High Throughput Computing and Sampling Issues for Optimization in Radiotherapy

Michael C. Ferris, University of Wisconsin
Alexander Meeraus, GAMS Development Corporation

Optimization Days, Montreal, May 2006
Outline

- Introduce a medical application
- Describe a sampling solution approach
- Convert the serial approach to parallel and/or distributed computing.
- Speculation about the resurrection of the service bureau
Radiation Treatment Planning

- Cancer is the 2nd leading cause of death in U.S.
  - Only heart disease kills more
- Expected this year in the U.S. (American Cancer Society)
  - New cancer cases = 1.33 million (> 3,600/day)
  - Deaths from cancer = 556,500 (> 1,500/day)
  - New brain/nerv. sys. cancer cases > 18,300 (> 50/day)
- Cancer treatments: surgery, radiation therapy, chemotherapy, hormones, and immunotherapy
Radiation As Cancer Treatment

- Interferes with growth of cancerous cells
- Also damages healthy cells, but these are more able to recover
- **Goal:** deliver specified dose to tumor (PTV) while avoiding excess dose to healthy tissue and at-risk regions (organs)
Conformal Radiotherapy

- Fire from multiple angles
- Superposition allows high dose in target, low elsewhere
- Beam shaping via collimator
- Gradient across beam via wedges
Conformal Radiotherapy
Example
Beam’s eye view at a given angle is determined based upon the beam source that intersects the tumor.

- The view is constructed using a multi-leaf collimator.
Wedges

- A metallic wedge filter can be attached in front of the collimator.
- It attenuates the intensity of radiation in a linear fashion from one side to other.
- Particularly useful for a curved patient surface.

- 5 positions considered: Open, North, East, South, and West.
Sample Display

[Image of a medical scan with various color-coded regions and numerical values marked on it.]
Dose delivered by a beam of unit weight to voxel (i,j,k) by an angle $A$
Experts determine an ideal dose distribution for a particular target

- Covers target (tumor)
- Limits radiation to healthy/at-risk regions

Delivery plan = optimization problem
Patient Example
Cumulative Dose Volume
\[
\begin{align*}
\text{min} & \quad f(Dose) \\
\text{subject to} & \quad Dose(i) = \sum_A w_A D_A(i) \\
& \quad Dose(Sens(k)) \leq U(k) \\
& \quad L \leq Dose(Target) \\
& \quad w_A \geq 0
\end{align*}
\]

plus some integrality constraints
Mixed Integer Approach

\[ \psi_A = \begin{cases} 
1 & \text{if use angle } A \\
0 & \text{else} 
\end{cases} \]

\[ 0 \leq w_A \leq W \psi_A \]

\[ \sum_A \psi_A \leq K \]

\[ Dose(i) := \sum_A w_A D_A(i) \]
Dose/Volume Constraints

- e.g. no more than 5% of region $R$ can receive more than $U$ Gy

\[(\bar{U} - U) \text{Viol}(i) \geq Dose(i) - U\]

\[\sum_{R} \text{Viol}(i) \leq \frac{5|R|}{100}\]

\[\text{Viol}(i) \in \{0, 1\}\]
Problems

- Large computational times
- Large variance in computing times
- Ineffective restarts (what if trials?)
- Large amounts of data

- Try sampling of voxels (size reduction)
- High level branching (choice reduction)
Solution Times

Pelvis example: solution times for various sample rates;

Processor time, [sec] vs. Sampling rate

1% 3% 5% 7% 9% 11% 13% 15% 17% 19% 21%
True Objective Values

Pelvis example: objective values for various sample rates;
Naïve sampling fails

- Normal tissue
  - Huge numbers of voxels
  - Streaking effects undesirable (hot spots)
  - Use 5x sample on 2nd largest structure

- Small structures
  - Minimum sample size

- Homogeneity/min/max on PTV
  - 2x sample on PTV, rind sampling

- Large gradients on OAR’s
  - 2x sample on OAR’s

- Need adaptive mechanism
Sampling Issues

- For Details see: Sampling Issues for Optimization in Radiotherapy, by Michael C Ferris, David Shepard, Rikhardur Einarsson and Ziping Jiang. Preprint from ferris@cs.wisc.edu).
- Three Phase adaptive Sampling proved to be very successful
- We will only hint at sampling issues
Processing Time

![Diagram showing box plots for processing time across different angles. The x-axis represents angles from 3 to 36, and the y-axis represents processing time in seconds, ranging from 0 to 800.]
Angle Histograms
Proportional Sampling
Three Phase Sampling

- Reduce solution time without compromising quality
- Phase I:
  - Sample 10 times at low rate to predict angles to use
  - Each structure sampled proportionally with largest structure sample limited
  - Determine angles used in “best few” solutions
- Phase II:
  - Increase sample rate, using only proposed angles
- Phase III:
  - Increase sample rate, fix angles and wedge orientations
Sampling Process

- Determine initial sample size
- Phase I: use all angles
  - 10 sample LP’s used to adapt sample
  - 10 adapted sample LP’s solutions determine
- Phase II: use reduced set of angles
  - 10 sample MIP’s determine
- Phase III: use further reduced set
  - Refine sample, solve single MIP, highly accurate solution
Patient Case Results

- **Head/neck case:**
  - Original time: 47,000 secs
  - Phase I time: 5.26 secs/sample
  - Phase II time: 51.21 secs/sample
  - Phase III time: 2.91 secs/sample
  - Solutions same: angles = 40, 140, 230 (+ wedges)

- **Pancreas case:**
  - Original time: 346,000 secs
  - Phase I time: 4 secs/sample
  - Phase II time: 77.31 secs/sample
  - Phase III time: 3.42 secs
  - Solutions same: angles = 80, 290, 350 (+ wedges)
Pelvis case

- 3K prostate, 1.5K bladder, 1K rectum, 557K normal
- Time for “full problem”: 12.5K secs
- Time Phase I: 32 secs/sample
- Time Phase II: 18 secs/sample
- Time Phase III: 147 secs
- Solution: 80, 110, 130, 240, 270, 320
Dose Histogram

Fraction of Volume vs. Relative Dose

- Blue line: Prostate
- Red line: Bladder
- Black line: Rectum
- Green line: Normal
Axial Slice
Key contributions

- Use multiple (small) samples and multiple phases to determine plan
- Adaptive sampling via linear program solution
- High level branching via multiple samples and ranking
- Significant time reductions without loss in quality
- Applicable to more general treatment planning domains, and MIP applications
Remaining Issues

- Overall solution times still high
- Would like to consider more angles
- Work with higher sampling rates
- Use more samples
- Exhausted smart modeling
- Considerer high throughput computing
- How to convert from serial to parallel and distributed computing
New Opportunities

- The original model was implemented in GAMS and used CPLEX
- GAMS introduced an experimental grid computing facility
- High Throughput Computing via the Condor system and the SUN Grid Engine connected to GAMS
- Multi CPU desktop systems available
A pool of connected computers managed and available as a common computing resource

- Allows parallel task execution
- Allows effective sharing of CPU power
- Licensing issues
- Scheduler handles management tasks
- Can be rented or owned in common
- E.g. Condor, Sun Grid Engine, Globus
Economics of Grid Computing

- Yearly cost, 2-CPU workstation: $5200
  - Hardware - $1200
  - Software - $4000
- Hourly cost on the grid: $2
  - $1/hour for CPU time (to grid operator)
  - $1/hour for software (GAMS, model owner)
- 1 workstation == 50 hrs/week grid time
- Up-front vs. deferred, as-needed costs
Use a GAMS Grid

- Solve the samples in parallel, e.g.
  - 200 CPUs: 15 minutes
- Marginal cost is $100
- No programming required (almost)
- Model stays maintainable
- Separation of model and solution maintained
Results for 4096 MIPS

- Submission start Jan 11 at 16:00 pm
- All job submitted by Jan 11 at 23:00 pm
- All jobs returned by Jan 12, 12:40 pm
  - 20 hours wall time, 5000 CPU hours
  - Peak number of CPUs: 500
- Different Instance:
  - 24 hours wall time, 3000 CPU hours
Serial Solve Loop

\[ \text{loop}(s, \]
\[ \quad b(j) = \text{dem}(s,j); \]
\[ \textbf{Solve} \ tr \text{ using lp minimizing } z; \]
\[ \quad \text{repx}(s,i,j) = x.l(i,j); \]
\[ \quad \text{repy}(s,'\text{solvestat}') = tr.\text{solvestat}; \]
\[ \quad \text{repy}(s,'\text{modelstat}') = tr.\text{modelstat} ); \]
parameter $h(s)$ store the instance handle;

```
tr.solvelink = 3;  // turn on grid option
loop(s,
    b(j) = dem(s,j)
    Solve tr using lp minimizing z;
    h(s) = tr.handle ); // save instance handle
```
Solution Collection Loop

Repeat

loop(s|h(s),

    if(handlestatus(h(s))=2,
        tr.handle = handle(s); execute_loadhandle tr;
        repx(s,i,j) = x.l(i,j); repy(s,'solvestat') = tr.solvestat;
        repy(s,'modelstat') = tr.modelstat;
        display$handedelete(h(s)) 'Could not remove handle';
        h(s) = 0) ; // indicate solution is loaded

    if(card(h), execute 'sleep 1');

    until card(h) = 0 or timeelapsed > 100;
Grid Specifics Scripts

```
#!/bin/bash

# Script to run a bash command with input parameters

# Write the bash command to a file
echo "#!/bin/bash" > ${3}runit.sh

# Write the input parameters to the file
echo $1 $2 >> ${3}runit.sh

# Append the gmscr_ux.out and $2 to the file
echo gmscr_ux.out $2 >> ${3}runit.sh

# Create a directory for the output
echo mkdir ${3}finished >> ${3}runit.sh

# Make the script executable
chmod 750 ${3}runit.sh

# Run the script and redirect output to /dev/null
${3}runit.sh > /dev/null &
```
@echo off > %3runit.cmd
echo %1 %2 >> %3runit.cmd
gmscr_nx.exe %2 >> %3runit.cmd
echo mkdir %3finished >> %3runit.cmd
echo echo exit >> %3runit.cmd
start /b /BELOWNORMAL %3runit.cmd > nul
Conclusions

- Massive parallel and distributed computing environments are becoming available (SUN just introduced a 5000 node network in the US).
- Simple language extensions in existing modeling systems provide easy access.
- Today's modeling languages are well suited to experiment with coarse grain parallel approaches for solving difficult problems.