# Optimization Modeling and Computational Issues in Radiation Therapy

(lecture developed in collaboration with Peng Sun)

February 5, 2002

# Outline

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

- 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

# Radiation Therapy

## Overview

- 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

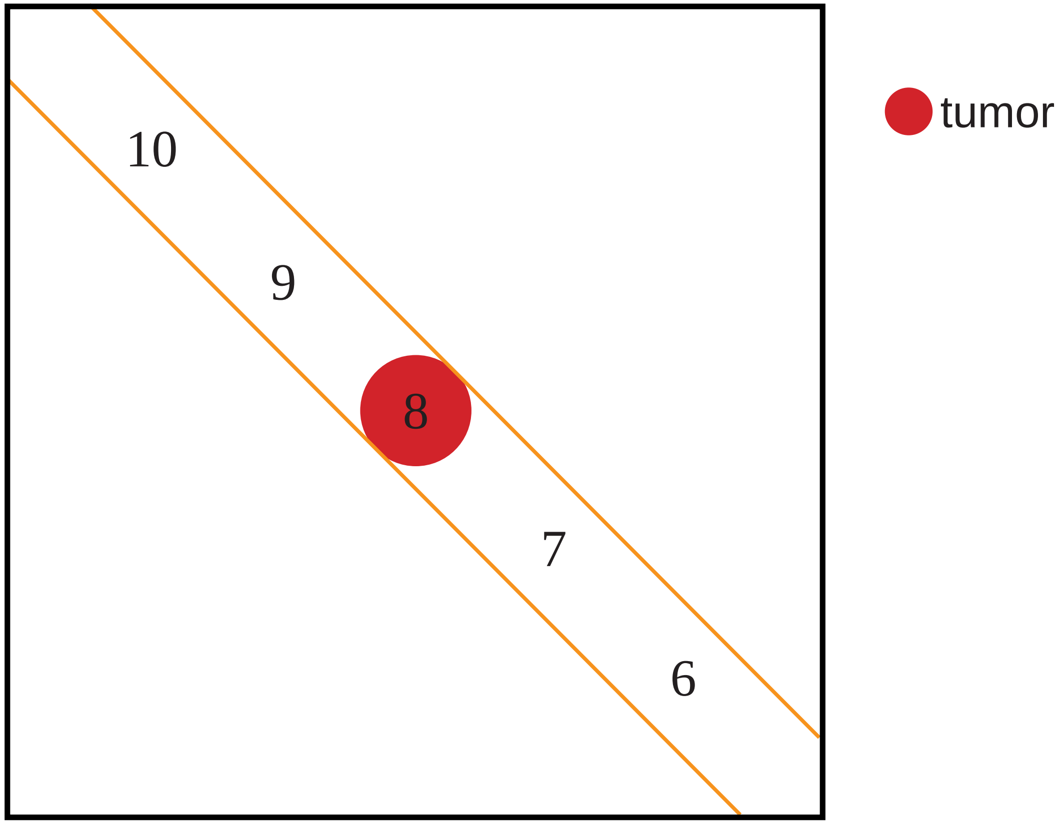
 

- Recent advances in radiation therapy now make it possible to:
  - map the cancerous region in greater detail
  - aim a larger number of different “beamlets” with greater specificity
- Spawned the new field of *tomotherapy*
- “Optimizing the Delivery of Radiation Therapy to Cancer Patients,” by Shepard, Ferris, Olivera, and Mackie, *SIAM Review*, Vol. 41, pp. 721–744, 1999.

# Radiation Therapy

## Overview

### Conventional Radiotherapy...

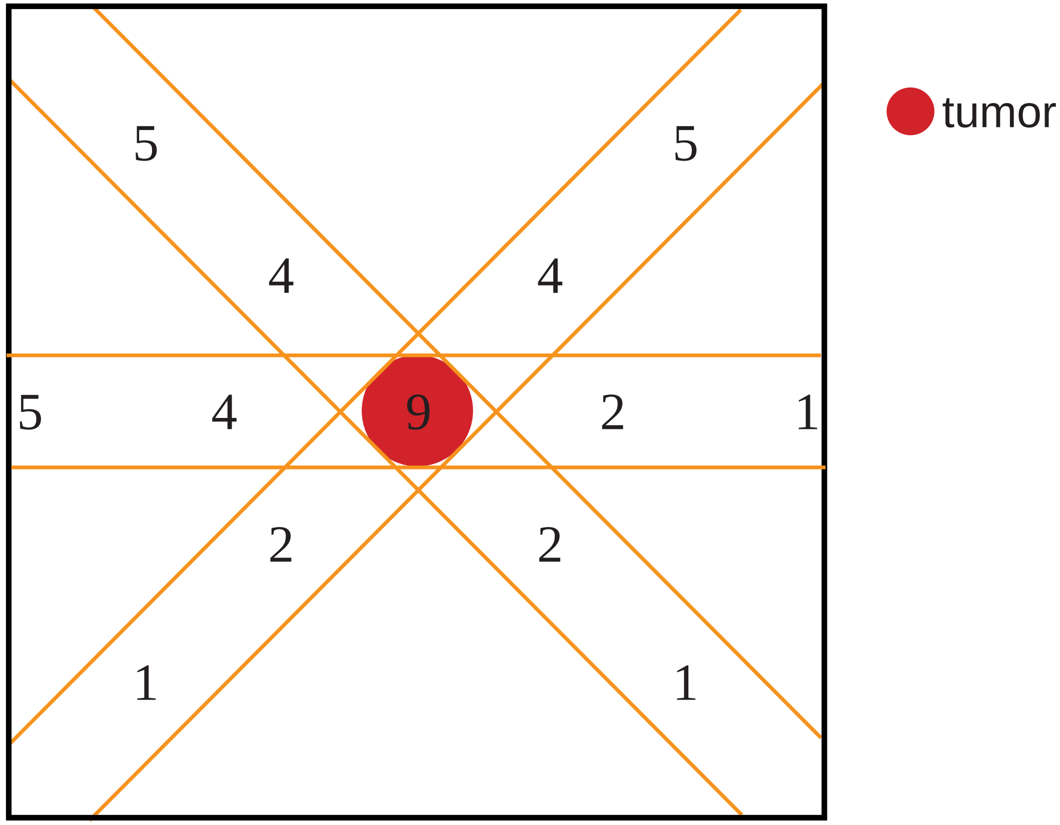



Relative Intensity of Dose Delivered

# Radiation Therapy

## Overview

### ...Conventional Radiotherapy...




Relative Intensity of Dose Delivered

# Radiation Therapy

## Overview

### ...Conventional Radiotherapy...

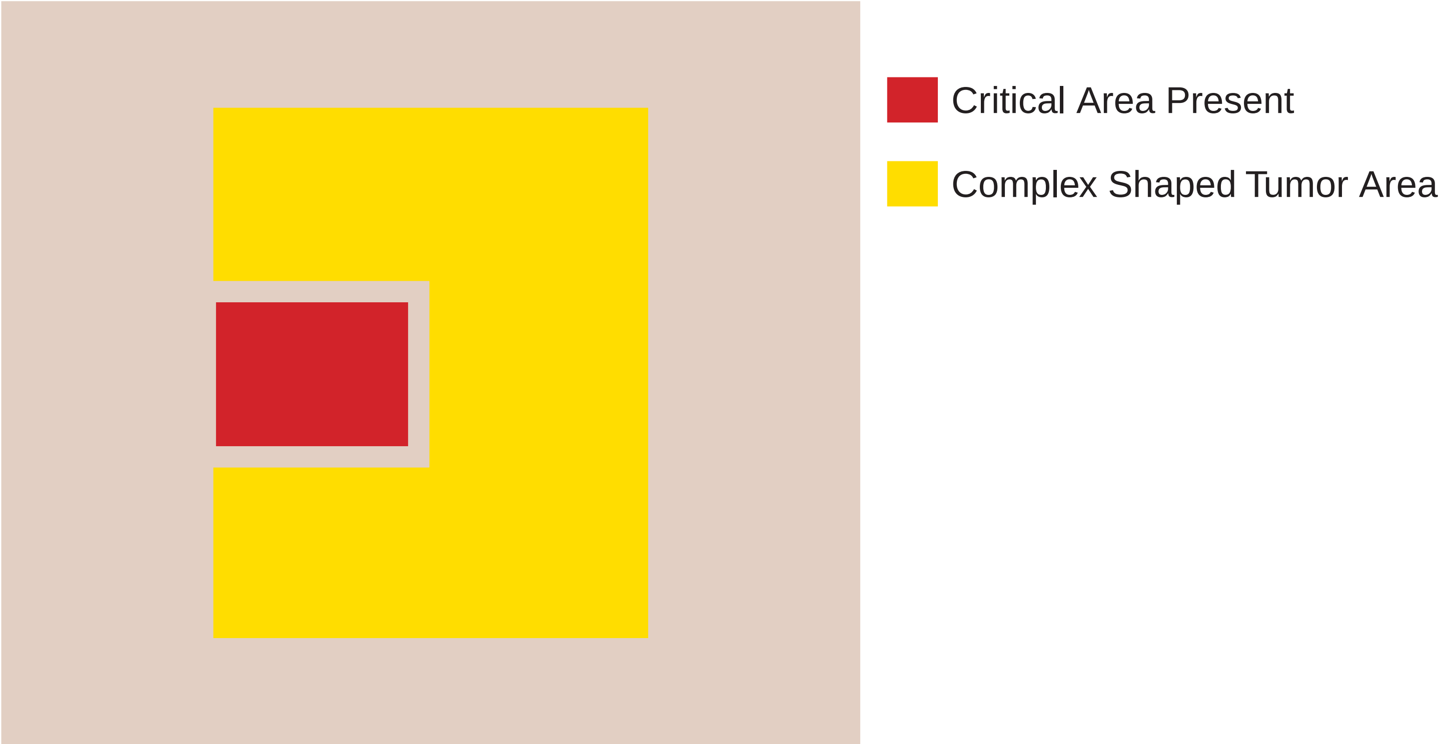
In conventional radiotherapy

- 3 to 7 beams of radiation
- radiation oncologist and physicist work together to determine a set of beam angles and beam intensities
- determined by manual "trial-and-error" process

# Radiation Therapy

## Overview

### ...Conventional Radiotherapy



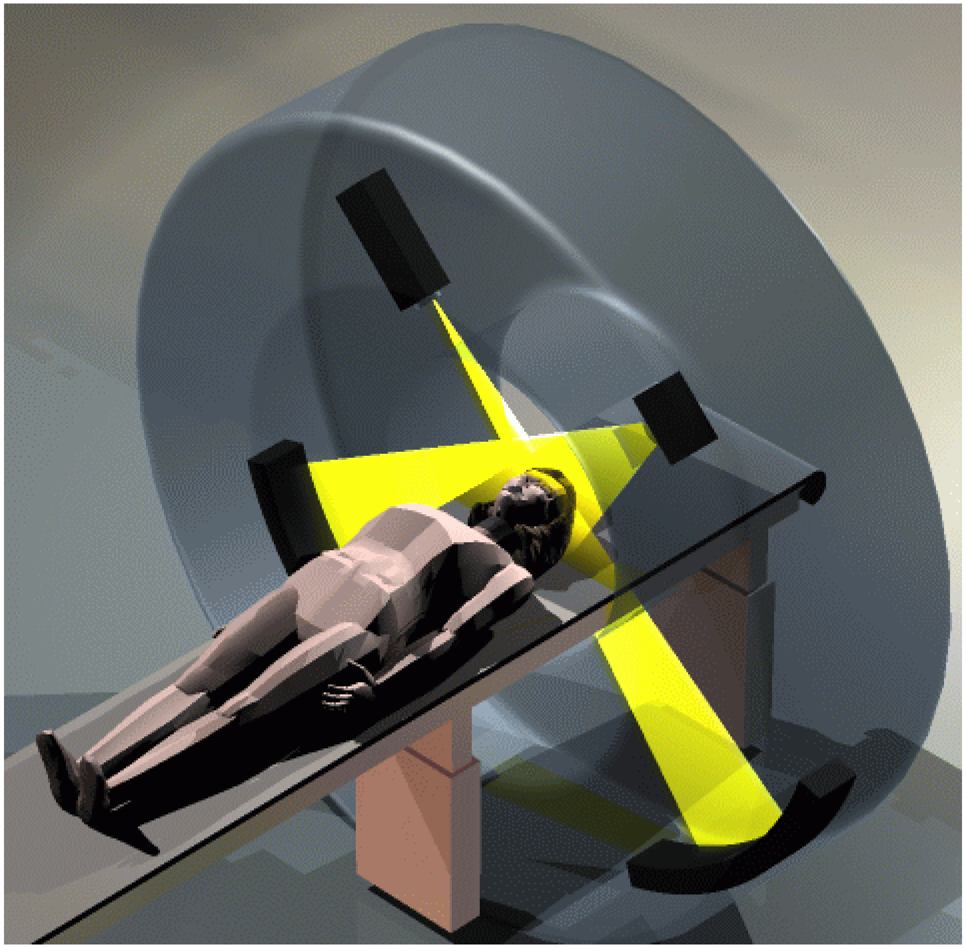

With only a small number of beams, it is difficult/impossible to deliver required dose to tumor without impacting the critical area.

- More accurate map of tumor area
  - CT — Computed Tomography
  - MRI — Magnetic Resonance Imaging
- More accurate delivery of radiation
  - IMRT: Intensity Modulated Radiation Therapy
  - Tomotherapy

# Radiation Therapy

## Overview

### ...Recent Advances



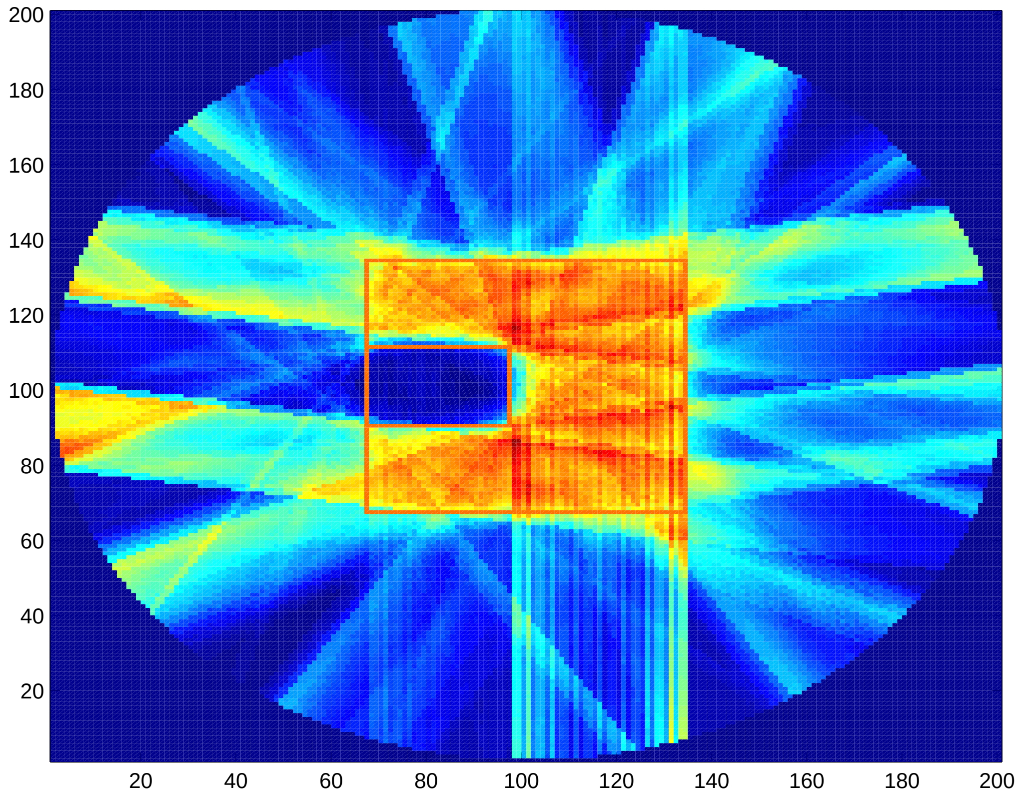
- For a given tumor and given critical areas
- For a given set of possible beamlet origins and angles
- Determine the weight on each beamlet such that:
  - dosage over the tumor area will be at least a target level $\gamma_L$
  - dosage over the critical area will be at most a target level $\gamma_U$

# Radiation Therapy

## Overview

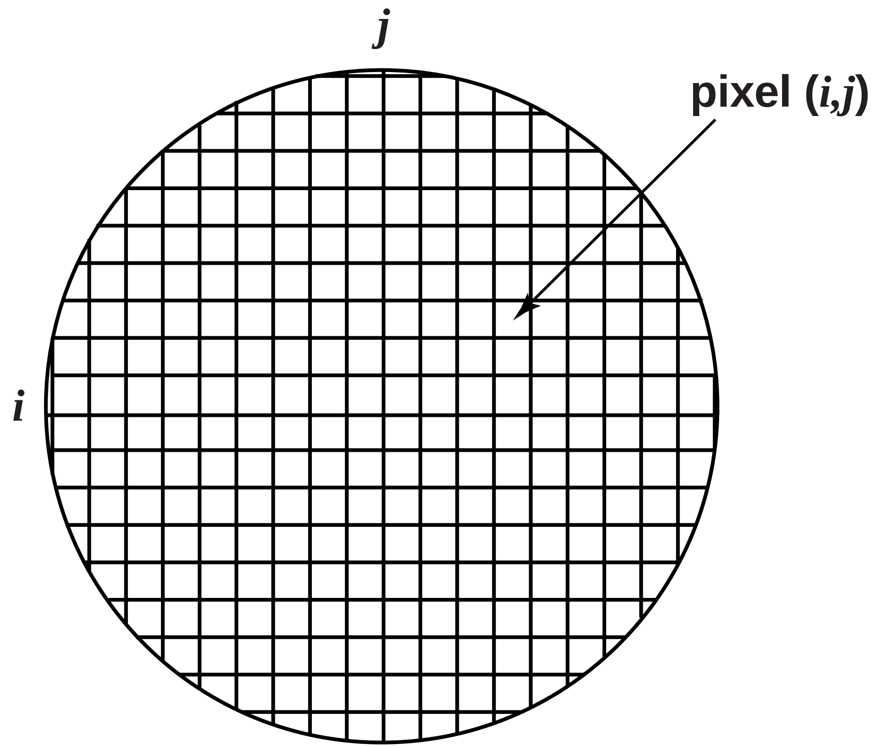
## ...Formal Problem Statement



# Discretize the Space

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



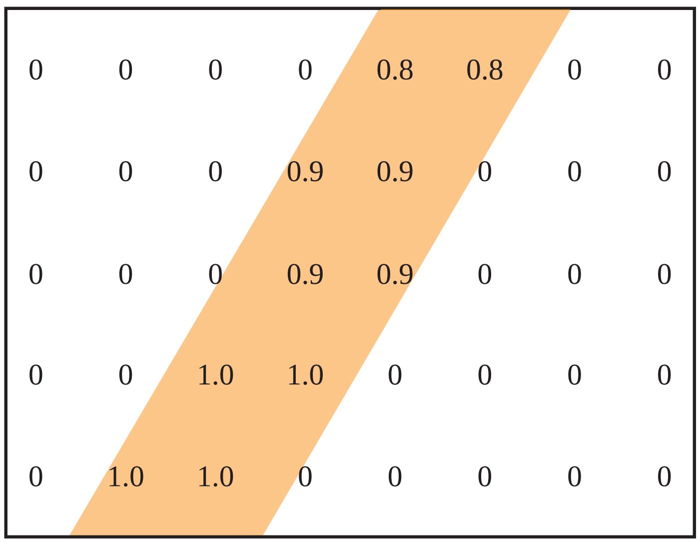

 

# Linear Optimization Models

## 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 $p$ .




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

# Dosage Equations

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

$w_p =$ intensity weight assigned to beamlet $p$,

$p = 1, \ldots, n.$

$$D_{i\,j} := \sum_{p=1}^{n} D_{i\,j}^{p}\, w_p$$

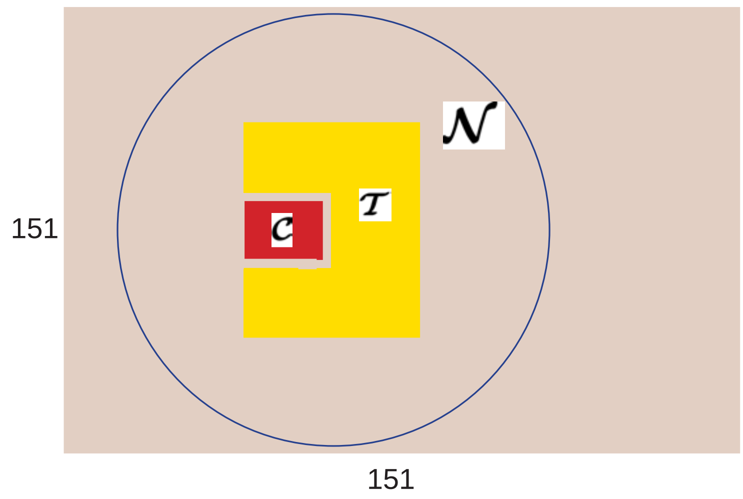
("$:=$" denotes "by definition")

$$D := \sum_{p=1}^{n} D^{p}\, w_p$$

is the matrix of the integral dose (total delivered dose)

# Definitions of Regions




$\mathcal{T}$ is the target area

$\mathcal{C}$ is the critical area

$\mathcal{N}$ is normal tissue

$$\mathcal{S} := \mathcal{T} \cup \mathcal{C} \cup \mathcal{N}$$

# Ideal Linear Model

$$\begin{aligned}
\underset{w,D}{\textbf{minimize}} \quad & \sum_{(i,j)\in\mathcal{S}} D_{i\,j} \\
\textbf{s.t.} \quad & D_{i\,j} = \sum_{p=1}^{n} D^{p}_{i\,j}\, w_p & (i,j)\in\mathcal{S} \\
& w \geq 0 \\
& D_{i\,j} \geq \gamma_L & (i,j)\in\mathcal{T} \\
& D_{i\,j} \leq \gamma_U & (i,j)\in\mathcal{C}
\end{aligned}$$

# Ideal Linear Model

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

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

# Linear Optimization Models

## Engineered Approaches

$$\underset{w,D}{\text{minimize}} \quad \theta_{\mathcal{T}} \sum_{(i,j)\in\mathcal{T}} D_{i\,j} + \theta_{\mathcal{C}} \sum_{(i,j)\in\mathcal{C}} D_{i\,j} + \theta_{\mathcal{N}} \sum_{(i,j)\in\mathcal{N}} D_{i\,j}$$

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

$$w \geq 0$$

$$\gamma_{i\,j}^{L} \leq D_{i\,j} \leq \gamma_{i\,j}^{U} \qquad (i,j) \in \mathcal{T}$$

$$w_m \leq \frac{\alpha}{n} \sum_{p=1}^{n} w_p \qquad m = 1, \ldots, n$$

$$(\text{typically use } \alpha = 5\,)$$

*Some other possible objective functions:*

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

$$\begin{aligned} \underset{w,D}{\text{minimize}} \quad & \max_{(i,j)\in\mathcal{S}} \left| D_{ij} - (\mathbf{Target})_{ij} \right| \\ \text{s.t.} \quad & D_{ij} = \sum_{p=1}^{n} D_{ij}^{p}\, w_p \quad (i,j) \in \mathcal{S} \\ & w \geq 0 \end{aligned}$$

# Linear Optimization Models

## Engineered Approaches

*This is the same as:*

$$\min_{\boldsymbol{w},\boldsymbol{D},\boldsymbol{\mu}} \quad \boldsymbol{\mu}$$

$$\textbf{s.t.} \qquad -\boldsymbol{\mu} \le \boldsymbol{D}_{i\,j} - (\textbf{Target})_{i\,j} \le \boldsymbol{\mu} \qquad (i,j) \in \mathcal{S}$$

$$\boldsymbol{D}_{i\,j} = \sum_{p=1}^{n} \boldsymbol{D}_{i\,j}^{p}\, \boldsymbol{w}_p \qquad (i,j) \in \mathcal{S}$$

$$\boldsymbol{w} \ge \mathbf{0}$$

*Here is another model:*

$$\underset{w,D}{\textbf{minimize}} \quad \sum_{(i,j)\in\mathcal{S}} \left| D_{i\,j} - (\textbf{Target})_{i\,j} \right|$$

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

$$w \geq 0$$

*This is the same as:*

$$\min_{w,D,\Delta} \quad \sum_{(i,j)\in\mathcal{S}} \boldsymbol{\Delta}_{i\,j}$$

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

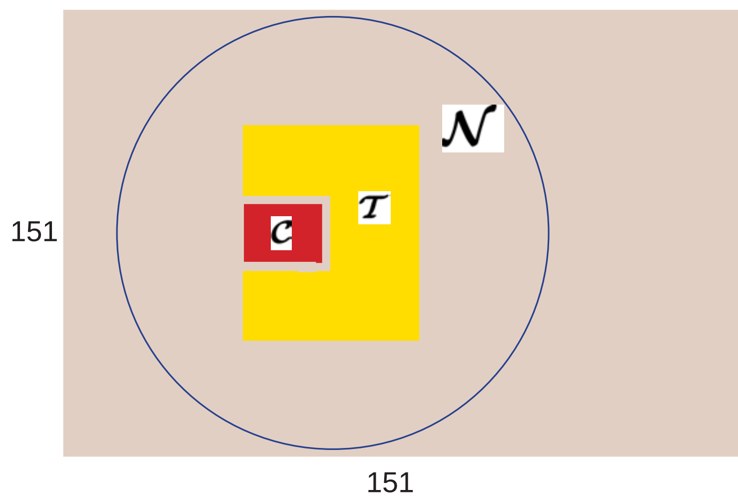
$$w \geq 0$$

$$-\boldsymbol{\Delta}_{i\,j} \leq D_{i\,j} - (\text{Target})_{i\,j} \leq \boldsymbol{\Delta}_{i\,j} \qquad (i,j)\in\mathcal{S}$$

# Computation

## Base Case Model

Consider the “base case” example problem:




$$(\mathbf{Target})_{ij} = \mathbf{16}, \; (i,j) \in \mathcal{T}$$

$$(\mathbf{Target})_{ij} = \mathbf{0}, \quad (i,j) \in \mathcal{C}$$

$$(\mathbf{Target})_{ij} = \mathbf{0}, \quad (i,j) \in \mathcal{N}$$

# Computation

## Base Case Model

$$\underset{w,D,\Delta}{\text{minimize}} \quad 1 \cdot \sum_{(i,j)\in\mathcal{N}} \Delta_{i\,j} + 100 \sum_{(i,j)\in\mathcal{C}} \Delta_{i\,j} + 30 \sum_{(i,j)\in\mathcal{T}} \Delta_{i\,j}$$

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

$$w \geq 0$$

$$-\Delta_{i\,j} \leq D_{i\,j} - (\text{Target})_{i\,j} \leq \Delta_{i\,j} \qquad (i,j) \in \mathcal{S}$$

# Computation

## Size of the Model

### Dimensional Analysis...

$$
\begin{array}{lll}
\underset{w,D,\Delta}{\text{minimize}} & 1 \cdot \sum_{(i,j)\in\mathcal{N}} \Delta_{i\,j} + 100 \sum_{(i,j)\in\mathcal{C}} \Delta_{i\,j} + 30 \sum_{(i,j)\in\mathcal{T}} \Delta_{i\,j} & \\
\text{s.t.} & D_{i\,j} = \sum_{p=1}^{n} D_{i\,j}^{p}\, w_p & (i,j)\in\mathcal{S} \\
& w \geq 0 & \\
& -\Delta_{i\,j} \leq D_{i\,j} - (\text{Target})_{i\,j} \leq \Delta_{i\,j} & (i,j)\in\mathcal{S}
\end{array}
$$

*Dimensional Analysis*:

number of pixels $= 31,397 (\approx \pi * 100^2)$

number of beamlets $= 564 \qquad (n)$

$|\mathcal{T}| = 3,859$; $\quad |\mathcal{C}| = 630$; $\quad |\mathcal{N}| = 26,908$

$|\mathcal{S}| = 31,397$

# Computation

## Size of the Model

### ...Dimensional Analysis...

$$
\begin{array}{lll}
\underset{w,D,\Delta}{\textbf{minimize}} & 1 \cdot \sum_{(i,j)\in\mathcal{N}} \Delta_{ij} + 100 \sum_{(i,j)\in\mathcal{C}} \Delta_{ij} + 30 \sum_{(i,j)\in\mathcal{T}} \Delta_{ij} & \\
\textbf{s.t.} & D_{ij} = \sum_{p=1}^{n} D^p_{ij}\, w_p & (i,j) \in \mathcal{S} \\
& w \geq 0 & \\
& -\Delta_{ij} \leq D_{ij} - (\text{Target})_{ij} \leq \Delta_{ij} & (i,j) \in \mathcal{S}
\end{array}
$$

| Decision Variables | Number |
|---:|---:|
| $D_{ij}$ | $31,397$ |
| $w$ | $564$ |
| $\Delta_{ij}$ | $31,397$ |
| Total | $63,358$ |

# Computation

## Size of the Model

### ...Dimensional Analysis

$$
\begin{array}{lll}
\underset{w,D,\Delta}{\textbf{minimize}} & 1 \cdot \sum\limits_{(i,j)\in\mathcal{N}} \Delta_{i\,j} + 100 \sum\limits_{(i,j)\in\mathcal{C}} \Delta_{i\,j} + 30 \sum\limits_{(i,j)\in\mathcal{T}} \Delta_{i\,j} & \\
\textbf{s.t.} & D_{i\,j} = \sum\limits_{p=1}^{n} D_{i\,j}^{p}\, w_p & (i,j) \in \mathcal{S} \\
& w \geq 0 & \\
& -\Delta_{i\,j} \leq D_{i\,j} - (\text{Target})_{i\,j} \leq \Delta_{i\,j} & (i,j) \in \mathcal{S}
\end{array}
$$

# Computation

## Size of the Model

### Number of Constraints

| Simple Variables Upper/Lower Bounds | Number |
|---|---|
| $w \geq 0$ | $\mathbf{564}$ |
| Total | $\mathbf{564}$ |

| Other Constraints* | Number |
|---|---|
| $D_{ij} =$ | $\mathbf{31,397}$ |
| $\leq D_{ij} - (\text{Target})_{ij} \leq$ | $\mathbf{62,794}$ |
| Total | $\mathbf{94,191}$ |

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

# Computation

## Size of the Model

### Summary

| Variables | Constraints* |
|---|---|
| $\mathbf{63,358}$ | $\mathbf{94,191}$ |

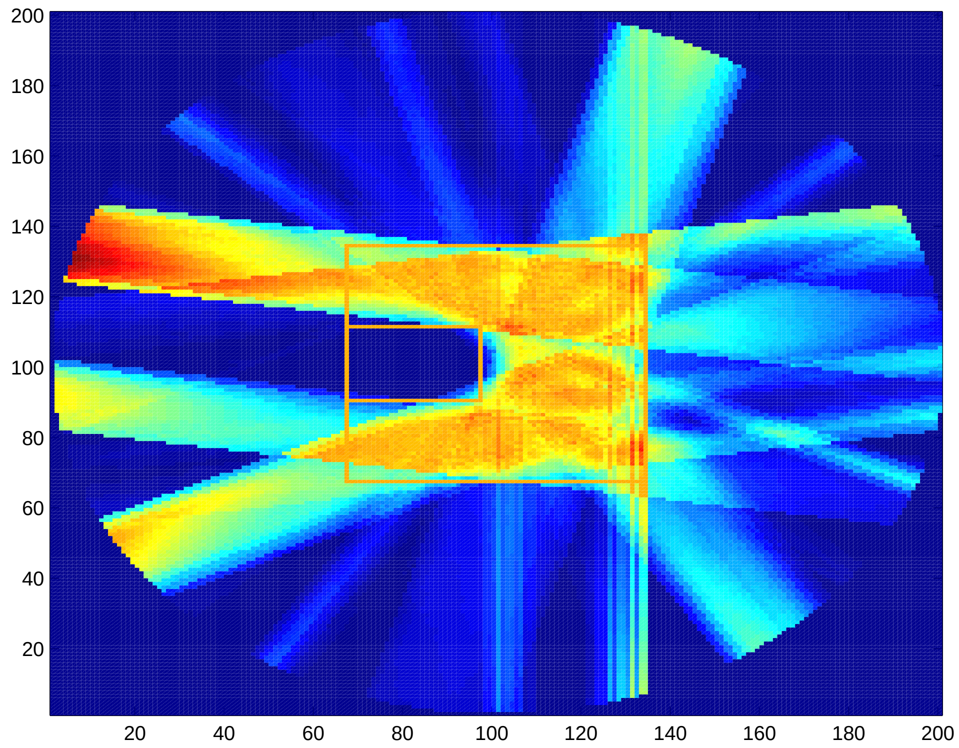
*Excludes variable upper/lower bounds.

# Computation

## Base Case Model

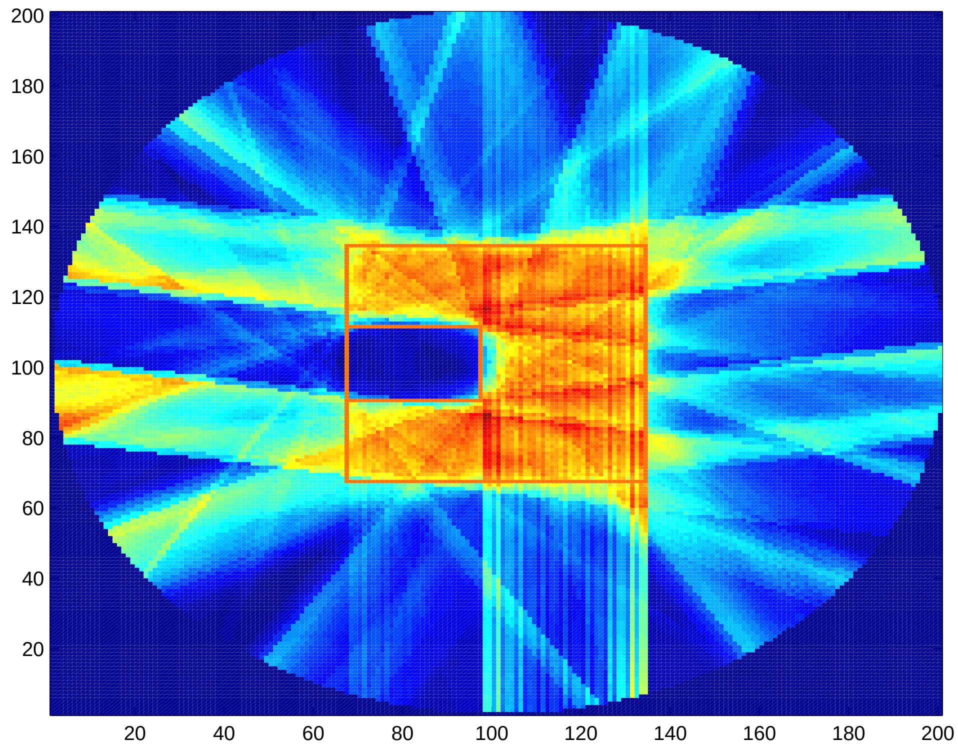
### Optimal Solution




Base Case Model Solution

# Computation

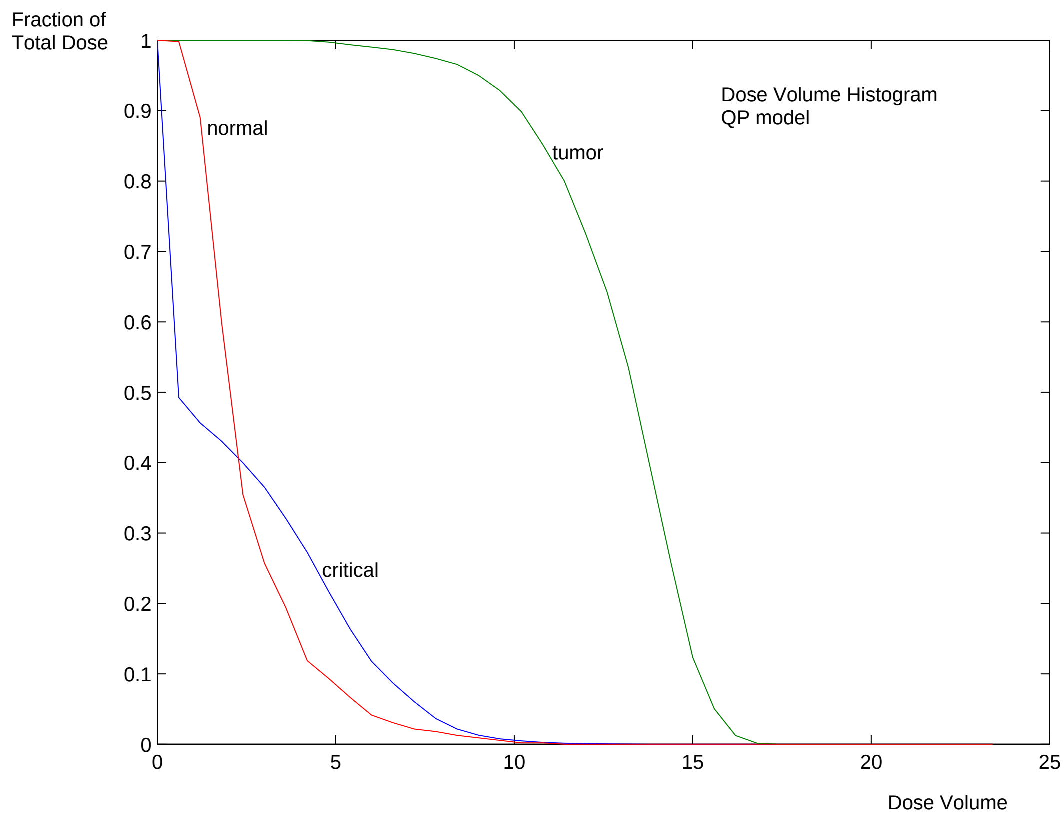
## Another Model Solution




Solution of a nonlinear model.

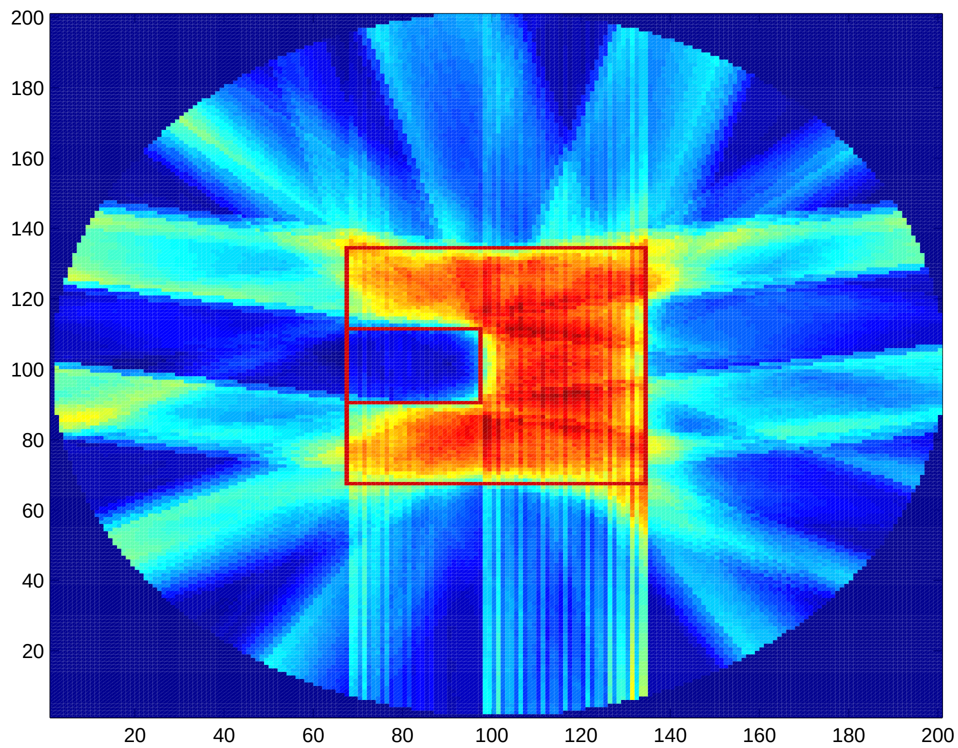
# Computation

# Dose Histogram of Solution



## Another Model Solution




Solution of a nonlinear model, where $\theta_{\mathcal{N}} = \theta_{\mathcal{C}} = \theta_{\mathcal{T}} = 1$.

# Computation

## Computational Issues

### Software/Algorithms

- Software codes:
  - CPLEX simplex (pivoting method)
  - CPLEX barrier
  - LOQO
- Algorithms:
  - Simplex method ("pivoting" method)
  - Interior-point method (IPM) ("barrier" method)

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

# Computation

## Computational Issues

### 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 nonzeroes of matrix systems to be solved
  - other problem characteristics specific to problem
  - idiosyncratic influences
  - pre-processing heuristics

# Computation

**Base Case**

**No Pre-Processing**

- Base Case Model
- No Pre-Processing

| Code | Algorithm | Iterations | Running Time CPU (sec) | Running Time Wall (minutes) |
|---|---|---|---|---|
| CPLEX | Simplex | 183,530 | 440 | 250 |
| CPLEX | Barrier | 49 | 13 | 37 |

# Computation

## Some Generic Rules

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

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \boldsymbol{c}^T\boldsymbol{x} \\ & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{\ell} \leq \boldsymbol{x} \leq \boldsymbol{u} \end{aligned}$$

   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.

# Computation

## Some Generic Rules

$$\begin{aligned} \min_{x} \quad & c^T x \\ & Ax = b \\ & \ell \leq x \leq u \end{aligned}$$

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 $10n$.

# Computation

## Some Generic Rules

$$\min_{\boldsymbol{x}} \quad \boldsymbol{c}^T\boldsymbol{x}$$
$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$
$$\boldsymbol{\ell} \le \boldsymbol{x} \le \boldsymbol{u}$$

5. The work per iteration of an interior-point method generally depends on the structure of the matrix

$$\boldsymbol{K} = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{A}^T \\ \boldsymbol{A} & \boldsymbol{0} \end{pmatrix}.$$

# Computation

## Some Generic Rules

$$\boldsymbol{K} = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{A}^T \\ \boldsymbol{A} & \boldsymbol{0} \end{pmatrix}.$$

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

$$\boldsymbol{K} = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{A}^T \\ \boldsymbol{A} & \boldsymbol{0} \end{pmatrix} \quad \boldsymbol{P} = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{A}^T \\ \boldsymbol{0} & -\boldsymbol{A}\boldsymbol{A}^T \end{pmatrix}.$$

7. The number of interior-point method iterations is typically $\mathbf{25}$–$\mathbf{80}$ (*independent* of $\boldsymbol{m}$ and/or $\boldsymbol{n}$).

# Computation

## Pre-Processing

### Heuristics...

Pre-Processing Heuristics in
Commercial-Grade Software

- Designed to Eliminate Constraints and/or Variables
- Example:

$$-5x \qquad +3y \qquad +z \qquad = 17$$

$$0 \le x \le 4 \quad 0 \le y \le 2 \quad 10 \le z \le 40$$

# Computation

## Pre-Processing

### ...Heuristics...

- Example:

$$\begin{array}{llll} -5x & +3y & +z & = 17 \end{array}$$

$$0 \leq x \leq 4 \quad 0 \leq y \leq 2 \quad 10 \leq z \leq 40$$

- $z = 17 + 5x - 3y \geq 17 + 5(0) - 3(2) = 11 \geq 10$
- $z = 17 + 5x - 3y \leq 17 + 5(4) - 3(0) = 37 \leq 40$
- 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.

# Computation

## Pre-Processing

### ...Heuristics

- Base Case Model
- With Pre-Processing

| Code | Algorithm | Iterations | Running Time CPU (sec) | Running Time Wall (minutes) |
|---|---|---|---|---|
| CPLEX | Simplex | 18,428 | 4.3 | 4 |
| CPLEX | Barrier | 16 | 130 | 133 |

Equivalent Formulation: (eliminate $D_{i\,j}$)

"Small" Model:

$$\begin{aligned}\underset{w,\Delta}{\textbf{minimize}} \quad & 1 \cdot \sum_{(i,j)\in\mathcal{N}} \Delta_{i\,j} + 100 \sum_{(i,j)\in\mathcal{C}} \Delta_{i\,j} + 30 \sum_{(i,j)\in\mathcal{T}} \Delta_{i\,j} \\ \textbf{s.t.} \quad & -\Delta_{i\,j} \le \sum_{p=1}^{n} D^{p}_{i\,j}\, w_p - (\textbf{Target})_{i\,j} \le \Delta_{i\,j} \qquad (i,j) \in \mathcal{S} \\ & w \ge 0\end{aligned}$$

# Computation

## Equivalent Formulation

### ..."Small" Model...

| | Base Case Model | Small Model |
|---|---|---|
| Variables | $\mathbf{63,358}$ | $\mathbf{31,961}$ |
| Constraints* | $\mathbf{94,191}$ | $\mathbf{62,794}$ |

*always excludes simple variable upper/lower bounds

# Computation

## Equivalent Formulation

### ..."Small" Model

- Small Model

| Code | Algorithm | Iterations | Running Time | |
|---|---|---|---|---|
| | | | CPU (sec) | Wall (minutes) |
| CPLEX | Simplex | 171,656 | 390 | 216 |
| CPLEX | Barrier | 57 | 80 | 31 |

# Computation

## Comparisons

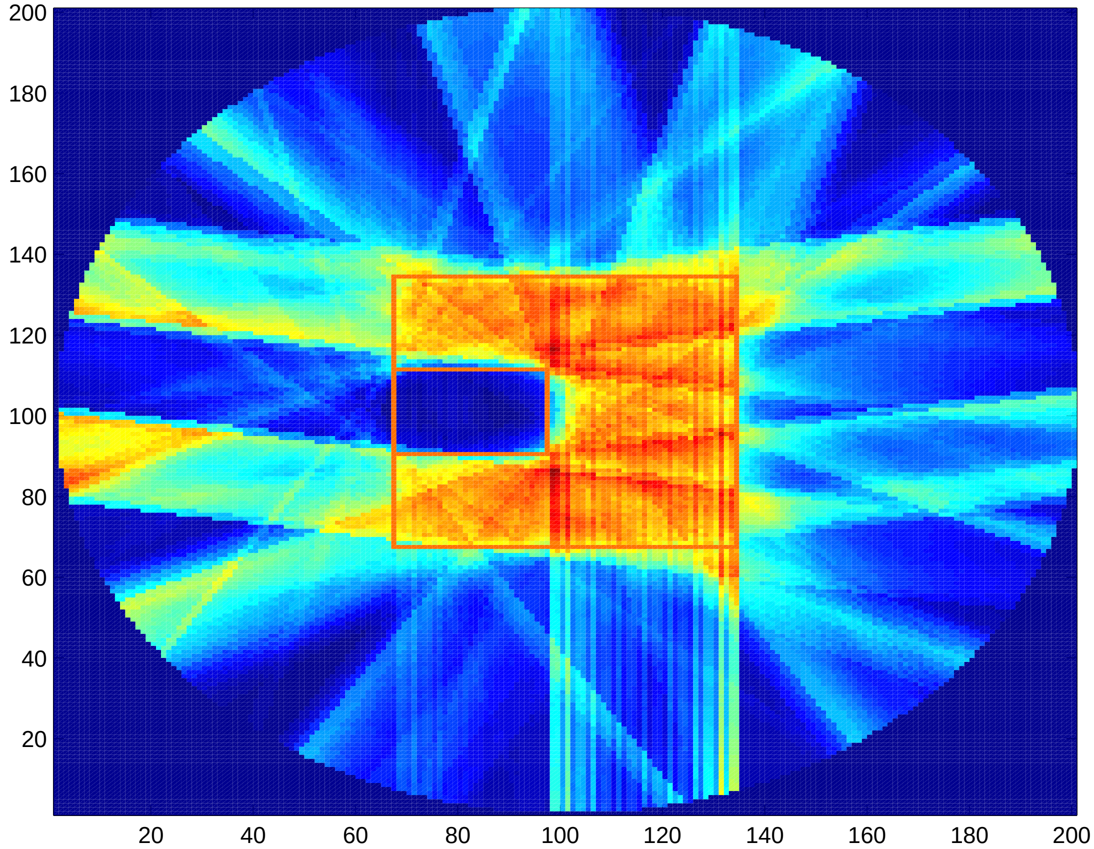
| Code | Algorithm | Model | Running Time Wall (minutes) |
|---|---|---|---|
| CPLEX | Simplex | Base Case | 250 |
| | | Pre-Processed | 4 |
| | | Small Model | 216 |
| CPLEX | Barrier | Base Case | 37 |
| | | Pre-Processed | 133 |
| | | Small Model | 31 |

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

# Nonlinear Optimization

## Quadratic Model

### Quadratic Model Output



## Computational Results

| | | | | Running Time |
|---|---|---|---|---|
| Model | Code | Algorithm | Iterations | CPU (sec) |
| Base Case QP Model | LOQO | Barrier | 31 | 82.7 |
| Small QP Model | LOQO | Barrier | 32 | 149.0 |

# Mixed Integer Optimization

## Limiting the Number of Beamlets

$$
\begin{array}{lll}
\underset{w,D,\Delta}{\text{minimize}} & 1 \cdot \sum_{(i,j)\in\mathcal{N}} \Delta_{i\,j} + 100 \sum_{(i,j)\in\mathcal{C}} \Delta_{i\,j} + 30 \sum_{(i,j)\in\mathcal{T}} \Delta_{i\,j} & \\
\text{s.t.} & D_{i\,j} = \sum_{p=1}^{n} D_{i\,j}^{p}\, w_p & (i,j) \in \mathcal{S} \\
& w \geq 0 & \\
& -\Delta_{i\,j} \leq D_{i\,j} - (\text{Target})_{i\,j} \leq \Delta_{i\,j} & (i,j) \in \mathcal{S} \\
& w_p \leq 100 y_p & p = 1, \ldots, n \\
& y_p \in \{0, 1\} & p = 1, \ldots, n \\
& \sum_{p=1}^{n} y_p \leq 15. &
\end{array}
$$

# Mixed Integer Optimization

## Computation

### CPLEX MIP Solver

| MIP Gap (%) | Simplex Pivots | Running Time CPU (seconds) | Running Time Wall (minutes) |
|---|---|---|---|
| 20 | 11,646 | 7 | 4 |
| 15 | 11,646 | 7 | 4 |
| 12 | 11,646 | 5 | 4 |
| 10 | 14,538 | 9 | 6 |
| 7 | 14,538 | 7 | 6 |
| 5 | 14,538 | 10 | 6 |
| 4 | 14,538 | 7 | 6 |
| 3 | 14,538 | 5 | 6 |
| 2 | 3,655,445 | 1,700 | 25.3 hours |

# Modifications of the Model



*Partial Volume Constraints*:

“No more than $\mathbf{20\%}$ of the critical region can exceed a dose of $\boldsymbol{30G_y}$.”

“No more than $\mathbf{5\%}$ of the critical region can exceed a dose of $\boldsymbol{50G_y}$.”

# Modifications of the Model

## Partial Volume Constraints

*Approach #1* (Integer Programming Model)

Let $M$ be a very large number,

$$\begin{array}{lll} D_{ij} \leq 30 + M \cdot y_{ij}, & y_{ij} \in \{0,1\}, & (i\,j) \in \mathcal{C} \\ D_{ij} \leq 50 + M \cdot z_{ij}, & z_{ij} \in \{0,1\}, & (i\,j) \in \mathcal{C} \end{array}$$

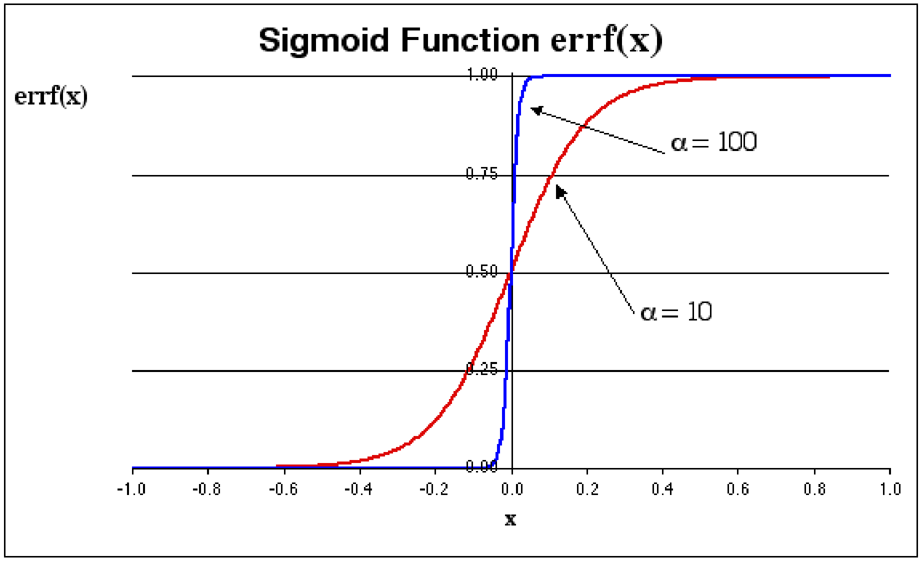
$$\sum_{(i\,j) \in \mathcal{C}} y_{ij} \leq |\mathcal{C}| \times 0.20$$

$$\sum_{(i\,j) \in \mathcal{C}} z_{ij} \leq |\mathcal{C}| \times 0.05$$

*Approach #2* (Error Function Approach)

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

$$\mathbf{err}f(x) = \frac{1}{1 + e^{-\alpha x}}$$




$$\mathbf{err}f(x) = \tfrac{1}{2} \ \textbf{at} \ x = 0$$

$$\mathbf{err}f(x) \to 1 \ \textbf{as} \ x \to \infty$$

$$\mathbf{err}f(x) \to 0 \ \textbf{as} \ x \to -\infty$$

# Modifications of the Model

## Partial Volume Constraints

Instead of integer variables, we use

$$\sum_{(i\,j)\in\mathcal{C}} \mathbf{err}\boldsymbol{f}(\boldsymbol{D}_{i\,j} - \mathbf{30}) \leq |\mathcal{C}| \times \mathbf{0.20}$$

$$\sum_{(i\,j)\in\mathcal{C}} \mathbf{err}\boldsymbol{f}(\boldsymbol{D}_{i\,j} - \mathbf{50}) \leq |\mathcal{C}| \times \mathbf{0.05}$$

# Looking Ahead

## Modeling Languages

### 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

# Looking Ahead

## Modeling Tools 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