Optimization Modelling and
Computational Issues in
Radiation Therapy

(lecture developed in collaboration with Peng Sun)

February 3, 2004
1 Outline

1. Radiation Therapy
2. Linear Optimization Models
3. Computation
4. Nonlinear and Mixed-Integer Models
5. Looking Ahead to the Course

2 Radiation Therapy

2.1 The Problem

2.2 Overview

• This year, 1,200,000 Americans will be diagnosed with cancer

• 600,000+ patients will receive radiation therapy
  – beam(s) of radiation delivered to the body in order to kill cancer cells

• Sadly, only 67% of “curable” patients will be cured

• High doses of radiation (energy/unit mass) can kill cells and/or prevent them from growing and dividing
  – true for cancer cells and normal cells

• Radiation is attractive because the repair mechanisms for cancer cells is less efficient than for normal cells

• Recent advances in radiation therapy now make it possible to:
  – map the cancerous region in greater detail
  – aim a larger number of different “beamlets” with greater specificity

• Spawned the new field of tomotherapy

2.2.1 Conventional Radiotherapy

Relative Intensity of Dose Delivered

- 3 to 7 beams of radiation
- Radiation oncologist and physicist work together to determine by manual “trial-and-error” process

With only a small number of beams, it is difficult/impossible to deliver required dose to tumor without impacting the critical area.
2.2.2 Recent Advances

- More accurate map of tumor area
  - CT — Computed Tomography
  - MRI — Magnetic Resonance Imaging

- More accurate delivery of radiation
  - IMRT: Intensity Modulated Radiation Therapy
  - Tomotherapy

2.2.3 Formal Problem Statement

- For a given tumor and given critical areas
- For a given set of possible beamlet origins and angles
- Determine the weight on each beamlet such that:
  - dosage over the tumor area will be at least a target level $\gamma_L$
  - dosage over the critical area will be at most a target level $\gamma_U$
3 Linear Optimization Models

3.1 Discretize the Space

Divide up region into a 2-dimensional (or 3-dimensional) grid of pixels.

![Pixel Grid](image)

3.2 Create Beamlet Data

Create the beamlet data for each of \( p = 1, \ldots, n \) possible beamlets. 

\( D^p \) is the matrix of unit doses delivered by beam \( p \).

\[
\begin{array}{cccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

\( D^p_{i, j} \) = unit dose delivered to pixel \((i, j)\) by beamlet \( p \).

3.3 Dosage Equations

Decision variables \( w = (w_1, \ldots, w_n) \)

\( w_p \) = intensity weight assigned to beamlet \( p \), \( p = 1, \ldots, n \).

\[
D_{i, j} := \sum_{p=1}^{n} D^p_{i, j} w_p
\]

("\( := \)" denotes "by definition")

\[
D := \sum_{p=1}^{n} D^p w_p
\]

is the matrix of the integral dose (total delivered dose)
3.4 Definitions of Regions

\[ \mathcal{T} \text{ is the target area} \]
\[ \mathcal{C} \text{ is the critical area} \]
\[ \mathcal{N} \text{ is normal tissue} \]
\[ S := \mathcal{T} \cup \mathcal{C} \cup \mathcal{N} \]

3.5 Ideal Linear Model

\[
\begin{align*}
\text{minimize} & \quad \sum_{(i,j) \in S} D_{i,j} \\
\text{s.t.} & \quad D_{i,j} = \sum_{p=1}^{n} D_{i,j}^p w_p \quad (i, j) \in S \\
& \quad w \geq 0 \\
& \quad D_{i,j} \geq \gamma_L \quad (i, j) \in \mathcal{T} \\
& \quad D_{i,j} \leq \gamma_U \quad (i, j) \in \mathcal{C}
\end{align*}
\]

\[
\begin{align*}
\text{minimize} & \quad \sum_{(i,j) \in S} D_{i,j} \\
\text{s.t.} & \quad D_{i,j} = \sum_{p=1}^{n} D_{i,j}^p w_p \quad (i, j) \in S \\
& \quad w \geq 0 \\
& \quad D_{i,j} \geq \gamma_L \quad (i, j) \in \mathcal{T} \\
& \quad D_{i,j} \leq \gamma_U \quad (i, j) \in \mathcal{C}
\end{align*}
\]

- Unfortunately, this model is typically infeasible.
- Cannot deliver dose to tumor without some harm to critical area(s).
3.6 Engineered Approaches

minimize \( w, D \) \( \theta_T \sum_{i,j \in T} D_{i,j} + \theta_C \sum_{(i,j) \in C} D_{i,j} + \theta_N \sum_{(i,j) \in N} D_{i,j} \)

s.t. \( D_{i,j} = \sum_{p=1}^{n} D_{i,j}^p w_p \quad (i, j) \in \mathcal{S} \)

\[ w \geq 0 \]

\[ D_{i,j} \geq \gamma_{i,j} \quad (i, j) \in \mathcal{T} \]

\[ w_m \leq 0.05 \sum_{p=1}^{n} w_p \quad m = 1, \ldots, n \]

Some other possible objective functions:

Let \((\text{Target}_{i,j})\) be the target prescribed dose to be delivered to pixel \((i, j)\)

minimize \( w, D \) \( \max_{(i,j) \in \mathcal{S}} |D_{i,j} - (\text{Target}_{i,j})| \)

s.t. \( D_{i,j} = \sum_{p=1}^{n} D_{i,j}^p w_p \quad (i, j) \in \mathcal{S} \)

\[ w \geq 0 \]

This is the same as:

minimize \( w, D, \mu \) \( \mu \)

s.t. \( -\mu \leq D_{i,j} - (\text{Target}_{i,j}) \leq \mu \quad (i, j) \in \mathcal{S} \)

\[ D_{i,j} = \sum_{p=1}^{n} D_{i,j}^p w_p \quad (i, j) \in \mathcal{S} \]

\[ w \geq 0 \]

Here is another model:

minimize \( w, D \) \( \sum_{(i,j) \in \mathcal{S}} |D_{i,j} - (\text{Target}_{i,j})| \)

s.t. \( D_{i,j} = \sum_{p=1}^{n} D_{i,j}^p w_p \quad (i, j) \in \mathcal{S} \)

\[ w \geq 0 \]
\[
\text{minimize } \sum_{(i,j) \in S} \Delta_{i,j}
\]

This is the same as:

s.t. \[D_{i,j} = \sum_{p=1}^{n} D_{ij}^p w_p \quad (i, j) \in S\]
\[w \geq 0\]
\[\Delta_{i,j} \leq D_{i,j} - \text{(Target)}_{i,j} \leq \Delta_{i,j} \quad (i, j) \in S\]

4 Computation

4.1 Base Case Model

Consider the “base case” example problem:

\[
\text{minimize } 1 \cdot \sum_{(i,j) \in N} \Delta_{i,j} + 100 \sum_{(i,j) \in C} \Delta_{i,j} + 30 \sum_{(i,j) \in T} \Delta_{i,j}
\]

s.t. \[D_{i,j} = \sum_{p=1}^{n} D_{ij}^p w_p \quad (i, j) \in S\]
\[w \geq 0\]
\[\Delta_{i,j} \leq D_{i,j} - \text{(Target)}_{i,j} \leq \Delta_{i,j} \quad (i, j) \in S\]

4.2 Size of the Model

4.2.1 Dimensional Analysis

\[
\text{minimize } 1 \cdot \sum_{(i,j) \in N} \Delta_{i,j} + 100 \sum_{(i,j) \in C} \Delta_{i,j} + 30 \sum_{(i,j) \in T} \Delta_{i,j}
\]

s.t. \[D_{i,j} = \sum_{p=1}^{n} D_{ij}^p w_p \quad (i, j) \in S\]
\[w \geq 0\]
\[\Delta_{i,j} \leq D_{i,j} - \text{(Target)}_{i,j} \leq \Delta_{i,j} \quad (i, j) \in S\]

Dimensional Analysis:

- number of pixels = 31,397 (≈ π * 100²)
- number of beamlets = 564 \((n)\)
- \(|T| = 3,859; \quad |C| = 630; \quad |N| = 26,908\)
- \(|S| = 31,397\)
\[
\begin{align*}
\text{minimize} \quad & \sum_{(i,j) \in E} \Delta_{ij} + 100 \sum_{(i,j) \in E} \Delta_{ij} + 30 \sum_{(i,j) \in T} \Delta_{ij} \\
\text{s.t.} \quad & D_{ij} = \sum_{p=1}^{n} D_{ij}^p w_p \quad (i,j) \in S \\
& w \geq 0 \\
& -\Delta_{ij} \leq D_{ij} - (\text{Target})_{ij} \leq \Delta_{ij} \quad (i,j) \in S
\end{align*}
\]

\[
\begin{array}{|c|c|}
\hline
\text{Decision Variables} & \text{Number} \\
\hline
D_{ij} & 31,397 \\
w & 664 \\
\Delta_{ij} & 31,397 \\
\text{Total} & 63,398 \\
\hline
\end{array}
\]

\[
\begin{align*}
\text{minimize} \quad & \sum_{(i,j) \in E} \Delta_{ij} + 100 \sum_{(i,j) \in E} \Delta_{ij} + 30 \sum_{(i,j) \in T} \Delta_{ij} \\
\text{s.t.} \quad & D_{ij} = \sum_{p=1}^{n} D_{ij}^p w_p \quad (i,j) \in S \\
& w \geq 0 \\
& -\Delta_{ij} \leq D_{ij} - (\text{Target})_{ij} \leq \Delta_{ij} \quad (i,j) \in S
\end{align*}
\]

### 4.2.2 Number of Constraints

<table>
<thead>
<tr>
<th>Simple Variable Upper/Lower Bounds</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>w ≥ 0</td>
<td>564</td>
</tr>
<tr>
<td>Total</td>
<td>564</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Other Constraints*</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>D_{ij} =</td>
<td>31,397</td>
</tr>
<tr>
<td>≤ D_{ij} - (Target)_{ij} ≤</td>
<td>62,591</td>
</tr>
<tr>
<td>Total</td>
<td>94,191</td>
</tr>
</tbody>
</table>

*We usually exclude simple variable upper/lower bounds when counting constraints.

### 4.2.3 Summary

<table>
<thead>
<tr>
<th>Variables</th>
<th>Constraints*</th>
</tr>
</thead>
<tbody>
<tr>
<td>63,398</td>
<td>94,191</td>
</tr>
</tbody>
</table>

*Excludes variable upper/lower bounds.
4.3 Base Case Model

4.3.1 Optimal Solution

Base Case Model Solution

4.4 Another Model Solution

Solution of a nonlinear model.

4.5 Dose Histogram

4.5.1 of Solution

4.6 Another Model Solution

Solution of a nonlinear model, where $\theta_N = \theta_C = \theta_T = 1$. 
5 Computation

5.1 Computational Issues

5.1.1 Software/Algorithms

- Software codes:
  - CPLEX simplex (pivoting method)
  - CPLEX barrier
  - LOQO

- Algorithms:
  - Simplex method ("pivoting" method)
  - Interior-point method (IPM) ("barrier" method)

5.1.2 Counting Iterations

- Iteration Counts:
  - Number of pivots for simplex method
  - Number of Newton steps for IPM

5.1.3 Issues in Running Times

- Running time will be affected by:
  - number of constraints
  - number of variables
  - software code
  - type of algorithm (simplex or IPM)
  - properties of linear algebra systems involved
    - density/patterns of nonzeros of matrix systems to be solved
  - other problem characteristics specific to problem
  - idiosyncratic influences
  - pre-processing heuristics

5.2 Base Case

5.2.1 No Pre-Processing

- Base Case Model
- No Pre-Processing
<table>
<thead>
<tr>
<th>Code</th>
<th>Algorithm</th>
<th>Iterations</th>
<th>Running Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Simplex</td>
<td>183,530</td>
<td>440</td>
</tr>
<tr>
<td>CPLEX</td>
<td>Barrier</td>
<td>49</td>
<td>13</td>
</tr>
</tbody>
</table>

5.3 Some Generic Rules

1. The simplex algorithm is designed to handle variables with lower bounds and upper bounds:

\[
\begin{align*}
\min_x & \quad c^T x \\
\text{subject to} & \quad Ax = b \\
& \quad \ell \leq x \leq u
\end{align*}
\]

where \( \ell_j = -\infty \) and/or \( u_j = +\infty \) is allowed.

2. We say \( x_j \) has no bounds if \( \ell_j = -\infty \) and \( u_j = +\infty \). Otherwise \( x_j \) is a bounded variable.

3. For the simplex method, the work per pivot generally depends on the number of nonzeros in \( A \).

4. If \( A \) is very sparse (its density of nonzero elements is low), then the work per pivot will be low.

5. The number of simplex pivots in a “good” model is roughly between \( m \) and \( 10n \).

\[
\begin{align*}
\min_x & \quad c^T x \\
\text{subject to} & \quad Ax = b \\
& \quad \ell \leq x \leq u
\end{align*}
\]

6. The work per iteration of an interior-point method generally depends on the structure of the matrix \( K = \begin{pmatrix} I & A^T \\ A & 0 \end{pmatrix} \).

\[
K = \begin{pmatrix} I & A^T \\ A & 0 \end{pmatrix}.
\]

6. The structure of \( K \) is often (but not always) related to the structure of the matrix \( AA^T \) because the following two matrices are “similar”:

\[
K = \begin{pmatrix} I & A^T \\ A & 0 \end{pmatrix}, \quad P = \begin{pmatrix} I & A^T \\ 0 & -AA^T \end{pmatrix}.
\]

7. The number of interior-point method iterations is typically 25–80 (independent of \( m \) and/or \( n \)).
5.4 Pre-Processing

5.4.1 Heuristics

Pre-Processing Heuristics in Commercial-Grade Software

- Designed to Eliminate Constraints and/or Variables

- Example:

\[-5x + 3y + z = 17\]

\[0 \leq x \leq 4 \quad 0 \leq y \leq 2 \quad 10 \leq z \leq 40\]

- Example:

\[-5x + 3y + z = 17\]

\[0 \leq x \leq 4 \quad 0 \leq y \leq 2 \quad 10 \leq z \leq 40\]

- \(z = 17 + 5x - 3y \geq 17 + 5(0) - 3(2) = 11 \geq 10\)

- \(z = 17 + 5x - 3y \leq 17 + 5(4) - 3(0) = 37 \leq 40\)

- Therefore we can eliminate the bounds on \(z\)

- Therefore we can treat \(z\) as a free variable

- Therefore we can eliminate \(z\) from our model altogether.

- Base Case Model

- With Pre-Processing

<table>
<thead>
<tr>
<th>Code</th>
<th>Algorithm</th>
<th>Iterations</th>
<th>Running Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPLEX</td>
<td>Simplex</td>
<td>18,428</td>
<td>4,3 4</td>
</tr>
<tr>
<td>CPLEX</td>
<td>Barrier</td>
<td>16</td>
<td>130 133</td>
</tr>
</tbody>
</table>

5.5 Equivalent Formulation

5.5.1 “Small” Model

“Small” Model:

Equivalent Formulation: (eliminate \(D_{ij}\))

\[
\begin{align*}
\text{minimize} & \quad w \cdot \Delta \\
\text{s.t.} & \quad -\Delta_{ij} \leq \sum_{p=1}^{n} D_{ij}^{p} w_{p} - \text{(Target)}_{ij} \leq \Delta_{ij} \\
& \quad (i, j) \in S \\
& \quad w \geq 0
\end{align*}
\]
<table>
<thead>
<tr>
<th></th>
<th>Base Case Model</th>
<th>Small Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variables</td>
<td>63,358</td>
<td>31,961</td>
</tr>
<tr>
<td>Constraints*</td>
<td>94,191</td>
<td>62,794</td>
</tr>
</tbody>
</table>

*always excludes simple variable upper/lower bounds

- Small Model

<table>
<thead>
<tr>
<th>Code</th>
<th>Algorithm</th>
<th>Iterations</th>
<th>CPU (sec)</th>
<th>Wall (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPLEX</td>
<td>Simplex</td>
<td>171,656</td>
<td>300</td>
<td>216</td>
</tr>
<tr>
<td>CPLEX</td>
<td>Barrier</td>
<td>57</td>
<td>80</td>
<td>31</td>
</tr>
</tbody>
</table>

5.6 Comparisons

<table>
<thead>
<tr>
<th>Code</th>
<th>Algorithm</th>
<th>Model</th>
<th>Running Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Base Case</td>
<td>250</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pre-Processed</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Small Model</td>
<td>216</td>
</tr>
<tr>
<td>CPLEX</td>
<td>Simplex</td>
<td></td>
<td>37</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Base Case</td>
<td>133</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pre-Processed</td>
<td>31</td>
</tr>
<tr>
<td>CPLEX</td>
<td>Barrier</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Small Model</td>
<td></td>
</tr>
</tbody>
</table>

6 Nonlinear Optimization

6.1 Quadratic Model

\[
\begin{align*}
\text{QP:} \quad & \text{minimize} & & 1 \cdot \sum_{(i,j) \in \mathcal{N}} [D_{i,j} - \text{Target}_{i,j}]^2 \\
& & & + 100 \cdot \sum_{(i,j) \in \mathcal{C}} [D_{i,j} - \text{Target}_{i,j}]^2 \\
& & & + 30 \cdot \sum_{(i,j) \in \mathcal{T}} [D_{i,j} - \text{Target}_{i,j}]^2 \\
& \text{s.t.} & & D_{i,j} = \sum_{p=1}^{n} D_{i,j}^p w_p \quad (i, j) \in \mathcal{S} \\
& & & w \geq 0
\end{align*}
\]
6.1.1 Quadratic Model Output

6.2 Quadratic Model

6.2.1 Computational Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Code</th>
<th>Algorithm</th>
<th>Iterations</th>
<th>Running Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base Case QP Model</td>
<td>LOQO</td>
<td>Barrier</td>
<td>31</td>
<td>82.7</td>
</tr>
<tr>
<td>Small QP Model</td>
<td>LOQO</td>
<td>Barrier</td>
<td>32</td>
<td>149.0</td>
</tr>
</tbody>
</table>

7 Mixed Integer Optimization

7.1 Limiting the Number of Beamlets

\[
\begin{align*}
\text{minimize} & \quad 1 \cdot \sum_{(i,j) \in \mathcal{N}} \Delta_{ij} + 100 \sum_{(i,j) \in \mathcal{C}} \Delta_{ij} + 30 \sum_{(i,j) \in \mathcal{T}} \Delta_{ij} \\
\text{s.t.} & \quad D_{i,j} = \sum_{p=1}^{n} D_{i,j}^p w_p \quad (i,j) \in \mathcal{S} \\
& \quad w \geq 0 \\
& \quad -\Delta_{ij} \leq D_{i,j} - (\text{Target})_{i,j} \leq \Delta_{ij} \quad (i,j) \in \mathcal{S} \\
& \quad w_p \leq 100y_p \quad p = 1, \ldots, n \\
& \quad y_p \in \{0, 1\} \quad p = 1, \ldots, n \\
& \quad \sum_{p=1}^{n} y_p \leq 15. 
\end{align*}
\]
7.2 Computation

7.2.1 CPLEX MIP Solver

<table>
<thead>
<tr>
<th>MIP Gap (%)</th>
<th>Simplex Pivots</th>
<th>Running Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>11,666</td>
<td>7 seconds</td>
</tr>
<tr>
<td>15</td>
<td>11,686</td>
<td>7 seconds</td>
</tr>
<tr>
<td>12</td>
<td>11,696</td>
<td>5 minutes</td>
</tr>
<tr>
<td>10</td>
<td>14,538</td>
<td>9 minutes</td>
</tr>
<tr>
<td>7</td>
<td>14,538</td>
<td>7 minutes</td>
</tr>
<tr>
<td>5</td>
<td>14,538</td>
<td>10 minutes</td>
</tr>
<tr>
<td>4</td>
<td>14,538</td>
<td>7 minutes</td>
</tr>
<tr>
<td>3</td>
<td>14,538</td>
<td>5 minutes</td>
</tr>
<tr>
<td>2</td>
<td>3,655,445</td>
<td>17 hours</td>
</tr>
</tbody>
</table>

8 Modifications of the Model

8.1 Partial Volume Constraints

Partial Volume Constraints:

“No more than 20% of the critical region can exceed a
dose of $30G_y$.”

“No more than 5% of the critical region can exceed a
dose of $50G_y$.”

Approach #1 (Integer Programming Model)

Let $M$ be a very large number,

\[
\begin{align*}
D_{i,j} &\leq 30 + M \cdot y_{i,j}, & y_{i,j} \in \{0, 1\}, & (i,j) \in \mathcal{C} \\
D_{i,j} &\leq 50 + M \cdot z_{i,j}, & z_{i,j} \in \{0, 1\}, & (i,j) \in \mathcal{C} \\
\sum_{(i,j) \in \mathcal{C}} y_{i,j} &\leq |\mathcal{C}| \times 0.20 \\
\sum_{(i,j) \in \mathcal{C}} z_{i,j} &\leq |\mathcal{C}| \times 0.05
\end{align*}
\]

Approach #2 (Error Function Approach)

The error function, or sigmoid function, is of the form:

\[
\text{err} f(x) = \frac{1}{1 + e^{-ax}}
\]

\[
\begin{align*}
\text{err} f(x) &= \frac{1}{2} \quad \text{at} \quad x = 0 \\
\text{err} f(x) &\to 1 \quad \text{as} \quad x \to \infty \\
\text{err} f(x) &\to 0 \quad \text{as} \quad x \to -\infty
\end{align*}
\]

Instead of integer variables, we use
\[
\sum_{(i,j) \in C} \text{err}(D_{ij} - 30) \leq |C| \times 0.20
\]
\[
\sum_{(i,j) \in C} \text{err}(D_{ij} - 50) \leq |C| \times 0.05
\]

9 Looking Ahead

9.1 Modeling Languages

9.1.1 Used in the Course

- Modeling languages and software used in the course
  - OPL Studio
    - linear and mixed-integer programming
    - solver is CPLEX simplex and/or CPLEX barrier
    - first half of course
  - AMPL
    - linear and nonlinear programming
    - solver is LOQO
    - second half of course

9.2 Modeling Tools

9.2.1 and Issues

- “Column Generation” (week 3)
  - generates new decision variables “on the fly”
- Exact optimization and exact feasibility
  - in models
  - in algorithms

- Computational Issues in LP (next lecture)
  - simplex method with upper/lower bounds
  - methods for updating the basis inverse