Optimization Modelling and Computational Issues in Radiation Therapy

(lecture developed in collaboration with Peng Sun)

February 3, 2004

1 Outline

SLIDE 1

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

SLIDE 2

- 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

SLIDE 3

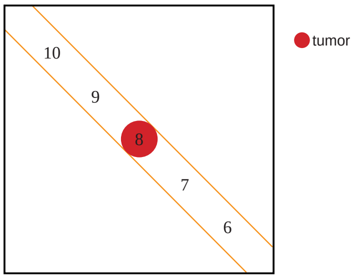
- High doses of radiation (energy/unit mass) can kill cells and/or prevent them from growing and dividing
  - true for cancer cells *and* normal cells
- Radiation is attractive because the repair mechanisms for cancer cells is less efficient than for normal cells

SLIDE 4

- 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.

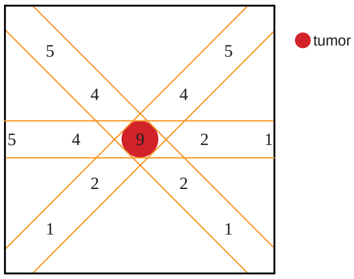
2.2.1 Conventional Radiotherapy

Slide 5



Relative Intensity of Dose Delivered

Slide 6



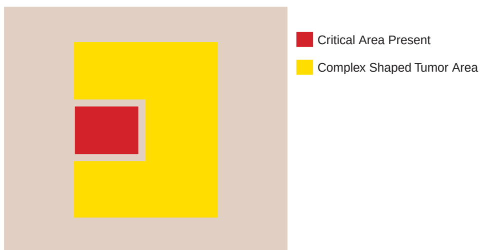
Relative Intensity of Dose Delivered

Slide 7

In conventional radiotherapy

- 3 to 7 beams of radiation
- radiation oncologist and physicist work together to
- determined by manual "trial-and-error" process

Slide 8



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

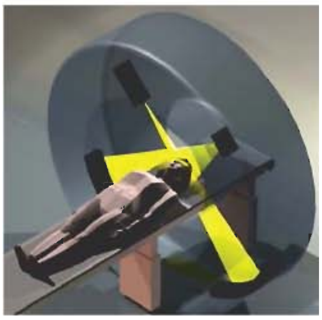
2.2.2 Recent Advances

SLIDE 9

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

SLIDE 10

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

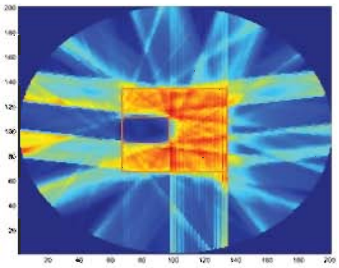


2.2.3 Formal Problem Statement

SLIDE 11

- 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 γ_L
  - dosage over the critical area will be at most a target level γ_U

SLIDE 12

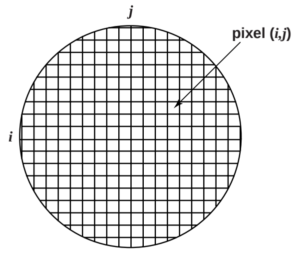


3 Linear Optimization Models

3.1 Discretize the Space

Slide 13

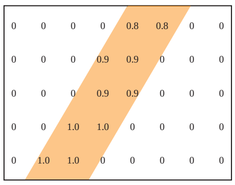
Divide up region into a 2-dimensional (or 3-dimensional) grid of pixels



3.2 Create Beamlet Data

Slide 14

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.

3.3 Dosage Equations

Slide 15

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

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

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

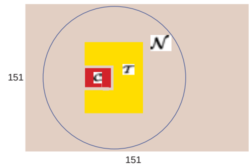
(":=" denotes "by definition")

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

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

3.4 Definitions of Regions

SLIDE 16



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

3.5 Ideal Linear Model

SLIDE 17

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

SLIDE 18

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

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

3.6 Engineered Approaches

SLIDE 19

$$\begin{array}{lll} \underset{w,D}{\text{minimize}} & \theta_{\mathcal{T}} \sum_{(i,j)\in\mathcal{T}} D_{i\,j} + \theta_{\mathcal{C}} \sum_{(i,j)\in\mathcal{C}} D_{i\,j} + \theta_{\mathcal{N}} \sum_{(i,j)\in\mathcal{N}} D_{i\,j} & \\ \text{s.t.} & D_{i\,j} = \sum_{p=1}^{n} D_{i\,j}^{p}\, w_p & (i,j)\in\mathcal{S} \\ & w \geq 0 & \\ & D_{i\,j} \;\geq\; \gamma_{i\,j}^{L} & (i,j)\in\mathcal{T} \\ & w_m \leq 0.05 \sum_{p=1}^{n} w_p & m = 1,\ldots,n \end{array}$$

SLIDE 20

Some other possible objective functions:

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

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

SLIDE 21

This is the same as:

$$\begin{array}{lll} \underset{w,D,\mu}{\text{minimize}} & \mu & \\ \text{s.t.} & -\mu \leq D_{i\,j} - (\text{Target})_{i\,j} \leq \mu & (i,j)\in\mathcal{S} \\ & D_{i\,j} = \sum_{p=1}^{n} D_{i\,j}^{p}\, w_p & (i,j)\in\mathcal{S} \\ & w \geq 0 & \end{array}$$

SLIDE 22

Here is another model:

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

SLIDE 23

This is the same as:

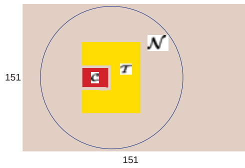
$$\begin{aligned} \underset{w,D,\Delta}{\text{minimize}} \quad & \sum_{(i,j)\in\mathcal{S}} \Delta_{i\,j} \\ \text{s.t.} \quad & D_{i\,j} = \sum_{p=1}^{n} D^{p}_{i\,j}\, w_p \qquad (i,j)\in\mathcal{S} \\ & w \geq 0 \\ & -\Delta_{i\,j} \leq D_{i\,j} - (\text{Target})_{i\,j} \leq \Delta_{i\,j} \qquad (i,j)\in\mathcal{S}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

4 Computation

4.1 Base Case Model

SLIDE 24

Consider the "base case" example problem:



$$\begin{aligned} (\text{Target})_{i\,j} &= 16, \quad (i,j)\in\mathcal{T} \\ (\text{Target})_{i\,j} &= 0, \quad (i,j)\in\mathcal{C} \\ (\text{Target})_{i\,j} &= 0, \quad (i,j)\in\mathcal{N} \end{aligned}$$

SLIDE 25

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

4.2 Size of the Model

4.2.1 Dimensional Analysis

SLIDE 26

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

Dimensional Analysis:

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

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

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

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

SLIDE 27

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

Decision Variables	Number
$D_{i\,j}$	31,397
w	564
$\Delta_{i\,j}$	31,397
Total	63,358

SLIDE 28

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

4.2.2 Number of Constraints

SLIDE 29

Simple Variable Upper/Lower Bounds	Number
$w \geq 0$	564
Total	564

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

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

4.2.3 Summary

SLIDE 30

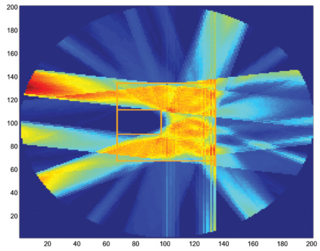
Variables	Constraints*
63,358	94,191

*Excludes variable upper/lower bounds.

4.3 Base Case Model

4.3.1 Optimal Solution

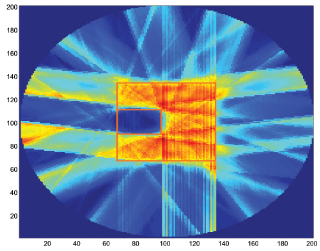
SLIDE 31



Base Case Model Solution

4.4 Another Model Solution

SLIDE 32

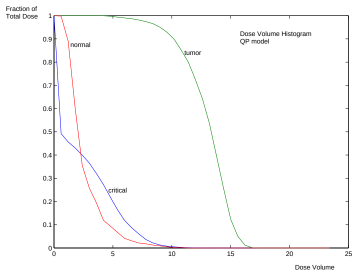


Solution of a nonlinear model.

4.5 Dose Histogram

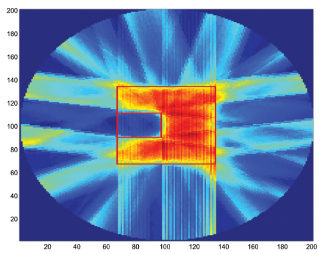
4.5.1 of Solution

SLIDE 33



4.6 Another Model Solution

SLIDE 34



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

5 Computation

5.1 Computational Issues

5.1.1 Software/Algorithms

SLIDE 35

- 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

SLIDE 36

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

5.1.3 Issues in Running Times

SLIDE 37

- 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

5.2 Base Case

5.2.1 No Pre-Processing

SLIDE 38

- Base Case Model
- No Pre-Processing

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

5.3 Some Generic Rules

SLIDE 39

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

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

   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.

SLIDE 40

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

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 $10n$.

SLIDE 41

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

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

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

SLIDE 42

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

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

SLIDE 43

Pre-Processing Heuristics in
Commercial-Grade Software

- Designed to Eliminate Constraints and/or Variables

- Example:

$$\begin{array}{cccccc} -5x & +3y & +z & = & 17 \\ 0 \le x \le 4 & 0 \le y \le 2 & 10 \le z \le 40 \end{array}$$

SLIDE 44

- Example:

$$\begin{array}{cccccc} -5x & +3y & +z & = & 17 \\ 0 \le x \le 4 & 0 \le y \le 2 & 10 \le z \le 40 \end{array}$$

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

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

SLIDE 45

- 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

5.5 Equivalent Formulation

5.5.1 "Small" Model

SLIDE 46

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

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

SLIDE 47

	Base Case Model	Small Model
Variables	$63,358$	$31,961$
Constraints*	$94,191$	$62,794$

*always excludes simple variable upper/lower bounds

SLIDE 48

- Small Model

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

5.6 Comparisons

SLIDE 49

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

6 Nonlinear Optimization

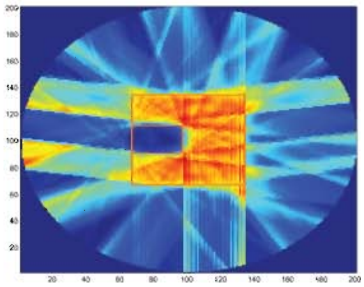
6.1 Quadratic Model

SLIDE 50

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

6.1.1 Quadratic Model Output

SLIDE 51



6.2 Quadratic Model

6.2.1 Computational Results

SLIDE 52

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

7 Mixed Integer Optimization

7.1 Limiting the Number of Beamlets

SLIDE 53

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

7.2 Computation

7.2.1 CPLEX MIP Solver

SLIDE 54

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

8 Modifications of the Model

8.1 Partial Volume Constraints

SLIDE 55

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 $50G_y$."

SLIDE 56

Approach #1 (Integer Programming Model)
Let M be a very large number,

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

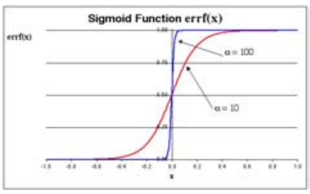
$$\sum_{(i\,j)\in\mathcal{C}} y_{i\,j} \leq |\mathcal{C}| \times 0.20$$

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

SLIDE 57

Approach #2 (Error Function Approach)
The *error function*, or *sigmoid function*, is of the form:

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



SLIDE 58

$$\begin{array}{llll} \mathrm{err}f(x) & = & \frac{1}{2} & \text{at} \quad x = 0 \\ \mathrm{err}f(x) & \to & 1 & \text{as} \quad x \to \infty \\ \mathrm{err}f(x) & \to & 0 & \text{as} \quad x \to -\infty \end{array}$$

Instead of integer variables, we use

$$\sum_{(i\,j)\in\mathcal{C}} \mathrm{err}f(D_{i\,j} - 30) \quad \leq \quad |\mathcal{C}| \times 0.20$$

$$\sum_{(i\,j)\in\mathcal{C}} \mathrm{err}f(D_{i\,j} - 50) \quad \leq \quad |\mathcal{C}| \times 0.05$$

9 Looking Ahead

9.1 Modeling Languages

9.1.1 Used in the Course

SLIDE 59

- 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

SLIDE 60

- "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