[v] <= (Q/100) * card{BRNSTMVOXELS};
subject to DVC {v in BRNSTMVOXELS}:
      sum {(a,i) in PENCILS} A[v,a,i] * x[a,i] <= P+U*y[v];
```

Figure 6. **AMPL code for including a single dose volume constraint in fluence map optimization models.**

to receive 50 Gy or more. This is achieved by reducing the total monitor units to 448.39, leaving 6 brain stem voxels to receive 50 Gy or more, with a maximal does of 51.82 Gy, but under dosing the target by up to 6.92 Gy.

The third area where integer programming models arise is the problem of segmentation of fluence maps for delivery of a treatment using a multileaf collimator (MLC) in step and shoot mode. The (real) values of a fluence map are usually discretized to a small number of fluence values. These discretized values define an integer intensity matrix I. It is then necessary to decompose I into a number of apertures for the MLC (configurations of the MLC leaves) in such a way that some objective is minimized. Mathematically, the integer matrix I is written as a weighted sum of 0-1 matrices, where the 0-1 matrices represent feasible apertures and the weights represent monitor units or beam-on time for the aperture. The objective is to minimize the sum of the weights (beam-on time), the number of apertures used (as a measure of set-up time), or the sum of the weights plus a scaled number of apertures (measuring total treatment time). We illustrate the problem with an example from [21].

Example 3.1.

$$
\begin{pmatrix} 3 & 6 & 4 \\ 2 & 1 & 5 \end{pmatrix} = 1 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix} + 2 \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} + 4 \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}
$$
$$
= 3 \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} + 1 \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} + 1 \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} + 1 \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}
$$

The first decomposition uses 3 apertures with a beam-on time of 7, whereas the second uses 4 apertures with a beam on time of 6.

This problem permits a variety of integer programming formulations, depending on the variables used. One may use binary variables to model exposed bixels, i.e. ones in the matrices defining apertures together with (real or) integer variables to model monitor units [59]; binary variables to model leaf positions together with binary

variables to model individual monitor units [33]; binary variables indexed by beam-on time to model the beam-on time of individual bixels of each aperture [59], or variables counting the number of apertures and bixels receiving a certain number of monitor units [3].

We discuss here the counter model, which follows the latter approach. Let n_b be a nonnegative integer variable indexed by monitor units b that counts the number of apertures receiving b monitor units. Let q_{ijb} be a nonnegative integer variable that counts the number of apertures with cell (i, j) exposed for b monitor units, and let s_{ijb} be a nonnegative integer variable that counts the number of apertures with cell (i, j) exposed for b monitor units in excess of those apertures having cell $(i, j-1)$ exposed for b monitor units. Then the decomposition problem is defined by the following constraints:

$$
\begin{aligned}
\sum_{b=1}^{b_{max}} b q_{ijb} &= I_{ij} \text{ for all } i, j \\
\sum_{j=1}^{n} s_{ijb} &\leq n_b \text{ for all } i, b \\
s_{i1b} &= q_{i1b} \text{ for all } i, b \\
s_{ijb} &\geq q_{ijb} - q_{i(j-1)b} \text{ for all } i, b, \text{ and } j \geq 2.
\end{aligned}
\tag{19}
$$

The parameter $b_{\max}$ is an upper bound on the beam-on time any aperture is allowed, which can be set to the maximum element of I. The first constraint ensures a decomposition of I into apertures. The second ensures that valid apertures are defined, i.e. all ones in the matrices appear in a single block in each row, and the last two guarantee the correct relationship between the q and s variables. In fact, it can be seen that the last three constraints together are the same as $\sum_{j=1}^{n} \max\{q_{ijb} - q_{i(j-1)b}, 0\} \leq n_b$ for all i, b (assuming that $q_{i0b} = 0$). However, this form of the constraints is nonlinear, whereas (19) disaggregates this into linear constraints. Alternative objective functions are

$$
\min \sum_{b=1}^{b_{max}} b n_b \quad \text{(minimize beam-on time)},
$$

$$
\min \sum_{b=1}^{b_{max}} n_b \quad \text{(minimize the number of apertures)},
$$

or a combination of both.

We have seen above that integer programming models for a variety of problems in Radiation Oncology can be easily formulated. This discussion may have generated a few questions: If incorporating beam angle optimization is so "simple", why is it not widely used? Why are there many different models for the segmentation problem? Shouldn't any one of them suffice? We try to address some of these questions below.

In the Operations Research community integer programming is a huge subfield. Its importance lies in the versatility of the modeling power of integer and binary variables. This versatility comes at a price since integer variables destroy convexity. So, while solution methods apply linear programming techniques to solve subproblems, which have to be solved repeatedly, solving integer programs is considerably more demanding computationally than solving convex optimization problems. We highlight a few of the computational problems.

1. Although the number of possible solutions of an IP is finite, it can be astronomically large. We use the beam selection problem as an example. If 5 beam angles are to be selected out of a candidate set of 360, then there are 4.9×10^{10} possible combinations. Any algorithm to solve a combined fluence map/beam angle optimization problem needs to implicitly enumerate all these combinations. Recalling that a fluence map optimization problem is solved for each selection of beam angles, the enormity of the problem size becomes evident, even acknowledging the fact that many or most combinations will never be explicitly considered due to bounding techniques. Thus, while IP models are easily formulated they just as easily become too large to be solved with current off-the-shelf hardware and software.

2. Integer programs are difficult to solve if there are few feasible solutions or many optimal solutions. In the first case, the solver spends a lot of computation time to find a feasible solution. Often, once a feasible solution is discovered, an optimal solution can be reached quickly. If there are many optimal solutions it may take a long time to prove optimality because many solutions need to be checked against the incumbent, optimal solution.

3. Not all IP formulations of a problem are equally good. One important issue is the tightness of the LP relaxation, i.e. how close the value of an optimal solution of the LP relaxation is to the optimal value of the IP. The tighter, the better, because better lower bounds allow more sub-problems to be discarded due to bounding. Such strong and weak formulations sometimes differ in only a minor aspect of the model. An indication of a "bad" formulation is the use of "big M constraints" such as $x_{ai} \leq Mw_a$ in the beam angle optimization models. From a modeling perspective, any large M will do. However, large numbers distort the numerics of LP solutions and should be avoided. Dose volume constraint models and many MLC segmentation models also contain big M constraints [34].

   Another issue is symmetry. This relates to the fact that some models allow the same solution to be feasible in different parts of the branch and bound tree. This is an undesirable property because it severely limits the power of the bounding step. A case in point is the variety of IP models for the MLC segmentation problem. The early model of [33] cannot be solved for instances of clinically relevant size even with modern computers and optimization software. It took several groups of researchers years to discover tractable models, and those of [3, 24, 57] appear to be reaching clinically relevant problem sizes.

4. Lastly, not all problems have the same difficulty. There is a subclass of problems that is easy to solve, meaning that efficient algorithms to solve them are known and even large instances can be solved. On the

other hand, there is a big class of problems, known as NP-hard problems, for which no such algorithm is known and is unlikely to exist. All known algorithms for these problems require computation time that increases exponentially with the size of the problem. Hence problems of clinically relevant size are difficult to solve in reasonable time. It is not trivial to distinguish the two classes. For example, it is known in the OR literature that the problem of minimizing beam-on time in MLC segmentation is "easy", i.e. solvable in polynomial time. In fact, the sweep algorithm [7] proposed in the medical physics literature was later proved by OR researchers to solve this problem optimally and in linear time [1]. If, however, the objective is to minimize the number of apertures, the problem becomes NP-hard (it even belongs to a particularly difficult subclass of these problems). That is precisely the reason why this problem has become popular in the OR community.

To summarize, the application of discrete optimization methods in Radiation Oncology requires careful modeling of the problems as well as care in the design of algorithms to solve those problems. Unlike in the case of convex optimization problems it is not possible to separate the two. This shows that close collaboration between clinical practice and Operations Research is a way to succeed.

4. Other Problem Types

The previous sections introduced two of the main problem types in deterministic OR, but the taxonomy of problems mentioned in Section 1 is substantially larger, and many problems in Medical Physics appropriately fit in problem classes outside those that have been discussed. Here we introduce conic programming, dynamic programming, constraint programming, and global optimization. Each subsection points to applications in Medical Physics, resources to learn more about each problem class, and software to model and/or solve problems.

4.1. Conic Programming

The idea behind conic programming is to extend the nonnegativity constraints of a standard linear program. A *cone* is a set of elements, say K, so that if x is in K, then λx is in K for any positive scalar λ, and a conic program looks like

$$\min\{c^T x : Ax = b, x \in K\}.$$

If K is the collection of nonnegative vectors, then this becomes a standard-form linear program. However, this problem class is significantly larger since we can embed any feasibility condition. This is due to the realization that if we have a general feasible set, say X, then $K = \{(\lambda, \lambda x) : x \in X, \lambda \geq 0\}$ is a cone that equates to X if $\lambda = 1$. With this cone, we see that

$$\min\{c^T x : x \in X\} = \min\{c^T x : \lambda = 1, (\lambda, \lambda x) \in K\},$$

which demonstrates the modeling flexibility permitted by this problem class.

Some cones tend to be more important than others in applications, and we present a model that has had wide practical appeal due to its modeling flexibility and due to its ability to be solved efficiently. Specifically, we consider the second order cone, which is the collection of n-vectors that for some j satisfy

$$\sqrt{\sum_{i \neq j} x_i^2} \leq x_j.$$

Second order cone programs (SOCPs) are particularly useful in radiotherapy because they allow us to consider the inherent clinical variations that arise during treatment design and delivery. This fact was first observed in [12], which was motivated by the delivery errors introduced from fractionating the treatment, and it was subsequently used in [39], which was motivated by variations between dose models (see [61] for related work on using SOCPs to model dose volume constraints). From a modeling perspective the idea is to consider the dose as one of several possibilities, and the goal is to minimize the objective subject to all the constraints that explain the range of possible outcomes.

We consider a small example to illustrate the technique. Suppose the unit dose delivered by two beams varies due to patient movement, realignment, and other delivery factors. Let the fluence variables be x_1 and x_2, and for the sake of simplicity, assume there are only two possible dose considerations and two voxels. We model the problem with the following data,

$$\begin{bmatrix} A_1 \\ \underline{} \\ A_2 \end{bmatrix} = \begin{bmatrix} 0.5 & 0.3 \\ 0.45 & 0.28 \\ \underline{} \\ 0.3 & 0.4 \\ 0.31 & 0.42 \end{bmatrix}, \ p = \begin{pmatrix} 1/4 \\ 3/4 \end{pmatrix}, \text{ and } P = \begin{bmatrix} 1/4 & 0 \\ 0 & 3/4 \end{bmatrix}. \tag{20}$$

The rows of the unit dose matrix on the far left are separated into two submatrices, A_1 and A_2. The first row of A_1 explains one of the two dose scenarios for the first voxel and shows that in this scenario the first voxel receives 0.5 Gy from beam 1 and 0.3 Gy from beam 2. The second row describes the second scenario, in which voxel 1 receives 0.45 Gy from beam 1 and 0.28 Gy from beam 2, assuming one monitor unit. The probability of the two scenarios is respectively 1/4 and 3/4 as listed in p, and the matrix P is conveniently the diagonal matrix corresponding to p. Suppose the upper bounds on the dose to voxels 1 and 2 are respectively $b_1 = 20$ and $b_2 = 5$. The task is to model a constraint for each voxel that restricts the probability of a dose violation, i.e. we ask that $\text{Prob}(D_v(x) > b_v) < \delta$, where δ is a measure of an acceptable risk of a dose violation. The development of such a constraint is found in [12], with the resulting form being

$$\|P^{1/2}(I - ep^T)A_i x\|_2 \leq \frac{1}{z\sqrt{N}}\left(b_i - Np^T A_i x\right),$$

where e is a vector of ones, N is the number of treatment fractions ($N = 1$ here), b_i is voxel i's upper bound, and z is the $1 - \delta$ percentile of the normal distribution with mean 0 and variance 1 (large values of z reduce the probability of violation). For the first voxel in our example with a single fraction and $z = 3$, this results in

$$\left\| \begin{bmatrix} 1/2 & 0 \\ 0 & \sqrt{3}/2 \end{bmatrix} \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 1/4 & 3/4 \\ 1/4 & 3/4 \end{bmatrix} \right) \begin{bmatrix} 0.5 & 0.3 \\ 0.45 & 0.28 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \right\|_2$$

$$\leq \left(\frac{1}{3} \right) \left(20 - \begin{pmatrix} 1/4, & 3/4 \end{pmatrix} \begin{bmatrix} 0.5 & 0.3 \\ 0.45 & 0.28 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \right),$$

which after multiplication is

$$\sqrt{(0.019x_1 + 0.008x_2)^2 + (0.011x_1 + 0.004x_2)^2} \leq (1/3)(20 - 0.463x_1 - 0.285x_2).$$

The corresponding constraint for the second voxel is

$$\sqrt{(0.002x_1 + 0.004x_2)^2 + (0.004x_1 + 0.008x_2)^2} \leq (1/3)(5 - 0.308x_1 - 0.415x_2).$$

We mention that although the SOCP constraint for the first voxel is not linear, it is nearly the linear constraint $(0.5(1/4) + 0.45(3/4))x_1 + (0.3(1/4) + 0.28(3/4))x_2 \leq 20$, which ensures that the expected dose to the first voxel is less than 20 Gy. Likewise, the SOCP constraint for the second voxel is nearly the linear constraint $(0.3(1/4) + 0.31(3/4))x_1 + (0.4(1/4) + 0.42(3/4))x_2 \leq 5$. As a minor point of note, this realization highlights that the first SOCP constraint is redundant because any fluence vector x satisfying the second SOCP constraint also satisfies the first.

We incorporated SOCP constraints into the acoustic neuroma case introduced in Section 2. Three scenarios were considered. The first assumes that each dose calculation over estimates the actual dose by 10^{-4}, the second assumes that each dose calculation is correct, and the third assumes that the dose calculation under estimates the actual dose by 10^{-4}. Allowing A to be the dose matrix so that $D(x) = Ax$, where as before a subscript can be used to indicate dose to the indicated structure, we find that dose is $D(x) = (A - 10^{-4}ee^T)x$ for the first scenario, $D(x) = Ax$ for the second scenario, and $D(x) = (A + 10^{-4}ee^T)x$ for the third scenario (e is again the vector of ones so that ee^T is the matrix with each component being 1). In relation to the notation in (20), we have a matrix with three rows for each voxel v so that

$$Avox_v = \begin{bmatrix} A - 10^{-4}ee^T \\ A \\ A + 10^{-4}ee^T \end{bmatrix}.$$

The probabilities vector was $p = (1/4, 1/2, 1/4)^T$, and we placed an upper bound on the eye socket of 5 Gy and an upper bound on the brain stem of 45 Gy. We assumed 30 fractions and set the $1 - \delta$ percentile to $z = 3$, which guaranteed the probability of a dose violation was at most 0.3%. Specifically, we solved

$$\min\{\|p^T A x - T\|_1 :$$
$$\|P^{1/2}(I - ep^T)A_v x\|_2 \leq (5 - 30p^T A_v x)/3\sqrt{30}, \ v \in \text{EYEVOXELS},$$
$$\|P^{1/2}(I - ep^T)A_v x\|_2 \leq (45 - 30p^T A_v x)/3\sqrt{30}, \ v \in \text{BRNSTMVOXELS},$$
$$Ax \geq 60, \ x \geq 0\}.$$

The AMPL code describing the variables, objective and constraints is located in Figure 7. We used SNOPT to find a solution through the Kestrel interface[1]. The optimal solution satisfies all the constraints and returns a treatment with the PTV receiving between 60 and 65.03 Gy. Opposing angles at $60°$ and $240°$ were used with respectively 220.185 and 161.561 monitor units.

SOCP

```
var x {PENCILS} >= 0;
var ScenarioDose {SCENARIOS, VOXELS} >= 0;
var z;

minimize WeightedDeviation: z;

subject to SOCPUpBounds {v in OARVOXELS}:
    sum {sr in SCENARIOS} (sum {sc in SCENARIOS} R[sr,sc]*
            (sum {(a,i) in PENCILS} Avox[sc,v,a,i]*x[a,i]))^2
        <= ((T[v] - NumFractions*(sum {s in SCENARIOS} p[s] *
            sum {(a,i) in PENCILS} Avox[s,v,a,i]*x[a,i]))) /
            (TreatmentPercentile * sqrt(NumFractions)))^2;

subject to TargetLowerBound {v in TARGETVOXELS}:
    sum {s in SCENARIOS} p[s] * (sum (a,i) in PENCILS Avox[s,v,a,i] * x[a,i]) >= T[v];

subject to TargetUpperBound {v in TARGETVOXELS}:
    sum {s in SCENARIOS} p[s] * (sum {(a,i) in PENCILS} Avox[s,v,a,i] * x[a,i]) <= T[v] + z;
```

Figure 7. AMPL code for an SOCP model that calculates an optimal fluence. The R matrix in SOCPUpBounds is $P^{1/2}(I - ep^T)$, which is calculated earlier in the code.

4.2. Dynamic Programming

Dynamic programming (DP) is concerned with time-dependent decisions that are based on an optimal process. In some way DP bridges two of the main sub-disciplines in optimization, those being mathematical programming

[1] *Kestrel allows AMPL to use a host of the solvers supported by NEOS, of which SNOPT is one. See* http://www-neos.mcs.anl.gov/ *for further information.*

and control theory. The essential difference is the type of vector space that is used to define the optimization problem. In mathematical programming, which is the topic of the previous sections, the decision variables are restricted to subsets of the real line. In control theory the unknowns are instead functions with appropriate analytic properties. If time is continuous, then the study of DP lives within variational calculus and is more like control theory since it is defined over a collection of functions. If time is discrete, then DP is more like staged mathematical programming, which is the case that we address here. Interested readers are directed to [41] for more information.

Since radiotherapy treatments are delivered over several fractions, the issue of time dependent decisions arises naturally, and Deng and Ferris [17] investigate the question of adapting the fractional doses by solving a stochastic DP. To keep with the introductory nature of this article, we present a deterministic version to introduce DP. Adapting our earlier notation, we let $D^k(x)$ be the vector of doses after the kth fraction has been delivered, where x is again a vector of fluences. If each fraction is uniform, then $D^k(x) = D(x)/K$, where K is the total number of fractions. For simplicity we only consider dose and not, at least directly, its dependence on fluence. As such, we remove the notational dependency on x and let D^k be the vector of doses indexed by voxels at the kth fraction. Alignment errors are modeled by linearly adjusting the dose at each fraction, and this linear adjustment is explained by a stochastic matrix W^k at each fraction. As an example, suppose there are only 4 voxels and that

$$
W^k = \begin{bmatrix} 3/4 & 1/5 & 0 & 0 \\ 1/4 & 3/5 & 1/7 & 0 \\ 0 & 1/5 & 6/7 & 1/3 \\ 0 & 0 & 0 & 2/3 \end{bmatrix}.
$$

This matrix indicates that each Gray targeted at the first voxel is actually distributed to the first two voxels, with the first voxel receiving 3/4 of the Gray and the second receiving 1/4. Each column has a similar interpretation. If the patient had been aligned perfectly, then the matrix would have been the identity. Assume the uniform fractional dose is supposed to be $d = (1, 2, 4, 0)^T$. Then the adjusted dose is

$$
W^k d = \begin{bmatrix} 3/4 & 1/5 & 0 & 0 \\ 1/4 & 3/5 & 1/7 & 0 \\ 0 & 1/5 & 6/7 & 1/3 \\ 0 & 0 & 0 & 2/3 \end{bmatrix} \begin{pmatrix} 1 \\ 2 \\ 4 \\ 0 \end{pmatrix} = \begin{pmatrix} 1.15 \\ 2.02 \\ 3.83 \\ 0.00 \end{pmatrix}.
$$

In this situation the first and second voxels are over treated and the third is under treated. Instead of delivering the uniform fractional dose, we would rather find a dose vector u^k, called a *control* in the study of DP and control theory, so that $W^k u^k = d$. A solution is $u^k = (0.8, 2.0, 4.2, 0)$, and if these doses are delivered during the kth fraction, then under the patient shift indicated by W^k the desired uniform fractional dose d is correctly

delivered. An important observation is that we are not necessarily able to find a control that always delivers the intended dose. To see this, note that if d had been $(2.8, 0.69, 1.85, 0.67)^T$, then the only solution to $W^k u^k = d$ would have been $u^k = (4, -1, 2, 1)^T$, which is impossible since dose can't be removed from the second voxel. Hence it is impossible to achieve the intended dose and some error is realized.

The central theme of control theory is to find a collection of controls that optimizes a measure of their effectiveness. The kth control is selected from a set of possible controls appropriate for the kth state, denoted by U^k. For this problem the kth state is the cumulative dose after the kth fraction, i.e. the kth state is D^k. Consecutive states are linked by the associated control u^k, and the dynamic iteration is

$$D^{k+1} = D^k + W^k u^k. \tag{21}$$

In the language of DP the problem is to select controls, $u^1, u^2, \ldots, u^k$, that optimize treatment delivery. There are many measures that could be used to assess delivery, and our presentation is based on

$$\sum_{k=1}^{K} \left\| W^{k-1} u^{k-1} - \frac{1}{K - (k-1)} \left(T - D^k \right) \right\|_1,$$

This objective aggregates the errors between each fraction's delivered dose and an updated uniform dose, which is the difference between the optimal dose and the cumulative dose through fraction k. Allowing

$$f(D^k, u^{k-1}) = \left\| W^{k-1} u^{k-1} - \frac{1}{K - (k-1)} \left(T - D^k \right) \right\|_1,$$

we aim to solve the following DP

$$Y(D^0) = \min \left\{ \sum_{k=1}^{K} f(D^k, u^{k-1}) : D^{k+1} = D^k + W^k u^k, \ u^k \in U^k, \ k = 0, 1, 2, \ldots, K \right\}. \tag{22}$$

The value of this minimum, denoted by $Y(D^0)$, is dependent on the initial state D^0, which is used to decide the initial control u^0. The set U^k is used to guarantee that each stage's control is realistic, with one obvious restriction being that $u^k \geq 0$. Other restrictions, such as ensuring there is a fluence that can achieve the dose, can be included. For example, U^k could be $\{u : Ax = u, \ u \geq 0\}$, where A is the dose matrix so that $D(x) = Ax$.

Solving (22) is not trivial, but there is a marvelous technique that allows us to consider the problem in stages. The method, which was developed by Bellman [5], shows how to establish a recurrence relation between the optimization problems associated with consecutive states. First, notice that u^1 is a function of the first control u^0 and the initial state $D^0 = 0$, which is zero since no dose has been delivered prior to the first fraction. Assuming (21) holds, we can streamline the presentation and re-write (22) as

$$\min_{u^0 \in U^0} \left\{ f(D^0 + W^0 u^0, u^0) + \min \left\{ \sum_{k=2}^{K} f(D^k, u^{k-1}) : u^k \in U^k, k = 1, 2, \ldots, K \right\} \right\}$$

$$= \min_{u^0 \in U^0} \left\{ f(W^0 u^0, u^0) + Y(D^1) \right\} = \min_{u^0 \in U^0} \left\{ f(W^0 u^0, u^0) + Y(D^0 + W^0 u^0) \right\}. \tag{23}$$

The importance of this observation is that deciding an initial optimal control can, in some way, be separated from deciding the following controls. The inner minimization problem in the first expression is a function of D^1 since this is the state from which control u^1 is decided. This fact is denoted in the next equality, and since D^1 is dependent on the prior state and control, we conclude in the final equality by expressing the problem of deciding the initial control as the minimization of the sum of two functions.

The relationship between the first two states in (23) can be continued. As an example, the case with three fractions is

$$Y(D^0) = \min_{u^0 \in U^0} \left\{ f(D^1, u^0) + \min_{u^1 \in U^1} \left\{ f(D^2, u^1) + \min_{u^2 \in U^2} \left\{ f(D^3, u^2) \right\} \right\} \right\}. \tag{24}$$

This expression is used to solve DPs, with the two standard methods being forward and backward techniques, although there are several variations, see [41, 44] for further details. For simplicity of presentation, consider the above case of three fractions and assume that $U^k = \{u', u''\}$ for each k. This collection of controls is not unrealistic since u' could represent no adjustment and u'' could represent a pre-calculated adjustment to a known alignment error. In this case, we are trying to decide whether or not to adjust the patient on any particular fraction.

A schematic depicting the problem of solving the DP is shown in Figure 8. In this figure there are two possible dose states after the first fraction, $D^1(u')$ or $D^1(u'')$, which depend on the control used to deliver the first fraction. Similarly, there are two possible dose states following the second [and third] fraction, $D^2(u')$ and $D^2(u'')$ [$D^3(u')$ and $D^3(u'')$]. Notice that to achieve the dose $D^2(u')$ we could have used control u' to deliver the first and second fractions, a course of treatment represented by the path from D^0 to $D^1(u')$ to $D^2(u')$, or we could have alternatively used control u'' in the first fraction and control u' in the second, a possibility represented by the path from D^0 to $D^1(u'')$ to $D^2(u')$. Each edge is assigned the value $f(D^k(u'), u')$ or $f(D^k(u''), u'')$ depending on which control is considered. So, the value of 2 on the edge from D^0 to $D^1(u'')$ shows that if we use control u'' to deliver the first fraction, then the maximum voxel deviation from the (updated) uniform fraction is 2 Gy, whereas the value of 1 on the edge between D^0 and $D^1(u')$ shows that the maximum voxel deviation is 1 if control u' is used instead. This may appear as though it is better to use control u' to deliver the first fraction, but this may not be the case depending on the deviations imposed by this choice on the subsequent fractions.

To solve the DP with a forward technique we first consider the two possible dose states after the first fraction. Since there is no choice in how to achieve these states from D^0, the edges between D^0 and $D^1(u')$ and between D^0 and $D^1(u'')$ are included as part of a possible optimal sequence of controls, which is illustrated in Figure 9 by thickening the edges. In relation to (24), this considers both possible values of $f(D^1, u^0)$.

We continue by moving to the second fraction. If we select u', then we realize a maximum voxel violation of either 3, in the case that u' was used for the first fraction, or 1, in the case that u'' was used for the first fraction.
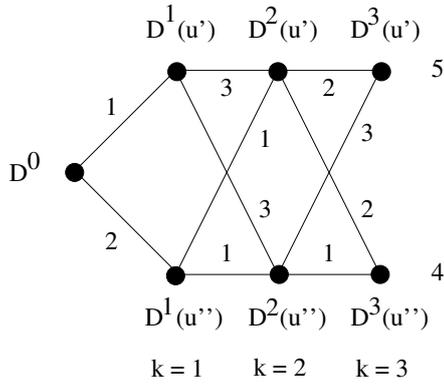
Figure 8. A depiction of the possible control sequences for dynamic programming.
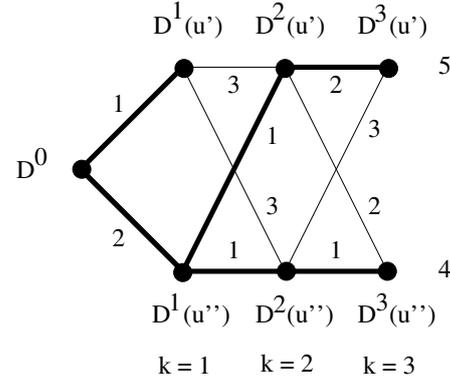
Figure 9. The optimal sequences to the possible dose states are indicated by the thicker edges.

This gives a cumulative, maximum voxel violation of either 4 or 3, and since 3 is smaller, we know that **if** the optimal control sequence includes using u' to deliver the second fraction, then we should use u'' in the first fraction. This information is shown in Figure 9 as a thick edge between $D^1(u'')$ and $D^2(u')$. Notice the edge between $D^1(u')$ and $D^2(u')$ remains thin since it is not possible to use u' for the second fraction after using u' for the first. A similar argument shows that if we use u'' for the first fraction, then we should use u'' for the second since this gives a cumulative, maximum violation of 3 instead of 4. Returning to the mathematical description in (24), we have decided which control u^0 should be depending on which control we use to deliver the second fraction. Importantly, we have considered all 4 possible values of $f(D^1, u^0) + f(D^2, u^1)$ and discarded two, those for which $u^0 = u'$ and $u^1 = u'$, and $u^0 = u'$ and $u^1 = u''$.

The solution procedure terminates by repeating the analysis for the third fraction. From the previous paragraph we know that the smallest cumulative, maximum violation after the second fraction is 3 at either $D^2(u')$ or $D^2(u'')$. The least additional violation that can be achieved to reach $D^3(u')$ is 2, and the edge between $D^2(u')$ and $D^3(u')$ with this value is thickened. Likewise, we include the edge between $D^2(u'')$ and $D^3(u'')$ since this gives the smallest cumulative, maximum violation of 4 at $D^3(u'')$. Based on our metric, we would select the control sequence u'' for each fraction since this minimizes the aggregate of the maximum voxel violations over the course of treatment. The cumulative dose may or may not be the same at $D^3(u')$ under the control sequence u'' for the first fraction and u' for the second and third fractions. Although this sequence has a higher cumulative, maximum deviation, these violations could occur in different voxels, and hence, it is possible the anatomical dose is the same. Indeed, the dose may in some ways be better. This observation might seem odd since the integral dose is what is evaluated in treatment design. However, the delivery problem is trying to best match the expected uniform fractionated dose over the entire treatment, and since the uniform fractions are important radiobiologically, this is reasonable.

The backwards techniques are nearly identical except that they propagate through the stages in reverse order. The power of both methods is that they forgo a search over all possible control sequences. In the example above, only 2 control sequences are identified out of the possible 8. The other 6 are discarded since they are identified as being incapable of being optimal.

Returning our focus to the clinical problem of adjusting the fractional doses to optimize delivery, we note that the above model assumes a priori knowledge about the adjusted patient coordinates for each fraction, i.e. the adjustment matrices W^k need to be known before the adjusted fractional doses are calculated, which is unrealistic. There are several imaging modalities that support patient relocation while the patient is on the couch, but clairvoyantly forecasting future patient positions is not possible. In this situation it makes sense to consider random patient adjustments in which each element of W^k is assumed to be randomly distributed. Mathematically we can consider any collection of distributions, but of course, in practice they should approximate the relocation process. In such a stochastic program the objective function is a random variable, and it is natural to consider the expected value of the objective. This is what is considered in [17]. Stochastic problems are generally harder to solve than their deterministic counterparts, and the authors of [17] use a technique called *Neuro-Dynamic Programming* to approximately solve the problem. While a full explanation of this method is beyond the scope of this article, the essence of the method lies in a Monte Carlo simulation that estimates the inner minimization at each iteration. The results consistently show that the expected error is reduced by solving a DP as compared to uniform fractions, for which improvements are significant, and to a reactionary regime that re-calculates the next fraction as the remaining dose to be delivered divided by the number of fractions remaining, for which there is less improvement.

Although solving a DP is not as systematic as other optimization problems, the modeling flexibility together with the ability to handle time-dependent decisions lends itself well to many design questions. DP has lead to major computational advances in computational biology and has continued to be a major theme in engineering, but it has not played a significant role in Medical Physics. In the authors' opinion, this is a promising research direction that could assist many clinical problems.

4.3. Constraint Programming

The field of constraint programming spans computer science and mathematical programming and is concerned with the study of satisfying constraints that are stated in a natural and logical manner. Although the focus is on constraint interrogation and satisfaction, such problems include the majority of optimization problems due to the Lagrange conditions (4). This important area of research has not yet made a significant impact on Medical Physics, but since constraint programming is focused on computing, we suspect that there is a wide-range

of applications in Medical Physics. We discuss the recent work in [3] and [8], which shows that constraint programming leads to favorable results to the NP-hard problem of minimizing the number of apertures needed to achieve an intensity map (19). For further reading, we direct interested readers to [16] as an introduction to constraint programming.

Assuming $q_{i0b} = 0$, we find that the last three constraints in (19) can be stated as

$$\sum_{j=1}^{n} \max\{q_{ijb} - q_{i(j-1)b}, 0\} \leq n_b \ \text{ for all } i, b,$$

and it is this aggregated constraint that counts the number of apertures that are used to deliver the treatment. Since minimizing the number of apertures is NP-hard, this constraint is critical to solving a computationally difficult problem, and finding small values of n_b that maintain feasibility is important. This constraint can be decomposed into linear constraints; one such decompositions is presented in (19), and is also used in [3]. Another from [8] is

$$s_{ijb} = q_{ijb} - q_{i(j-1)b}, \ w_{ijb} = \max\{s_{ijb}, 0\}, \ \sum_{j=1}^{n} w_{ijb} \leq n_b, \tag{25}$$

where s_{ijb} is the same as in (19) and w_{ijb} is a newly introduced variable. We want to deduce small values of n_b that subsequently lead to either optimal or near optimal objective values. The third inequality in (25) provides a lower bound on n_b, and hence, we want to search for w_{ijb} so that the summation defining the lower bound is small and so that the overall system remains feasible. This means we want to shrink the domains of the variables in a way that removes non-optimal solutions. The process of reducing variable domains is called *constraint propagation* in constraint programming, and an algorithm that details a specific technique is called a *propagator*.

As a simple example of the deduction process, consider the case of the decomposition in (25) for which i and b are fixed and for which there are three integer variables q_{i1b}, q_{i2b}, and q_{i3b}. Suppose that other design concerns ensure that $q_{i1b} = 0$, $q_{i2b} \in \{0, 1\}$ and $q_{i3b} = 1$. In this situation we deduce that $w_{i1b} = 0$, $w_{i2b} \in \{0, 1\}$ and $w_{i3b} \in \{0, 1\}$, and hence, $n_b \geq 2$. However, q_{i2b} cannot be both 0 and 1, from which we see that either w_{i2b} is 0 and w_{i3b} is 1, or that w_{i2b} is 1 and w_{i3b} is 0. This means $n_b \geq 1$ in actuality. In this case the variable domains/bounds in and of themselves do not propagate to the smallest lower bound, and in such a case we say that the decomposition does not propagate completely. The decomposition in (19) similarly fails to propagate completely.

A propagator to improve the calculation of the lower bounds on n_b is presented in [8]. For each i and b, the propagator performs three passes through the j index. A forward pass calculates a lower bound on n_b that is based on the ranges of each w_{ijb}. This lower bound is denoted by n'_b. A backward pass finds a smallest and largest value of each w_{ijb}, denoted by w'_{ijb} and w''_{ijb} respectively, so that the summation of both the largest and

smallest values achieves the lower bound of n'_b. At this point it is natural, but misleading, to think that we have solved the segmentation problem since we have found a smallest value of each n_b based on the ranges of each w_{ijb}. However, we have not satisfied the the first constraint in (19). This constraint may force w_{ijb} to assume values outside $[w'_{ijb}, w''_{ijb}]$, and the domain reduction pass updates these bounds to reflect this necessity. The result of the propagator satisfies a property called *bounds consistency*, which unfortunately is not uniquely defined [11]. However, in this setting the property means that the lower bounds n'_b and the objective value $\sum_{b=1}^{b_{max}} n'_b$ is possibly optimal since there is an associated feasible solution that provides this objective value. Moreover, the calculation of this value is efficient. The use of this propagator over other decompositions impressively improves computation time, with the results in [8] showing speed-up factors ranging from 2.3 to 23.8. So, even in the least successful case this propagator reduces the solution time by over half of the other methods.

4.4. Global Optimization

Global optimization is the study of problems that may have multiple local optima, and much of the study is about deciding which of the local optima are indeed global optima. Problems can be continuous or discrete. The field is often concerned with solving problems that are not directly approachable with standard and provably optimal techniques from other areas of optimization. For example, minimizing a strictly convex, quadratic function would not be part of the study of global optimization since there are numerous ways to solve such a problem exactly. From this perspective, global optimization problems are among the most difficult to solve since we typically don't have a mathematical certificate of global optimality. However, there are many meaningful global problems, see for example the treatment design problems in [10] and [58] and some of the image registration problems in [48]. Solving these problems is an important area of research, and we point readers to the supplement on Global Optimization found at the Mathematical Programming Glossary [45] and to the Handbook on Global Optimization [40].

Global optimization's impact on Medical Physics has largely been in the area of heuristic search, with simulated annealing and genetic algorithms being popular optimization methods, see for example [10, 18, 26, 29, 56]. Since these methods are generally understood in the Medical Physics literature, we forego a detailed discussion of these heuristics. Moreover, even a shallow presentation of the plethora of methods within the discipline would in itself constitute a complete review article, and our goal is to mention a couple of the central alternatives to simulated annealing and genetic algorithms so that readers can investigate other techniques. We also want to give a few words of caution. Heuristic methods, such as simulated annealing, are in many ways marvelous since they accommodate nearly any optimization problem and are easily implemented. However, they typically do not provide a guarantee of optimality, and while they may calculate good solutions, the lack of such a guarantee eliminates a factual basis of how much better a solution could be. We encourage readers to use exact methods that provide such a guarantee if possible. For example, least-squares problems are convex and can be solved

efficiently with standard quadratic solvers. Simulated annealing may give an equally good solution, but upon termination, the algorithm itself doesn't ensure confidence that the solution is optimal.

One collection of heuristics partitions the domain of the objective into regions and then uses a local search method to find a local minimum. This idea works well if the function has ascertainable analytical properties over the individual regions, a condition not always available. Algorithmic decisions include a method to sub-divide the domain and the selection of the local search method, which can vary from region to region. Another heuristic is tabu search by which moves are forbidden for a few iterations of the algorithm, a paradigm that allows the method to search portions of the domain that may at first appear less promising than others. Tabu search is arguably the most popular meta-heuristic, which is a collection of methods that search through the feasible region. Variations and applications are wide ranging, and we point readers to [25].

The fact that most heuristic methods are tailored to the specific problem makes their implementation difficult to automate. However, there are a few software packages that link to modeling systems like AMPL. Reviews of software packages are found in [47]. We specifically mention LGO [46], which interfaces with a variety of supporting software packages like AMPL, GAMS, MATLAB, and MAPLE. This software was used to design radiotherapy treatments in [58].

5.   Multiple Objective Problems

In this section we focus on optimization problems that contain several objective functions that are to be minimized simultaneously. Hence (2) becomes

$$\min\{(f_1(x), \ldots, f_p(x)) : g(x) \leq 0\}. \tag{26}$$

Multiple objective problems can be convex, discrete, or any other type of optimization problem. We first need to specify what minimization means since feasible solutions may not be comparable due to the fact that each feasible solution is evaluated with a vector of values, e.g. is (1,2) smaller than (2,1) or vice versa? We let $F(x) = (f_1(x), \ldots, f_p(x))$ and introduce the notation of an *efficient solution* (or *Pareto optimal solution*). A feasible solution $\hat{x}$ of (26) is called *efficient* if there does not exist another feasible solution x such that $f_k(x) \leq f_k(\hat{x})$ for all $k = 1, \ldots, p$ and $f(x) \neq f(\hat{x})$. In multiobjective optimization it is useful to consider the set of all feasible outcomes. Recall that $X = \{x \in R^n : g(x) \leq 0\}$ is the feasible region of (26). The function F maps X to $Y = \{F(x) : x \in X\}$, the outcome set of (26). In single objective optimization Y is simply a half-line or an interval and minimization means finding the left endpoint of that line or interval. For optimization problems with $p \geq 2$ objectives, Y exists in a p-dimensional space and there is no such thing as a unique minimum in a p-dimensional set. All outcomes on the lower left boundary of Y are incomparable and share the property that

Y does not contain any other outcome that is located both to the left and down from the point. We call those outcomes nondominated outcomes of the multiobjective optimization problem (26). An efficient solution $x \in X$ defines a *nondominated outcome* $y = f(x) \in Y$.

The above discussion shows that solving a multiobjective optimization problem means finding efficient solutions of (26). This can be interpreted as finding all efficient solutions or one efficient solution for each nondominated outcome in Y. The former is analogous to finding all optimal solutions in a single objective optimization problem. This is different from the latter, because there are usually many feasible solutions with the same outcome. Hence the more common interpretation, and the one more aligned with the usual single objective practice of finding one optimal solution, is the latter. Moving from one efficient solution to another implies that at least one objective improves, whereas at least one deteriorates, i.e. there is a tradeoff between the different objectives.

There are several types of solutions due to the sense of minimization and the difference between the feasible region and its image under F. A feasible solution $\hat{x} \in X$ of a multiple objective optimization problem (26) is (*strictly/weakly*) *efficient* if there is no $x \in X$ such that $(F(x) \le F(\hat{x})/F(x) \le F(\hat{x}))$ $F(x)F(\hat{x})$. Here $F(x)F(\hat{x})$ indicates $F(x)F(\hat{x})$ but $F(x) \ne F(\hat{x})$. The outcome $\hat{y} = F(\hat{x})$ is called (*weakly*) nondominated. A strictly efficient solution $\hat{x}$ does not allow another x such that $F(x) = F(\hat{x})$. Moreover, an efficient solution $\hat{x}$ is called *properly efficient* if there is a real number $M > 0$ such that for all x and i with $f_i(x) < f_i(\hat{x})$ there is an index j such that $f_j(x) > f_j(\hat{x})$ and $(f_i(\hat{x}) - f_i(x))/(f_j(x) - f_j(\hat{x})) < M$, i.e. the tradeoffs are bounded.

Example 5.1.

We illustrate the concept of a multiple objective optimization problem with a small example of fluence map optimization. Assume that there are only two voxels, one (the top row) being a tumor voxel, which is to be treated with a dose of 60 Gy, and the other (the bottom row) being an organ at risk, which is to receive no radiation. Let the dose matrix be

$$A = \begin{pmatrix} 0.5 & 0.3 \\ 0.45 & 0.28 \end{pmatrix},$$

i.e. there are also only two bixels.

It is clear that it is possible to achieve the zero dose to the organ at risk with the zero fluence $x = 0$, and it is also clear that the target dose of 60 Gy to the tumor is possible with, for example, $x = (120, 0)^T$. In this case $Ax = (60, 54)^T$. The former solution fails to treat the target, and the latter delivers 54 Gy to the organ at risk. Applying the infinity norm model (12) with equal λ for PTV and OAR we obtain a solution of $x = (63.16, 0)^T$ with an under dose to the tumor and an over dose to the organ at risk of 28.42 Gy. But these are not the only options. In fact, with every increase of x_1 by one monitor unit, the dose delivered to the tumor voxel will get 0.5 closer to 60, but the dose to the organ at risk voxel will also increase by 0.45 and deviate away from the goal dose of 0. Hence, any fluence $x = (x_1, 0)^T$ with $0 \le x_1 \le 120$ is an efficient solution of a multiple objective optimization problem, where we simultaneously minimize the under dose to the tumor voxel (say z_1) and the over dose to the organ at risk voxel (say z_2). This multiple objective program is

$$\min \left\{ \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} : \begin{pmatrix} 0.50 & 0.30 \\ 0.45 & 0.28 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \ge \begin{pmatrix} 60 \\ 0 \end{pmatrix}; \right.$$
$$\left. - \begin{pmatrix} 0.50 & 0.30 \\ 0.45 & 0.28 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \ge \begin{pmatrix} -60 \\ 0 \end{pmatrix}; x, z \ge 0 \right\}.$$

In fact, the tradeoff between the two goals is constant and equals 10/9 of increase in z_1 for every unit of decrease in z_2.

The methods to solve multiobjective optimization problems depend on the type of problem, just as in the single objective case. We mention the most important facts and refer readers to [19] for a comprehensive introduction to multiobjective optimization. We first discuss solution methods for convex problems. The fundamental results for convex multiobjective optimization problems are summarized in the following theorem.

Theorem 5.1.
Assume that all objective functions and all constraints in (26) *are convex. Then the following hold.*

1. *A feasible solution $\hat{x}$ is weakly efficient if and only if there is a vector $\lambda \in R^p$, $\lambda 0$ such that $\hat{x}$ is an optimal solution of the single objective optimization problem*

$$\min \left\{ \sum_{k=1}^{p} \lambda_k f_k(x) : g(x) \leq 0 \right\}. \tag{27}$$

2. *A feasible solution $\hat{x}$ is properly efficient if and only if there is a vector $\lambda \in R^p$, $\lambda > 0$ such that $\hat{x}$ is an optimal solution of* (27).

3. *If, moreover, all objectives and constraints are linear, then a feasible solution $\hat{x}$ is efficient if and only if there is a vector $\lambda \in R^p$, $\lambda > 0$ such that $\hat{x}$ is an optimal solution of* (27).

It is important to notice the slight differences between the three statements. In fact, convex multiobjective optimization problems allow the characterization of weakly and properly efficient solutions via optimal solutions of (27) – distinguished by a nonnegative nonzero and positive λ, respectively – but not necessarily that of efficient ones. For linear problems, however, all efficient solutions are properly efficient (i.e. all trade-offs are bounded) and this difference disappears.

Theorem 5.1 is a powerful result when solving convex multiobjective optimization problems because it suggests we can solve (26) by repeatedly solving the single objective problem (27) for different λ vectors. However, deriving algorithms from the theorem is non-trivial. For linear multiobjective programming problems this leads to a variety of algorithms extending single objective simplex algorithms or algorithms that attempt to compute the set of nondominated outcomes.

Noting that it is always possible to scale λ so that $\sum_{k=1}^{p} \lambda_k = 1$, there is a temptation to interpret the coefficients λ_k as weights of importance of the objectives. However, the mathematical theory does not warrant that interpretation: The theorem does not help in constructing a λ vector that makes a particular solution efficient, i.e. determine the weights that ensure $\hat{x}$ is an optimal solution of (27). There can indeed be infinitely many (very) different λs that result in the same efficient solution. Furthermore, slightly different λs may result in very

different efficient solutions. In Example 5.1, noting that the problem is linear, applying the simplex algorithm to solve the weighted sum problem will, for any choice of $\lambda > 0$, always result in either $x = (120, 0)^T$ or $x = (0, 0)^T$ as the optimal solution. This is a property of the solution algorithm, which needs to be considered.

For nonconvex (such as discrete) optimization problems there exist efficient solutions that are not optimal solutions of (27) as the following observation shows. Consider a multiobjective optimization problem with three outcome vectors, i.e. $X = Y = \{(1, 5)^T, (4, 4)^T, (5, 1)^T\}$. Clearly minimizing $\lambda_1 x_1 + \lambda_2 x_2$, or equivalently $\mu x_1 + (1 - \mu) x_2$ (where $\mu = \lambda_1 / (\lambda_1 + \lambda_2)$), over X results in either $(1, 5)^T$ if $0.5 \leq \mu \leq 1$ or $(5, 1)^T$ if $0 \leq \mu \leq 0.5$, but never in $(4, 4)^T$. Nevertheless, $(4, 4)^T$ is efficient. From this we see that not every efficient solution is the result of a weighted sum of the individual objectives, if feasible set or objectives are not convex, i.e. the results of Theorem 5.1 are not guaranteed without convexity. Moreover, the interpretation of λ as importance weights has no meaning for nonconvex multiobjective optimization problems. Hence, to solve nonconvex multiple objective optimization problems, methods not relying on Theorem 5.1 are required.

Somewhat contrary to the lack of interpretation of λ, the idea of identifying efficient solutions by solving substitute, single objective optimization problems is prevalent. These substitute problems depend on the variables, objectives, and constraints of (26) and some additional parameters, which we collect in a vector μ. The resulting single objective problem is $\min s(f, g, x, \mu)$ and is solved for different values of μ. E.g. in problem (27), the parameter μ is the weigt vector λ and function s is the weighted sum of the components of f. This technique is called solving (26) by scalarization. Mathematically, scalarization is justified by showing that (a) for each choice of μ an optimal solution of $\min s(f, g, x, \mu)$ is a (weakly, strictly, properly) efficient solution of (26) and that, vice versa, for every (weakly, strictly, properly) efficient solution x of (26) there is a value of μ such that x is an optimal solution of $\min s(f, g, x, \mu)$. The precise mathematical properties of a scalarization method depend on the characteristics of the multiobjective problem as well as on the properties of s and μ. We refer to [19] and [23] for a summary of different scalarization techniques. The most popular of those is the ε-constraint method. The idea is to convert all but one of the objective functions of (26) into constraints $f_k(x) \leq \varepsilon_k$ for $k \neq j$ while maintaining minimization of the objective function f_j in the scalarized problem. This method can be applied to nonconvex and discrete problems, i.e. it can be shown that all efficient solutions are optimal solutions of some ε-constraint problem. Unfortunately, the resulting single objective problems are often difficult to solve.

Returning our discussion to Medical Physics, notice that the goal of radiation therapy is to treat the tumor while at the same time protecting organs at risk and normal tissue form the adverse effects of radiation. Thus radiotherapy pursues multiple and conflicting goals, making the multiobjective nature self evident. The earliest multiobjective optimization models for fluence map optimization are [15] and [31].

Consider (12). This is actually the weighted sum scalarization of Theorem 5.1 applied to the multiobjective optimization problem

$$\min \{ (\|D(x) - T\|_p, \|D(x) - T\|_p$$
$$\|D(x) - T\|_p, \|D(x) - T\|_p) : x \geq 0 \}. \tag{28}$$

We are assured that for every choice of the weights in (12) an efficient solution of (28) is obtained. This means that solving (12) yields a fluence map in which not all of the p-norm deviations from the prescripted dose for the target, organs at risk, and normal tissue can be improved simultaneously. As mentioned in Section 2, it is necessary to choose appropriate λ values to obtain a suitable fluence map. The multiobjective model, however, provides a wider perspective. We first note that only efficient solutions of (28) are meaningful candidates for a fluence map in clinical practice. The problem then becomes that of finding a suitable efficient solution. Theorem 5.1 assures that this can be found with appropriate weights λ. Yet the unknown relationship between these weights and efficient solutions often leads to a trial-and-error process of adjusting the weights in (12) and resolving the problem until a satisfactory solution is found. Here lies the pitfall of applying multiobjective optimization – just choosing a set of weights and computing the corresponding solutions is not sufficient. Observing that most fluence map optimization problems (without DVH constraints) are convex it is possible to exploit Theorem 5.1 and design algorithms to access the set of nondominated outcomes or efficient solutions of (28) in a meaningful way. This approach is pursued in [32].

For the cases of the 1- and infinity-norms, problem (28) becomes a multiobjective linear program. The research in [54] and [55] presents algorithms to approximate the nondominated outcomes of this MOLP to any prescribed accuracy for the infinity norm case. The authors of [53] present a method to find a finite set of efficient solutions that are guaranteed to cover the set of nondominated outcomes. In Figure 10 we show the nondominated set of our acoustic neuroma case using a similar model to (28) with $p = \infty$, where, however, the deviations for the brain stem and the eye are considered together as $\|D_O AR(x) - T_{OAR}\|_p$. Thus, this problem has three objectives, and the nondominated solutions can be displayed in a 3-dimensional figure. Note that the axes in this figure measure the maximal deviation of delivered dose from the prescripted dose for the PTV (α), the OARs (β), and the normal tissue (γ). Figure 11 shows a set of nondominated points from the set shown in Figure 10. Note that each point in Figure 11 defines a different treatment plan with its own characteristic tradeoffs between achieving the target doses for the PTV, the organs at risk, and the normal tissue.

To comment briefly on a discrete multiobjective optimization problem in Radiation Oncology, we refer to Example 3.1, where we have seen that minimizing beam-on time and minimizing set-up time are contradictory objectives in the optimal segmentation of fluence maps. Hence, the segmentation problem can be formulated as a biobjective
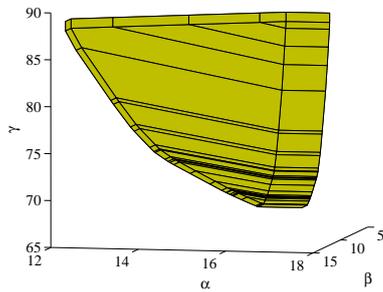
Figure 10. The nondominated set of a multiobjective fluence map optimization problem for the acoustic neuroma case (from [54]).
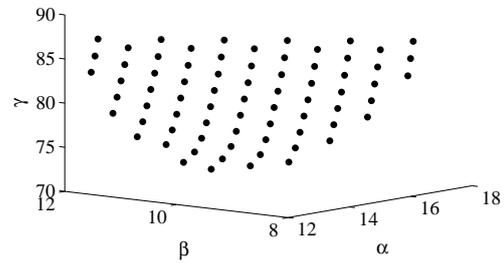

Figure 11. A representative set of nondominated set from Figure 10 (from [53]).

optimization problem with one objective being to minimize beam-on time and the other being to minimize set-up time. This fact has been used in [59] to solve the segmentation problem with the objective to minimize total treatment time, which is a weighted sum of the beam-on time and set-up time (number of apertures). The importance of this multiobjective optimization problem for clinical practice remains to be seen, however, because both [21] and [33] observed that there are only a few efficient solutions. It appears that navigating the tradeoffs is possible.

6. Conclusion

Optimization's supportive role in the field of Medical Physics is growing, and as we have discussed here, the spectrum of applications is wide. The essential optimization problems already applied to clinical problems will continue to adapt to increasingly sophisticated technology, and the complexity and size of the resulting optimization problems will grow. For this reason, the modeling and solution methods will need to be kept current to approach clinical demand. Also, new problems are, and will be, emerging that will require new models and possibly new solution methods. As the Medical Physics community addresses these challenges, the authors hope that this article demonstrates the value of an OR approach.

References

[1] R.K. Ahuja and H.W. Hamacher. A network flow algorithm to minimize beam-on-time for unconstrained multileaf collimator problems in cancer radiation therapy. *Networks*, 45(1):36–41, 2004.

[2] D. Aleman, J., Wallgren H. E. Romeijn, and J. Dempsey. A fluence map optimization model for restoring traditional fractionation in imrt treatment planning. *Optimization Letters*, 2013.

[3] D. Baatar, N. Boland, S. Brand, and P.J. Stuckey. Minimimum cardinality matrix decomposition into consecutive-ones matrices: CP and IP approaches. In P. Van Hentenrynck and L. Wolsey, editors, *Integration*

of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems – 4th International Conference, CPAIOR 2007, Brussels, Belgium, May 23-26, 2007, volume 4510 of Lecture Notes in Computer Science, pages 1–15. Springer-Verlag, Berlin, 2007.

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

[5] R. Bellman. Dynamic Programming. Princeton University Press, Princeton, NJ, 1957.

[6] D. Bertsimas, V. Cacchiani, D. Craft, and O. Nohadani. A hybrid approach to beam angle optimization in intensity-modulated radiation therapy. Computers & and Operations Research, 40(9):2187–2197, 2013.

[7] T.R. Bortfeld, A.L. Boyer, D.L. Kahler, and T.J. Waldron. X-ray field compensation with multileaf collimators. International Journal of Radiation Oncology, Biology, Physics, 28(3):723–730, 1994.

[8] S. Brand. The sum-of-increments constraint in the consecutive-ones matrix decomposition problem. In Proceedings of the 2009 ACM symposium on Applied Computing, pages 1417–1418. ACM, New York, NY, 2009.

[9] W. Cao, G. Lim, X. Li, Y. Li, R. Zhu, and X. Zhang. Incorporating deliverable monitor unit constraints into spot intensity optimization in impt treatment planning. Physics in Medicine and Biology, 58:5113–5125, 2013.

[10] D. Cheek, A. Holder, M. Fuss, and B. Salter. The relationship between the number of shots and the quality of gamma knife radiosurgeries. Optimization and Engineering, 6:541–555, 2006.

[11] C.W. Choi, W. Harvey, J.H.M. Lee, and P.J. Stuckey. Finite domain bounds consistency revisited. In A. Sattar and B.H. Kang, editors, AI 2006: Advances in Artificial Intelligence, volume 4304 of Lecture Notes in Computer Science, pages 49–58, 2006.

[12] M. Chu, Y. Zinchenko, S. Henderson, and M. Sharpe. Robust optimization for intensity modulated radiation therapy treatment planning under uncertainty. Physics in Medicine and Biology, 50:5463–5477, 2005.

[13] D. Conforti, F. Gueriero, R. Guido, and M. Veltri. An optimal decision-making approach for the management of radiotherapy patients. OR Spectrum, 33:123–148, 2011.

[14] W.W. Cooper, L.M. Seiford, and J. Zhu, editors. Handbook on Data Envelopment Analysis, volume 164 of International Series in Operations Research & Management Science. Springer Verlag, New York, NY, 2011.

[15] C. Cotrutz, M. Lahanas, C. Kappas, and D. Baltas. A multiobjective gradient-based dose optimization algorithm for external beam conformal radiotherapy. Physics in Medicine and Biology, 46:2161–2175, 2001.

[16] R. Dechter. Constraint Programming. Morgan Kaufmann, St. Louis, MO, 2003.

[17] G. Deng and M. Ferris. Neuro-dynamic programming for fractionated radiotherapy planning. In C.J.S. Alves, P.M. Pardalos, and L.N. Vicente, editors, Optimization in Medicine, volume 12 of Optimization and its Applications, pages 47–70. Springer Verlag, New York, NY, 2008.

[18] D. Djajaputra, Q. Wu, Y. Wu, and R. Mohan. Algorithm and performance of a clinical IMRT beam-angle optimization system. Physics in Medicine and Biology, 48:3191–3212, 2003.

[19] M. Ehrgott. *Multicriteria Optimization*. Springer Verlag, Berlin, 2nd edition, 2005.

[20] M. Ehrgott, Ç. Güler, H. W. Hamacher, and L. Shao. Mathematical optimization in intensity modulated radiation therapy. *4OR*, 6(3):199–262, 2008.

[21] M. Ehrgott, H.W. Hamacher, and M. Nußbaum. Decomposition of matrices and static multileaf collimators: A survey. In C.J.S Alves, P.M. Pardalos, and L.N. Vicente, editors, *Optimization in Medicine*, volume 12 of *Optimization and its Applications*, pages 27–48. Springer Verlag, Berlin, 2008.

[22] M. Ehrgott, A. Holder, and J. Reese. Beam selection in radiotherapy design. *Linear Algebra and its Applications*, 428(5-6):1272–1312, 2008.

[23] M. Ehrgott and M.M. Wiecek. Multiobjective programming. In J. Figueira, S. Greco, and M. Ehrgott, editors, *Multicriteria Decision Analysis: State of the Art Surveys*, volume 78 of *International Series in Operations Research & Management Science*, chapter 17, pages 667–722. Springer New York, 2005.

[24] A.T. Ernst, V.H. Mak, and L.R. Mason. An exact method for the minimum cardinality problem in the treatment planning of intensity-modulated radiotherapy. *INFORMS Journal on Computing*, 21(4):562–574, 2009.

[25] F. Glover and M. Laguna. *Tabu Search*. Springer, New York, NY, 1998.

[26] O. Haas, K. Burnham, and J. Mills. Optimization of beam orientation in radiotherapy using planar geometry. *Physics in Medicine and Biology*, 43:2179–2193, 1998.

[27] F. Hillier and G. Lieberman. *Introduction to Operations Research*. McGraw Hill, Columbus, OH, 9th edition, 2008.

[28] A. Holder and B. Salter. A tutorial on radiation oncology and optimization. In H.J. Greenberg, editor, *Tutorial on Emerging Methodologies and Applications in Operations Research*, volume 76 of *International Series in Operations Research and Management Science*, chapter 4, pages 4–1 – 4–45. Kluwer, New York, NY, 2004.

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

[30] T. Kapamara, K. Sheibani, O.C.L. Haas, C.R. Reeves, and D. Petrovic. A review of scheduling problems in radiotherapy. In K. J. Burnham and O. C. L. Haas, editors, *Proceedings of the Eighteenth International Conference on Systems Engineering (ICSE2006), Coventry University, UK*, pages 201–207, 2006.

[31] K.-H. Küfer and H.W. Hamacher. A multicriteria optimization approach for inverse radiotherapy planning. In W. Schlegel and T. Bortfeld, editors, *XIIIth International Conference on the Use of Computers in Radiation Therapy*, pages 26–28, Heidelberg, Germany, 2000. Springer Verlag, Berlin.

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

[33] M. Langer, V. Thai, and L. Papiez. Improved leaf sequencing reduces segments of monitor units needed to deliver IMRT using MLC. *Medical Physics*, 28:2450–58, 2001.

[34] J. Lim, M. Ferris, S. Wright, D. Shepard, and M. Earl. An optimization framework for conformal radiation treatment planning. *INFORMS Journal on Computing*, 19:366–380, 2007.

[35] K.M. Lin, J. Simpson, G. Sasso, A. Raith, and M. Ehrgott. Quality assessment for vmat prostate radiotherapy planning based on data envelopment analysis. *Phyics in Medicine and Biology*, 58(16), 2013.

[36] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, New York, NY, 1988.

[37] F. Newman and M. Assadi-Zeydabadi. An optimization model and solution for radiation shielding design of radiotherapy treatment vault. *Medical Physics*, 35:171–180, 2008.

[38] P. Nizin and R. Mooij. An approximation of centeral-axis absorbed dose in narrow photon beams. *Medical Physics*, 24(11):1775–1780, 1997.

[39] O. Nohadani, J. Seco, B. Martin, and T. Bortfeld. Dosimetry robustness with stochastic optimization. *Physics in Medicine and Biology*, 54:3421–3432, 2009.

[40] P.M. Pardalos and H.E. Romeijn, editors. *Handbook of Global Optimization Volume 2 (Nonconvex Optimization and Its Applications)*. Kluwer, Dordrecht, 2002.

[41] P. Pedregal. *Introduction to Optimization*. Springer-Verlag, New York, NY, 2004.

[42] S. Petrovic, W. Leung, X. Song, and S. Sundar. Algorithms for radiotherapy treatment booking. In *Proceedings of the 25th Workshop of the UK Planning and Scheduling Special Interest Group (PlanSIG'2006), Nottingham, UK*, 2006.

[43] M.H. Phillips and C. Holdsworth. When is better best? A multiobjective perspective. *Medical Physics*, 38:1635–1640, 2011.

[44] D. Pierre. *Optimization Theory with Applications*. Dover, New York, NY, 1986.

[45] J. Pintér. Global optimization. In A. Holder, editor, *Mathematical Programming Glossary*, http://glossary.computing.society.informs.org/second.php?page=GlobalOptimization.html, 1996-2013. INFORMS Computing Society.

[46] J. Pintér. *Computational Global Optimization in Nonlinear Systems: An Interactive Tutorial*. Lionheart Publishing, Atlanta, GA, 2001.

[47] J. Pintér. Global optimization: Software, test problems, and applications. In P.M. Pardalos and H.E. Romeijn, editors, *Handbook of Global Optimization*, volume 2, pages 515–569. Kluwer, 2002.

[48] J. Pluim, B. Maintz, and M. Viergever. Mutual information based registration of medical images: A survey. *IEEE Transactions on Medical Imaging*, 22:986–1004, 2003.

[49] C. Raphael. Mathematical modeling of objectives in radiation therapy treatment planning. *Physics in Medicine and Biology*, 37(6):1293–1311, 1992.

[50] 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.

[51] H.E. Romeijn, R.K. Ahuja, J.F. Dempsey, A. Kumar, and J.G. Li. A novel linear programming approach to

fluence map optimization for intensity modulated radiation therapy treatment planning. *Physics in Medicine and Biology*, 48:3521–3542, 2003.

[52] H.E. Romeijn and J.F. Dempsey. Intensity modulated radiation therapy treatment plan optimization. *Top*, 16:215–243, 2008.

[53] L. Shao and M. Ehrgott. Finding representative nondominated points in multiobjective linear programming. In P.P. Bonissone, editor, *IEEE Symposium on Computational Intelligence in Multicriteria Decision Making, Honolulu 1-5 April 2007, Proceedings*, pages 245–252. IEEE, 2007.

[54] L. Shao and M. Ehrgott. Approximately solving multiobjective linear programmes in objective space and an application in radiotherapy treatment planning. *Mathematical Methods of Operations Research*, 68(2):257–276, 2008.

[55] L. Shao and M. Ehrgott. Approximating the nondominated set of an MOLP by approximately solving its dual problem. *Mathematical Methods of Operations Research*, 68:469–492, 2008.

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

[57] Z.C. Taşkin, J.C. Smith, H.E. Romeijn, and J.F. Dempsey. Optimal multileaf collimator leaf sequencing in IMRT treatment planning. *Operations Research*, 58(3):674–690, 2010.

[58] J. Tervo, P. Kolmonen, T. Lyyra-Laitinen, J. Pintér, and T. Lahtinen. An optimization-based approach to the mutliple static delivery technique in radiation therapy. *Annals of Operations Research*, 119:205–227, 2003.

[59] G.M.G.H. Wake, N. Boland, and L.S. Jennings. Mixed integer programming approaches to exact minimization of total treatment time in cancer radiotherapy using multileaf collimators. *Computers & Operations Research*, 36:795–810, 2009.

[60] L.A. Wolsey. *Integer Programming*. John Wiley & Sons, New York, 1998.

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