Optimization in Radiotherapy: A Review

Michael C. Ferris

University of Wisconsin-Madison

Tomotherapy, January 16, 2008

Abstract

Optimization tools are effective in many application areas. Over the past decade, optimization models have been used extensively in the field of radiation therapy. We outline some of the recent optimization developments that have had some impact in this area, and describe other techniques that have promise for future application. Some discussion of perceived advantages and limitations of these approaches will be given.

The problem

$$\min F(d) \text{ s.t. } d = Px, x \in X, d \in D$$

- P is the pencil beam matrix, x are bixel weights
- X represents constraints on the bixel weights (typically $x \geq 0$, or cardinality restrictions)
- D represents constraints on the dose distribution (bound constraints, DVH-constraints)

Issues

$$\min F(d) \text{ s.t. } d = Px, x \in X, d \in D$$

- P is extremely large and dense
- Must solve problem relatively quickly
- Ability to modify solution quickly

In many cases, objective is made up of a weighted sum of objectives:

$$F(d) = \sum_s w_s F^s(d)$$

s may range, for example, over *structures*

Concrete setting: QP

Can choose $F^s(d)$ for each structure s to be a quadratic violation penalty

$$\min_{d,x} \sum_s w_s \left\| d^s - \bar{d}^s \right\|_2^2 \text{ s.t. } d = Px, x \geq 0$$

leading to a bound constrained quadratic program

$$\min_x \sum_s w_s (P_{S.}x - \bar{d}^s)^T (P_{S.}x - \bar{d}^s) \text{ s.t. } x \geq 0$$

Key step for algorithms (gradient projection, two-metric projection, conjugate gradients):

- calculate $P_{S.}^T P_{S.} v$ for any v and any s
- or $P_{I.}^T P_{I.} v$ where $I \subseteq S$

Alternatives: EUD

- $EUD_{R,a}(d) = \left(\frac{1}{|R|} \sum_{i \in R} d_i^a \right)^{1/a}$ if $d > 0$
- Convex for $a \geq 1$ and concave for $a \leq 1$.
- $U_{R,a,\nu,EUD_0}(d) = \left(1 + \left(\frac{EUD_{R,a}(d)}{EUD_0} \right)^{\nu} \right)^{-1}$
- $L_{R,a,\nu,EUD_0}(d) = \left(1 + \left(\frac{EUD_0}{EUD_{R,a}(d)} \right)^{\nu} \right)^{-1}$
- Choose F^s as $-\ln U$ or $-\ln L$
- Solve $\min_{x \geq 0} F(Px)$ using two-metric projection.
- Approximate Hessian of F on active and inactive set.
- Solve for direction using conjugate gradient steps (uses result of Alber et al).

Alternatives: Linear Programming

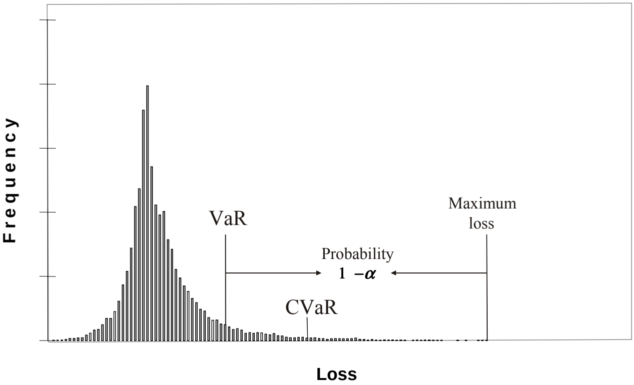
$$F(d) = \left\| d - \bar{d} \right\|_1$$

$$\min \sum_i d_i^+ + d_i^- \text{ s.t. } d^+ - d^- = Px, \; x, d^+, d^- \geq 0$$

Similarly any piecewise linear convex penalty leads to a linear program.

- How to restart? $P \to P + \Delta P$
- Barrier method - hard
- Simplex method - can generate a feasible solution easily

$\overline{CVaR}_\alpha(d)$



$\overline{CVaR}_\alpha$: mean of upper tail at level α the average dose received by the subset of relative volume $(1-\alpha)$ receiving the highest dose. (Think of $\alpha = 0.95$ and this is then the mean of the upper tail, ie those values beyond the 95th percentile).

Key observation

Clear that this is equal to the average dose of the $(1-\alpha)N$ voxels (point volumes) receiving highest dose.
Rewriting this in symbols:

$$\overline{CVaR}_\alpha(d) = \overline{VaR}_\alpha(d) + \frac{1}{(1-\alpha)N} \sum_{j=1}^{N} \left(d - \overline{VaR}_\alpha(d)\right)_+$$

Thus $\overline{CVaR}$ is just $\overline{VaR}$ moved to the right by the average of the tail.
The next step is a clever theorem due to Ogryczak and Tamir (2003) that states this expression can be written as:

$$\overline{CVaR}_\alpha(d) = \min_{a \in \mathbf{R}} \left\{ a + \frac{1}{(1-\alpha)N} \sum_{j=1}^{N} (d-a)_+ \right\}$$

Thus can impose linear constraints to get $\overline{CVaR}_\alpha(d) \le U$

Generalization

Note that the "average" term can be defined slightly more generally in terms of expectations, so in fact we could write the last expression in its general form as:

$$\overline{CVaR}_\alpha = \min_{a \in \mathbf{R}} \left\{ a + \frac{1}{(1-\alpha)} \mathbf{E} \left(d - a \right)_+ \right\}$$

Other ways possible to shape dose distribution (Ferris et al)

Apertures instead of beamlets

$$\min F(d) \text{ s.t. } d = Ax, x \geq 0$$

- Columns of A are the dose deposited to voxels from an aperture, not a pencil beam.
- x is now the aperture weight.
- Note that each column of A corresponds to a sum of columns of P.
- Too many apertures, generate them on the fly.

Master problem

Only use a subset J of the apertures:

$$\min F(d) \text{ s.t. } d = A_{\cdot J} x_J, x_J \geq 0$$

Optimality conditions:

$$\begin{aligned} \pi &= \nabla F(d) \\ d &= A_{\cdot J} x_J \\ 0 \leq A_{\cdot J}^T \pi \quad &\perp \quad x_J \geq 0 \end{aligned}$$

So solving Master Problem gives solution x and d *and* π.

Pricing problem

Generate a new column a of A that violates optimality conditions:

$$a^T \pi < 0$$

Aperture is made up of sum of beamlets; choose beamlets to

$$\min_w (Pw)^T \pi \text{ s.t. } w_j \in \{0, 1\}, \; w \in W$$

- Decomposes over beams (i.e. $P = [P^1 \; P^2 \; \cdots P^n]$)
- Easy to solve if all apertures are feasible
- Specialized subproblems (network flows) for
  - Interdigitation
  - Disconnected apertures
- DAO approach is related, generates columns of A on the fly via simulated annealing, limits number of apertures from each angle.

IMAT

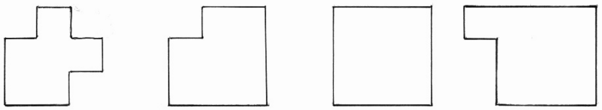
- Suppose you have a number of sweeps (around target)
- Idea 1: sequencer produces a set of shapes at each angle in the sweep (same number of shapes at each angle)
- Assign shapes to sweeps to minimize total leaf distance, or maximum leaf distance, etc
- Problem is an easy network flow problem.
- How much can weights change as you move?

IMAT

- Idea 2: adapt DAO (simulated annealing)
- Given forced change of shape on one sweep, minimize TLD on sweep to get feasible.
- Update weights based on new shapes.

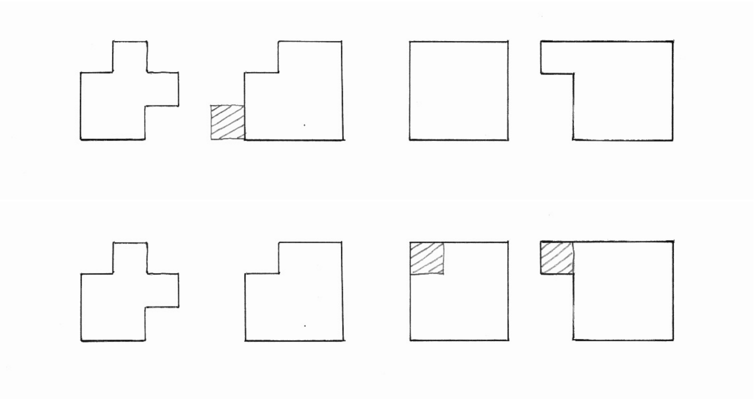
Example of Apertures in a Sweep

Four possible apertures in a given sweep. Assumed that each horizontal leaf can only move one unit between each angle.



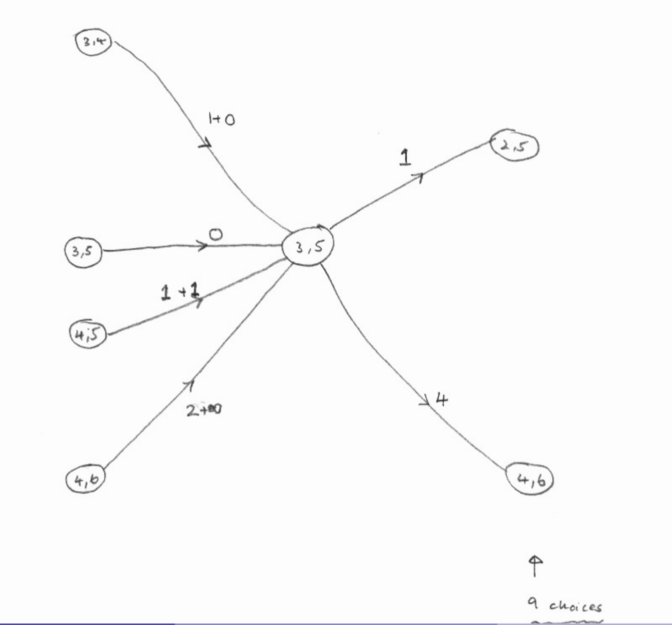
Neighborhood sweep

In first case, change is only made to one aperture, others remain feasible.



In second case, change in third aperture forces a change in fourth aperture.

Network formulation



Update of weights

$$\min F(Ax) \text{ s.t. } x \geq 0$$

$$A \rightarrow A + \Delta A$$

$$x(\alpha) = \left(x^i - \alpha A^T \nabla F(Ax^i)\right)_+$$

Choose α by linesearch on $F(x(\alpha))$.

Other extensions

$$d = \sum_{b \in B} A^b x^b, \; x^b \geq 0, \forall b \in B$$

$$A^b = \left[A^b_{s1} \; A^b_{s2} \; \cdots A^b_{sn} \right]$$

- Do we need $\left\| x^b - x^{b+1} \right\|_\infty \leq \delta$?
- Could implement this as $Cx \leq f$.
- Could we do column generation here?
- But arrange via sweeps, i.e. change A^b_s for a sweep or add a sweep.

Other work of interest

- Sampling (Ferris et al, Martin)
- Robust Optimization (Chan et al, Wright et al)
- Stochastic Optimization
- Image segmentation (Hochbaum)