Mathematical Optimization in Radiotherapy Treatment Planning

Ehsan Salari

Department of Radiation Oncology
Massachusetts General Hospital and Harvard Medical School

HST S14
May 6, 2013
Outline

1. Intensity-modulated Radiotherapy (IMRT)
2. Sequential Method: Fluence-map Optimization (FMO)
3. Sequential Method: Leaf Sequencing (LS)
4. Direct Aperture Optimization (DAO)
Intensity-modulated Radiotherapy (IMRT)

- In 3D-CRT, radiation fluence across the opening area of the aperture is constant.
- To better spare organs-at-risk more fluence modulation is needed.
- *Intensity-modulated radiotherapy* (IMRT) is a more recent modality that allows for more fluence modulation at each beam.

*Figure:* [Webb, 2001]
Comparing 3D-CRT and IMRT

- 3D-CRT shapes apertures that conform to tumor shape
- IMRT creates a *fluence map* (intensity profile) per beam
Multi-leaf Collimator (MLC)

- In IMRT
  - gantry head is equipped with a *multi-leaf collimator* (MLC) system
  - MLC leaves form apertures with different shapes and intensities

![MLC Image](image1)

![IMRT Image](image2)
Creating Fluence Maps using MLC

- Using MLC a desired fluence map can be created

![Fluence Maps Example]

- Example Fluence Maps:
  - Row 1: 2 2 0 0
  - Row 2: 2 2 2 0
  - Row 3: 0 0 0 0
  - Row 4: 0 0 0 0
  - Row 5: 0 0 0 0

- Graph showing intensity distribution with Column and Row indices.
IMRT planning is to determine a set of apertures and their intensities that yield a dose distribution that

- adequately covers target(s)
- preserves functionality of critical structures
Solution Approaches to IMRT Treatment Planning

- **Sequential method**
  1. **Beam orientation optimization (BOO)**
     - determines a set of beam directions
     - is usually performed manually
  2. **Fluence-map optimization (FMO)**
     - determines an intensity profile for each beam
  3. **Leaf sequencing (LS)**
     - decomposes intensity profiles to deliverable apertures

- **Direct aperture optimization (DAO)**
  - integrates FMO and LS
Solution Approaches to IMRT Treatment Planning

- **Sequential method**
  - (1) **Beam orientation optimization** (BOO)
    - determines a set of beam directions
    - is usually performed manually
  - (2) **Fluence-map optimization** (FMO)
    - determines an intensity profile for each beam
  - (3) **Leaf sequencing** (LS)
    - decomposes intensity profiles to deliverable apertures

- **Direct aperture optimization** (DAO)
  - integrates FMO and LS
Solution Approaches to IMRT Treatment Planning

- **Sequential method**
  1. **Beam orientation optimization** (BOO)
     - determines a set of beam directions
     - is usually performed manually
  2. **Fluence-map optimization** (FMO)
     - determines an intensity profile for each beam
  3. **Leaf sequencing** (LS)
     - decomposes intensity profiles to deliverable apertures

- **Direct aperture optimization** (DAO)
  - integrates FMO and LS
Solution Approaches to IMRT Treatment Planning

- **Sequential method**
  1. *Beam orientation optimization* (BOO)
     - determines a set of beam directions
     - is usually performed manually
  2. *Fluence-map optimization* (FMO)
     - determines an intensity profile for each beam
  3. *Leaf sequencing* (LS)
     - decomposes intensity profiles to deliverable apertures

- *Direct aperture optimization* (DAO)
  - integrates FMO and LS
Solution Approaches to IMRT Treatment Planning

- **Sequential method**
  1. **Beam orientation optimization (BOO)**
     - determines a set of beam directions
     - is usually performed manually
  2. **Fluence-map optimization (FMO)**
     - determines an intensity profile for each beam
  3. **Leaf sequencing (LS)**
     - decomposes intensity profiles to deliverable apertures

- **Direct aperture optimization (DAO)**
  - integrates FMO and LS
Fluence-map Optimization (FMO)

- Rectangular beams are discretized into \textit{beamlets} \( i \in I \)
- Using pencil-beam dose calculation method, \textit{beamlet dose deposition coefficients} \( D = [D_{iv}] (i \in I, v \in V) \) are computed
- Using optimization methods, optimal fluence maps \( x_i (i \in I) \) are determined
  - large-scale problem: \( \mathcal{O} (10^3) \) beamlets and \( \mathcal{O} (10^5) \) voxels
FMO Mathematical Formulation

- Mathematical formulation for the FMO problem

\[
\begin{align*}
\min \ G(d) \\
\text{subject to} \\
d &= D^\top x \\
H(d) &\leq 0 \\
x &\geq 0
\end{align*}
\]

- Notation
  - \( x \): vector of beamlet intensities
  - \( D = [D_{iv}] \): matrix of beamlet dose deposition coefficients
FMO problem can be solved using *interior-point method* (*barrier method*)

- see [Bazaraa et al., 2006]
- transform the constrained problem to unconstrained problem using barrier function
- sets a barrier against leaving the feasible region

\[
\min_{\mathbf{x}} \ G \left( D^\top \mathbf{x} \right) + \mu \underbrace{B(\mathbf{x})}_{\text{barrier}} \quad \mu > 0
\]

Barrier function characteristics are

- nonnegative and continuous over \( \{ \mathbf{x} : \mathbf{x} \geq 0, H(D^\top \mathbf{x}) \leq 0 \} \)
- approaches \( \infty \) as \( \mathbf{x} \) approaches the boundary from interior
Barrier Method for FMO

- We formulate a parametric problem

\[ \phi(\mu) = \min_x G(D^T x) + \mu B(x) \]

- It can be shown that

\[ \lim_{\mu \to 0^+} \phi(\mu) = \min_x \left\{ G(D^T x) : x \geq 0, H(D^T x) \leq 0 \right\} \]
Barrier Method for FMO

1. Initialize: choose interior point $\mathbf{x}_0 > \mathbf{0}$, $\mu_0 > 0$, and $0 < \beta < 1$

2. Main step: at iteration $k$ solve unconstrained problem

$$\min_{\mathbf{x}} \ G\left( D^\top \mathbf{x} \right) + \mu_k B\left( \mathbf{x} \right)$$

3. Termination condition: if $\mu_k B\left( \mathbf{x}_k \right) < \epsilon$, stop; otherwise, $\mu_{k+1} = \beta \mu_k$ and go to step 2
Consider unconstrained optimization problem

\[
\min F(x)
\]

\(\bar{x}\) is a \textit{global minimum} if \(F(\bar{x}) \leq F(x)\) for all \(x \in \mathbb{R}^n\)

\(\bar{x}\) is a \textit{local minimum} if there is an \(\epsilon\)-\textit{neighborhood} \(N_\epsilon(\bar{x})\) around \(\bar{x}\) such that \(F(\bar{x}) \leq F(x)\) for all \(x \in N_\epsilon(\bar{x})\)
  
  - we assume differentiability
  - see [Bazaraa et al., 2006]
Characterizing Local Minimum

- \textbf{s} is a \textit{descent} direction at \( \bar{x} \) if

\[
\lim_{\lambda \to 0^+} \frac{F(\bar{x} + \lambda s) - F(\bar{x})}{\lambda} = \nabla F(\bar{x})^\top s < 0
\]

- Necessary condition: if \( \bar{x} \) is a local minimum, then \( \nabla F(\bar{x}) = 0 \)

- Sufficient condition: if \( \nabla F(\bar{x}) = 0 \) and \( \nabla^2 F(\bar{x}) \succ 0 \), then \( \bar{x} \) is a local minimum
Class of Convex Functions

- **Convex functions**
  - **Definition** \( \forall \bar{x}, \hat{x} \in \mathbb{R}^n \)
  
  \[
  F(\lambda \bar{x} + (1 - \lambda)\hat{x}) \leq \lambda F(\bar{x}) + (1 - \lambda)F(\hat{x}) \quad \lambda \in (0, 1)
  \]

  - \( F \) is convex if and only if \( \nabla^2 F \) is positive semi-definite everywhere
  - If \( F \) is convex, then \( \bar{x} \) is a global minimum if and only if \( \nabla F(\bar{x}) = 0 \)
    - a desired property for unconstrained optimization
Steepest Descent for Unconstrained Optimization

- Starting from \( \bar{x} \) it iteratively moves toward local minimum
- *Steepest descent* at \( \bar{x} \) can be obtained by
  \[
  \min \nabla F (\bar{x})^\top s
  \]
  subject to
  \[
  \|s\| \leq 1
  \]
  which yields
  \[
  s = - \frac{\nabla F (\bar{x})}{\|\nabla F (\bar{x})\|}
  \]
Steepest Descent for Unconstrained Optimization

1. Initialize: Let $\epsilon > 0$, choose starting point $x_0$

2. Steepest descent direction: At iteration $k$, let

$$s_k = -\frac{\nabla F(x_k)}{\|\nabla F(x_k)\|}$$

3. Termination condition: If $\|s_k\| < \epsilon$ stop; else, go to step 4

4. Line search:

$$\lambda^* = \arg\min_{\lambda \geq 0} F(x_k + \lambda s_k)$$

5. Update solution: $x_{k+1} = x_k + \lambda^* s_k$ and go to step 2
Line Search

- **Line search** is to find optimal step length to move from point $\mathbf{x}$ along direction $\mathbf{s}$
- It is rarely possible to obtain analytical solutions

$$
\frac{\partial}{\partial \lambda} F(\mathbf{x} + \lambda \mathbf{s}) = \mathbf{s}^\top \nabla F(\mathbf{x} + \lambda \mathbf{s}) = 0
$$

- Numerical methods are commonly used
  - see [Bazaraa et al., 2006]
Line Search: Uncertainty Interval

- Derivative-free numerical solution method for

\[ \min_{a \leq \lambda \leq b} \theta (\lambda) = F(x + \lambda s) \]

- To reduce uncertainty interval \([a, b]\) we evaluate \(\theta (\lambda)\) for different \(\lambda \in [a, b]\)

- Suppose \(\theta\) is strictly quasi-convex (unimodal). Let \(\lambda_1, \lambda_2 \in [a, b]\)
  - If \(\theta (\lambda_1) \leq \theta (\lambda_2)\), then \(\theta (\lambda) \geq \theta (\lambda_1)\) for \(\lambda \in [\lambda_2, b]\)
  - If \(\theta (\lambda_1) \geq \theta (\lambda_2)\), then \(\theta (\lambda) \geq \theta (\lambda_2)\) for \(\lambda \in [a, \lambda_1]\)
Line Search: Dichotomous Search

1. Initialize: set initial uncertainty interval $[a_0, b_0]$, distinguishing param. $2\epsilon > 0$, and threshold param. $\delta$

2. Main step: let $\lambda_1 = \frac{a_k + b_k}{2} - \epsilon$ and $\lambda_2 = \frac{a_k + b_k}{2} + \epsilon$, then

$$[a_{k+1}, b_{k+1}] = \begin{cases} [a_k, \lambda_2] & \text{if } \theta(\lambda_1) \leq \theta(\lambda_2) \\ [\lambda_1, b_k] & \text{otherwise} \end{cases}$$

3. Termination condition, if $b_{k+1} - a_{k+1} < \delta$ stop; otherwise, go to Step 2
Example of FMO Mathematical Formulation

- Dose evaluation criteria: summation of piecewise quadratic voxel-based penalties for all relevant structures $s \in S$

$$\min_{x \geq 0} G \left( D^T x \right) = \sum_{s \in S} \sum_{v \in V_s} \gamma_s^+ \max \left\{ \sum_{i \in I} D_{iv} x_i - t_v, 0 \right\}^2$$

overdosing penalty

$$+ \gamma_s^- \max \left\{ t_v - \sum_{i \in I} D_{iv} x_i, 0 \right\}^2$$

underdosing penalty

- We assume only nonnegativity constraints $x \geq 0$, results can be generalized to include dose constraints
Example of FMO Mathematical Formulation

- Logarithmic barrier function for nonnegativity of beamlet intensities

\[ \phi(\mu) = \min_x G\left(D^T x\right) - \mu \sum_{i \in I} \ln(x_i) \]

- We solve \( \phi(\mu) \) for \( \mu > 0 \) using Steepest Descent method

- Alternatively we can use primal-dual interior-point method
  - see [Aleman et al., 2010]
Primal-dual Interior Point Method

- To obtain $\phi(\mu)$ we find $x^*$ where gradient vanishes

$$\frac{\partial G(D^T x)}{\partial x_i} - \mu \sum_{i \in I} \ln(x_i) = \frac{\partial G(D^T x)}{\partial x_i} - \frac{\mu}{x_i} = 0 \quad i \in I$$

- Variable transformation ($x, \lambda$: primal and dual variables)

$$\lambda_i = \frac{\mu}{x_i} \quad i \in I$$

- Solve nonlinear system of equations for $x, \lambda$

$$\nabla_x G - \lambda = 0$$

$$\Lambda \cdot X = \mu e$$
Primal-dual Interior Point Method

- Newton method to solve nonlinear system of equation
  1. Main step: determine direction and step length
     \[
     \begin{pmatrix}
     \Delta x_{(k)} \\
     \Delta \lambda_{(k)}
     \end{pmatrix}
     =
     -
     \begin{pmatrix}
     \nabla^2_{xx} G_{(k)} & -I \\
     \Lambda_{(k)} & \Lambda_{(k)} \cdot X_{(k)}
     \end{pmatrix}
     ^{-1}
     \begin{pmatrix}
     \nabla_x G_{(k)} - \lambda \\
     \Lambda_{(k)} \cdot X_{(k)} - \mu_{(k)} e
     \end{pmatrix}
     \]
  2. Update solution
     \[
     \begin{pmatrix}
     x_{(k+1)} \\
     \lambda_{(k+1)}
     \end{pmatrix}
     =
     \begin{pmatrix}
     x_{(k)} \\
     \lambda_{(k)}
     \end{pmatrix}
     +
     \alpha_{(k)}
     \begin{pmatrix}
     \Delta x_{(k)} \\
     \Delta \lambda_{(k)}
     \end{pmatrix}
     \]
  3. Termination condition: if \( x_{(k+1)}^\top \lambda_{(k+1)} < \epsilon \), then stop; otherwise go to Step 1
Leaf Sequencing Problem

How to decompose fluence map into collection of deliverable apertures?

- we assume *step-and-shoot* delivery
- apertures are *binary matrices* with *consecutive ones* at each row
Leaf Sequencing (LS)

- There is a large number of possible decompositions
- *Leaf Sequencing* (LS) aims at finding decomposition with
  - minimal total monitor units (*beam-on time*)
  - minimal number of binary matrices
  - total treatment time depends on beam-on time and number of apertures
- Assumptions
  - there is only *row-convexity* constraint on aperture shapes
    - see [Baatar et al., 2005] for additional MLC hardware constraints
  - integral intensities by rounding fluence map $X$
**LS Formulation: Minimizing Beam-on Time**

- Beam-on time minimization
  \[
  \min \sum_{k \in K} \alpha_k
  \]
  subject to
  \[
  X = \sum_{k \in K} \alpha_k S_k
  \]
  \[
  \alpha_k \geq 0 \quad k \in K
  \]

- Notation
  - \( K \): set of \( M \times N \) binary matrices with consecutive-ones property at each row
  - \( S_k \): binary matrix \( k \in K \)
  - \( \alpha_k \): number of MU for binary matrix \( k \in K \)
Minimizing Beam-on Time: Example

Example

\[ \sum_{k \in K} c_k = 5 \]

\[
\begin{pmatrix}
2 & 5 & 3 \\
3 & 4 & 2
\end{pmatrix}
= 2 \begin{pmatrix}
1 & 1 & 1 \\
1 & 1 & 1
\end{pmatrix} + 1 \begin{pmatrix}
0 & 1 & 1 \\
1 & 0 & 0
\end{pmatrix} + 2 \begin{pmatrix}
0 & 1 & 0 \\
0 & 1 & 0
\end{pmatrix}
\]

- \( S_1, S_2, S_3 \) have positive monitor units
- \( \alpha_k = 0 \) for all other binary matrices \( k \in K \)
Minimizing Beam-on Time: Solution Approach

- Fluence map

\[ X = \begin{pmatrix} 2 & 5 & 3 \\ 3 & 4 & 2 \end{pmatrix} \]

- Difference matrix

\[ \tilde{X} = \begin{bmatrix} X_{m,n} - X_{m,n-1} \end{bmatrix}_{M \times (N+1)} = \begin{pmatrix} 2 & 3 & -2 & -3 \\ 3 & 1 & -2 & -2 \end{pmatrix} \]

- Whenever \( \tilde{X}_{m,n} > 0 \), decomposition needs to use interval with left boundary in bixel \( m \) with at least \( \tilde{X}_{m,n} \) MU.
Minimizing Beam-on Time: Solution Approach

- Sum of positive gradient (SPG)

\[
SPG_m = \sum_{n=1}^{N+1} \max \left\{ \tilde{X}_{m,n}, 0 \right\} = \sum_{n=0}^{N} \max \left\{ 0, X_{m,n+1} - X_{m,n} \right\}
\]

\[
SPG(X) = \max_m \{ SPG_m \} = \max \{ 4, 5 \}
\]

- \( SPG(X) \) provides a lower bound on the required beam-on time for delivering \( X \)
  - there are decompositions for which LB can be obtained
Minimizing Beam-on Time: Solution Approach

- One can decompose each fluence row individually

\[ X = \begin{pmatrix} 2 & 5 & 3 \\ 3 & 4 & 2 \end{pmatrix} \]

\[ X_1. = \begin{pmatrix} 2 & 5 & 3 \end{pmatrix} \rightarrow \sum_{k \in K} \alpha_{1k} V_k \]

\[ X_2. = \begin{pmatrix} 3 & 4 & 2 \end{pmatrix} \rightarrow \sum_{k \in K} \alpha_{2k} V_k \]

- Row decompositions can be combined to form LS with minimal beam-on time

\[ \alpha_{k''} S_{k''} = \min \{ \alpha_{1k}, \alpha_{2k'} \} \begin{pmatrix} V_k \\ V_{k'} \end{pmatrix} \]
Minimizing Beam-on Time: Network-flow Model

- Decomposing fluence row m: \((X_{mn} : n \in N)^\top\)
  - see [Ahuja and Hamacher, 2005]
- Let \(K\) be the collection of all binary vectors with consecutive-ones property

\[
\min \sum_{k \in K} \alpha_k
\]

subject to

\[
\begin{pmatrix}
1 & 0 & 0 & 1 & \ldots & 1 \\
0 & 1 & 0 & 1 & \ldots & 1 \\
0 & 0 & 1 & 0 & \ldots & 1
\end{pmatrix}
\begin{pmatrix}
\alpha_1 \\
\alpha_2 \\
\vdots \\
\alpha_K
\end{pmatrix}
= 
\begin{pmatrix}
X_{m1} \\
X_{m2} \\
\vdots \\
X_{mN}
\end{pmatrix}
\]

\(\alpha_k \geq 0 \quad \forall k \in K\)
Minimizing Beam-on Time: Network-flow Model

- It can be represented using a *network-flow* model
- by adding a zero vector at the end of constraint matrix and subtracting row $n$ from $n+1$

\[
\begin{pmatrix}
1 & 0 & 0 & \ldots & 1 \\
-1 & 1 & 0 & \ldots & 0 \\
0 & -1 & 1 & \ldots & 0 \\
0 & 0 & -1 & \ldots & -1 \\
\end{pmatrix}
\begin{pmatrix}
\alpha_1 \\
\alpha_2 \\
\vdots \\
\alpha_K \\
\end{pmatrix}
= 
\begin{pmatrix}
X_{m1} \\
X_{m2} - X_{m1} \\
\vdots \\
X_{mN} - X_{mN-1} \\
-X_{mN} \\
\end{pmatrix}
\]
Minimizing Beam-on Time: Network-flow Model

- Network representation
  - nodes are bixels \( n = 1, \ldots, N + 1 \)
  - arcs are binary vectors with consecutive ones \( k \in K \)
  - nodes have supply/demand

- What is the minimum-cost flow to satisfy all node demands?

\[
X_{m1} \quad X_{m2} - X_{m1} \quad X_{m3} - X_{m2} \quad \ldots \quad X_{mN} - X_{mN-1} - X_{mN}
\]
Network-flow Model: Min-cost Flow Algorithm

- Surplus/demand for node $n = 1, \ldots, N$ is defined as
  
  $$b(n) = X_{mn} - X_{mn-1}$$

- Flow is sent from nodes with surplus $b(n) > 0$ to nodes with demand $b(n') < 0$

1. Initialize: $u_0 = \min \{n : b_n > 0\}$, $v_0 = \min \{n : b_n < 0\}$
2. Main step: at iteration $k$
   
   $$\alpha(u_k, v_k) = \min \{b(u_k), -b(v_k)\}$$
   
   $$b(u_k) \leftarrow b(u_k) - \alpha(u_k, v_k)$$
   
   $$b(v_k) \leftarrow b(v_k) + \alpha(u_k, v_k)$$

3. Termination condition: if $u_k, v_k = N + 1$, stop; else, increment accordingly
Minimizing Number of Apertures

- LS with objective of minimizing number of apertures is more involved
  - belongs to the class of NP-hard problems (see [Baatar et al., 2005])
    - in contrast with minimizing beam-on time which can be solved in polynomial time
  - fluence rows cannot be decomposed individually

- Solution approaches seek for decompositions using minimal number of apertures while constraining beam-on time to SPG
LS Formulation: Minimizing Number of Apertures

- Minimizing number of apertures
  \[
  \min \| \alpha \|_0
  \]
  subject to
  \[
  X = \sum_{k \in K} \alpha_k S_k
  \]
  \[
  \sum_{k \in K} \alpha_k \leq SPG(X)
  \]
  \[
  \alpha_k \geq 0 \quad k \in K
  \]

- Notation
  - \( K \): set of \( M \times N \) binary matrices with consecutive-ones property at each row
  - \( S_k \): binary matrix \( k \in K \)
  - \( \alpha_k \): number of MU for binary matrix \( k \in K \)
Minimizing Number of Apertures: Heuristics

1 Initialize: $\hat{X}_0 = X$

2 Main step: at iteration $k$ find $\alpha_k > 0$ and a binary matrix $S_k$ such that

$$\hat{X}_k = \hat{X}_{k-1} - \alpha_k S_k \geq 0$$

$$SPG(\hat{X}_k) = SPG(\hat{X}_{k-1}) - \alpha_k$$

3 Termination condition: if $\hat{X}_k = 0$ stop; else, go to Step 2

To minimize number of apertures we choose maximum possible $\alpha_k$ and corresponding $S_k$ (see [Engel, 2005])
Minimizing Number of Apertures: Heuristics

1. Initialize: \( \hat{X}_0 = X \)

2. Main step: at iteration \( k \) find \( \alpha_k > 0 \) and a binary matrix \( S_k \) such that

\[
\hat{X}_k = \hat{X}_{k-1} - \alpha_k S_k \geq 0
\]

\[
SPG \left( \hat{X}_k \right) = SPG \left( \hat{X}_{k-1} \right) - \alpha_k
\]

3. Termination condition: if \( \hat{X}_k = 0 \) stop; else, go to Step 2

- To minimize number of apertures we choose maximum possible \( \alpha_k \) and corresponding \( S_k \) (see [Engel, 2005])
Shortcoming of the Sequential Method

- There is often dose discrepancy between FMO and LS solutions
  - some LS methods require rounding fluence maps
  - limited number of apertures are used

- Knowledge of *shape* and *intensity* of apertures are required to model several aspects of IMRT treatment plan

- DAO frameworks have been developed to address these issues
Shortcoming of the Sequential Method

- There is often dose discrepancy between FMO and LS solutions
  - some LS methods require rounding fluence maps
  - limited number of apertures are used
- Knowledge of shape and intensity of apertures are required to model several aspects of IMRT treatment plan
- DAO frameworks have been developed to address these issues
Direct aperture optimization (DAO) aims at directly finding optimal collection of apertures and their intensities.

- FMO and LS are integrated into a single problem.

- In contrast with 3D-conformal radiotherapy where apertures conform to tumor shape in beam’s eye-view, in DAO any deliverable aperture by MLC may be used.

- We discuss DAO solution methods proposed in [Romeijn et al., 2005] and [Hårdemark et al., 2003].
Direct Aperture Optimization

- Direct aperture optimization (DAO) aims at *directly* finding optimal collection of apertures and their intensities
  - FMO and LS are integrated into a single problem
- In contrast with 3D-conformal radiotherapy where apertures conform to tumor shape in beam’s eye-view, in DAO any deliverable aperture by MLC may be used
- We discuss DAO solution methods proposed in [Romeijn et al., 2005] and [Hårdemark et al., 2003]
Direct aperture optimization (DAO) aims at directly finding optimal collection of apertures and their intensities.

- FMO and LS are integrated into a single problem.

In contrast with 3D-conformal radiotherapy where apertures conform to tumor shape in beam’s eye-view, in DAO any deliverable aperture by MLC may be used.

We discuss DAO solution methods proposed in [Romeijn et al., 2005] and [Hårdemark et al., 2003].
Column Generation Method: Formulation

- Mathematical formulation for the DAO problem

$$\min G(d)$$

subject to

$$d_v = \sum_{k \in K} D_{kv} y_k \quad v \in V$$

$$y_k \geq 0 \quad k \in K$$

- Notation

- $$\mathbf{y} = (y_k : k \in K)^\top$$: vector of aperture intensities
- $$\mathbf{D} = [D_{kv}]$$: matrix of aperture dose deposition coefficients
- $$\mathbf{d} = (d_v : v \in V)^\top$$: vector of dose distribution
Naive application of convex optimization techniques to this problem is computationally prohibitive.

- $K$ contains a large number of apertures
  - $\mathcal{O}(10^{45})$ deliverable apertures per beam angle

- We are interested in sparse solutions
  - clinically reasonable number of apertures ($\leq 50$ per beam)
DAO Solution Method: Search for Local Minimum

- Necessary optimality conditions for unconstrained problems (i.e., $\nabla F (\tilde{y}) = 0$) can be extended to constrained ones.
- If $\tilde{y}$ is a local minimum, then, under some regularity conditions, it satisfies Karush-Kuhn-Tucker (KKT) conditions.
- For a convex problem with affine equality constraints, KKT conditions are necessary and sufficient for global optimality.
KKT Optimality Conditions

- If \( \bar{y} \) is a local minimum, then there exist vectors of Lagrange multipliers \( \bar{u}, \bar{v} \) such that this system of equations are satisfied

\[
\begin{align*}
\nabla F(\bar{y}) + \sum_{\ell \in L_1} \bar{u}_\ell \nabla P_\ell (\bar{y}) + \sum_{\ell \in L_2} \bar{v}_\ell \nabla Q_\ell (\bar{y}) &= 0 \\
\end{align*}
\]

\[
\begin{align*}
\min F(y) \\
s.t. \\
P_\ell (y) &\leq 0 \quad \ell \in L_1 \\
Q_\ell (y) &= 0 \quad \ell \in L_2 \\
\bar{u}_\ell P_\ell (\bar{y}) &= 0 \quad \ell \in L_1 \\
P_\ell (\bar{y}) &\leq 0 \quad \ell \in L_1 \\
Q_\ell (\bar{y}) &= 0 \quad \ell \in L_2 \\
\bar{u}_\ell &\geq 0 \quad \ell \in L_1
\end{align*}
\]
KKT Conditions for DAO

KKT conditions for the DAO problem are as follows:

$$\min G(d)$$

subject to

$$d_v = \sum_{k \in K} D_{kv} y_k \quad v \in V$$

$$y_k \geq 0 \quad k \in K$$

$$\pi_v \geq 0 \quad v \in V$$

$$\rho_k \geq 0 \quad k \in K$$

$$d_v = \sum_{k \in K} D_{kv} y_k \quad v \in V$$

$$\sum_{v \in V} D_{kv} \pi_v - \rho_k = 0 \quad k \in K$$

$$y_k \rho_k = 0 \quad k \in K$$

$$\pi_v, \rho: \text{vectors of voxels and apertures Lagrange multipliers}$$
We aim at finding $(\bar{y}, \bar{d}, \bar{\pi}, \bar{\rho})$ that satisfy KKT conditions.

Due to large number of apertures we cannot incorporate all of them.

We start by considering only a subset of apertures $\hat{K} \subset K$ and sequentially add remaining ones until KKT conditions are all met.
Consider restricted DAO problem in which $\hat{K} \subset K$

$$
\text{min } G(d)
$$

subject to

$$
d_v = \sum_{k \in \hat{K}} D_{kv} y_k \\
y_k \geq 0 \\
v \in V \\
k \in \hat{K}
$$

This can be solved using a constrained optimization method to obtain $(y^*, d^*)$

- barrier method or projected gradient method
Restricted DAO Problem: KKT Conditions

- Solution \((y^*, d^*)\) satisfies KKT conditions for the restricted DAO problem

\[
d^*_v = \sum_{k \in \hat{K}} D_{kv} y^*_k \quad v \in V
\]

\[
\pi^*_v = \left. \frac{\partial G}{\partial d_v} \right|_{d=d^*} \quad v \in V
\]

\[
\rho^*_k = \sum_{v \in V} D_{kv} \pi^*_v \quad k \in \hat{K}
\]

\[
y^*_k \rho^*_k = 0 \quad k \in \hat{K}
\]

\[
y^*_k \geq 0, \rho^*_k \geq 0 \quad k \in \hat{K}
\]

- We then construct solution \(\bar{y}\) as

\[
\bar{y}_k = \begin{cases} 
y^*_k & k \in \hat{K} \\
0 & k \in K \setminus \hat{K} \end{cases}
\]
DAO Solution Approach: KKT Conditions

- We substitute $\tilde{y}$ in KKT conditions of original DAO problem

$$
\tilde{d}_v = \sum_{k \in K} D_{kv} \tilde{y}_k \quad v \in V
$$

$$
\tilde{\pi}_v = \left. \frac{\partial G}{\partial d_v} \right|_{d=\tilde{d}} \quad v \in V
$$

$$
\tilde{\rho}_k = \sum_{v \in V} D_{kv} \tilde{\pi}_v \quad k \in K
$$

$$
\tilde{y}_k \tilde{\rho}_k = 0 \quad k \in K \quad \text{why?}
$$

$$
\tilde{y}_k \geq 0 \quad k \in \hat{K}
$$

$$
\tilde{y}_k = 0 \quad k \in K \setminus \hat{K}
$$

$$
\tilde{\rho}_k \geq 0 \quad k \in \hat{K} \quad \text{why?}
$$

$$
\tilde{\rho}_k \geq 0 \quad k \in K \setminus \hat{K}
$$
DAO Solution Approach: Pricing Problem

To ensure if $\bar{\rho}_k \geq 0$ for $k \in K$ we formulate and solve the *pricing problem*

$$\min_{k \in K} \bar{\rho}_k = \sum_{v \in V} D_{kv} \bar{\pi}_v$$

Aperture $k \in K$ consists of a collection, $A_k$, of exposed beamlets $i \in I$

$$D_{kv} \approx \sum_{i \in A_k} D_{iv}$$

Pricing problem is reformulated using beamlet dose deposition coefficients

$$\min_{k \in K} \sum_{i \in A_k} \sum_{v \in V} D_{iv} \bar{\pi}_v$$
DAO Solution Approach: Pricing Problem

- To ensure if $\bar{\rho}_k \geq 0$ for $k \in K$ we formulate and solve the **pricing problem**

$$\min_{k \in K} \bar{\rho}_k = \sum_{v \in V} D_{kv} \bar{\pi}_v$$

- Aperture $k \in K$ consists of a collection, $A_k$, of exposed beamlets $i \in I$

$$D_{kv} \approx \sum_{i \in A_k} D_{iv}$$

- Pricing problem is reformulated using beamlet dose deposition coefficients

$$\min_{k \in K} \sum_{i \in A_k} \sum_{v \in V} D_{iv} \bar{\pi}_v$$
DAO Solution Approach: Pricing Problem

- To ensure if $\bar{\rho}_k \geq 0$ for $k \in K$ we formulate and solve the *pricing problem*

  $$\min_{k \in K} \bar{\rho}_k = \sum_{v \in V} D_{kv} \bar{\pi}_v$$

- Aperture $k \in K$ consists of a collection, $A_k$, of exposed beamlets $i \in I$

  $$D_{kv} \approx \sum_{i \in A_k} D_{iv}$$

- Pricing problem is reformulated using beamlet dose deposition coefficients

  $$\min_{k \in K} \sum_{i \in A_k} \sum_{v \in V} D_{iv} \bar{\pi}_v$$
DAO Solution Approach: Pricing Problem

It finds aperture \( k \) with most negative Lagrange multiplier at \( \bar{y} \)

- \( \bar{\rho}_k \): rate of change in \( G \) as intensity of aperture \( k \) increases (reduced gradient)

\[
\min_{k \in K} \bar{\rho}_k = \sum_{i \in A_k} \sum_{v \in V} D_{iv} \bar{\pi}_v
\]

- given reduced gradient of all beamlets, it finds collection of beamlets that
  - forms a deliverable aperture
  - has most negative cumulative reduced gradient

```plaintext
<table>
<thead>
<tr>
<th></th>
<th>-1</th>
<th>-2</th>
<th>-3</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>3</td>
<td>-1</td>
<td>-3</td>
<td>2</td>
</tr>
<tr>
<td>-1</td>
<td>2</td>
<td>-5</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>
```
Pricing Problem: Pricing Problem

- Pricing problem can be solved for individual beam angles $b \in B$
  \[
  \min_{k \in K_b} \bar{\rho}_k
  \]

- Pricing problem can be separated over beamlet rows
  - finding a *sub sequence* of beamlets with most negative cumulative reduced gradient

| 2 | -1 | 3 | -2 | 1 | -2 | 2 | -1 |
Pricing Problem: Minimum Subsequent Sum

- It can be solved by a single pass over bealmests in a row \( i = 1, \ldots, N \)

1. Initialize: \( b_i = \sum_{v \in V} D_{iv} \pi_v \) for \( i \in I \), \( \text{minSoFar} = 0 \), \( \text{minEndingHere} = 0 \)

2. Main step: For \( i = 1 : N \)

   \[
   \text{minEndingHere} = \min \left\{ 0, \text{minEndingHere} + b_i \right\}
   \]

   \[
   \text{minSoFar} = \min \left\{ \text{minEndingHere}, \text{minSoFar} \right\}
   \]

3. Output: \( \text{minSoFar} \)

- The pricing problem can be extended to incorporate additional MLC hardware constraints
Column Generation Method

- Column generation method for DAO solves restricted and pricing problems iteratively
  - common approach to solve large-scale optimization problems
- It can be terminated if no promising aperture exists or if current solution is satisfactory
Leaf Refinement Problem

- Given a fixed number of apertures, the *leaf refinement problem* aims at finding their optimal leaf positions as well as their intensities
  - see [Hårdemark et al., 2003, Cassioli and Unkelbach, 2013]

- Major difference from column generation approach is that the number of apertures is fixed
Leaf Refinement Method: Dose Deposition

Expressing dose deposited in voxel $v \in V$ in terms of aperture intensities and leaf positions

$$d_v \left( x^{(l)}, x^{(r)}, y \right) \equiv \sum_{k \in K} y_k \sum_{m \in M} \left( \phi^v_{mb(k)} \left( x^{(r)}_{mk} \right) - \phi^v_{mb(k)} \left( x^{(l)}_{mk} \right) \right)$$

Notation

- $y = \left( y_k : k \in K \right)^\top$: vector of aperture intensities
- $x^{(l)} = \left( x^{(l)}_{mk} : m \in M, k \in K \right)^\top$: vector of left leaf positions
- $x^{(r)} = \left( x^{(r)}_{mk} : m \in M, k \in K \right)^\top$: vector of right leaf positions
- $\phi^v_{mb} (x)$: dose deposited in voxel $v \in V$ under unit intensity from row $m \in M$ in beam angle $b \in B$ when interval $[0, x]$ is exposed
Leaf Refinement Method: Dose Deposition

Given beamlet dose deposition coefficient matrix \([D_{iv}]\), \(\phi_{mb}^v\) can be approximated using a piecewise-linear function.
Leaf Refinement Problem: Formulation

- **Mathematical formulation**

  \[
  \min G(d(z))
  \]

  subject to

  \[
  Ax \leq 0 \\
  H(d(x, y)) \leq 0 \quad \rightarrow \quad F_\ell(d(z)) \leq 0 \quad \ell \in L \\
  y \geq 0
  \]

- **Notation**
  - \(d = (d_v : v \in V)^\top\): vector of dose distribution
  - \(y = (y_k : k \in K)^\top\): vector of aperture intensities
  - \(z = (x^{(l)} x^{(r)} y)^\top\): vector of all variables
  - \(A\): constraint matrix of leaf positions
Leaf Refinement Problem: Solution Approach

- *Sequential quadratic programming* (SQP) can be used
- SQP aims at finding a solution that satisfies KKT conditions

\[
\nabla G(z) + \sum_{\ell \in L} u_\ell \nabla F_\ell(z) = 0
\]

\[
\begin{align*}
    u_\ell F_\ell(z) &= 0 \quad \ell \in L \\
    u &\geq 0
\end{align*}
\]

- One can use Newton method to solve this system
  - Lagrangian is defined as \( \mathcal{L} \equiv G + \sum_{\ell \in L} u_\ell F_\ell \)
  - we let \( \nabla F^T = \begin{pmatrix} \nabla F_1^T \\ \vdots \\ \nabla F_L^T \end{pmatrix} \)
Sequential Quadratic Programming

- Newton method at iteration \( k \) requires to solve

\[
\begin{pmatrix}
\nabla^2 \mathcal{L}(k) & \nabla F^T(k) \\
u_1 \nabla F^T_{1(k)} & F_{1(k)} \ 0 \ \cdots \ 0 \\
u_2 \nabla F^T_{2(k)} & 0 \ F_{2(k)} \ \cdots \ 0 \\
u_L \nabla F^T_{L(k)} & 0 \ 0 \ \cdots \ F_{L(k)} \\
\end{pmatrix}
\begin{pmatrix}
z - z(k) \\
u - u(k) \\
\end{pmatrix}
= 
\begin{pmatrix}
\nabla G(k) + \sum_{\ell \in L} u_{\ell(k)} \nabla F_{\ell(k)} \\
u_{1(k)} F_{1(k)} \\
\vdots \\
u_{L(k)} F_{L(k)} \\
\end{pmatrix}
\]

this reduces to

\[
\nabla^2 \mathcal{L}(k) (z - z(k)) + \nabla F^T(k) u = -\nabla G(k)
\]

\[
u_\ell \left( \nabla F^T_{\ell(k)} (z - z(k)) + F_{\ell(k)} \right) = 0 \quad \ell \in L
\]
Sequential Quadratic Programming

Along with $\mathbf{u} \geq \mathbf{0}$ these are also KKT conditions for the following quadratic programming (QP) problem

$$\min G_{(k)} + \nabla G_{(k)}^\top \mathbf{v} + \frac{1}{2} \mathbf{v}^\top \nabla^2 \mathcal{L}_{(k)} \mathbf{v}$$

subject to

$$\nabla F_{\ell(k)}^\top \mathbf{v} + F_{\ell(k)} \leq 0 \quad \ell \in L$$

in which we substituted $\mathbf{v} = \mathbf{z} - \mathbf{z}_{(k)}$
Sequential Quadratic Programming

- One can alternatively solve this QP problem at iteration $k$ of Newton method
  - objective function is quadratic approximation of $G$ plus curvature of constraints at $z = z_k$
  - constraints are linear approximation of $F_\ell (\ell \in L)$ at $z = z_k$
- The QP problem requires computing $\nabla^2 \mathcal{L}_(k)$
  - computationally expensive
  - may not be positive definite
Sequential Quadratic Programming: BFGS Update

- To overcome this issue quasi-newton method is employed

\[ \nabla^2 \mathcal{L}(k) \approx B(k) \succeq 0 \]

- Positive definite approximations of Hessian using (Broyden-Fletcher-Goldfarb-Shanno) BFGS update

\[
B_{(k+1)} = B_{(k)} + \frac{q_{(k)}q_{(k)}^\top}{q_{(k)}^\top p_{(k)}} - \frac{B_{(k)}p_{(k)}p_{(k)}^\top B_{(k)}}{p_{(k)}^\top B_{(k)}p_{(k)}} \\
p_k = z_{(k+1)} - z_{(k)} \quad q_k = \nabla \mathcal{L}_{(k+1)} - \nabla \mathcal{L}_{(k)}
\]
1 Initialization: select initial variables $z(0)$, lagrange multipliers $u(0)$, and Hessian p.d. approximation $B(0)$

2 Main Step: at iteration $k$ solve the QP problem

$$\min_v G(k) + \nabla G^\top(k)v + \frac{1}{2}v^\top B(k)v$$

subject to

$$F_{\ell(k)} + \nabla F_{\ell(k)}^\top v \leq 0 \quad \ell \in L$$

3 Termination condition: if $\|v^*\| < \epsilon$, stop; otherwise, $z(k+1) = z(k) + v^*$, update $B(k+1)$ using BFGS method and go to Step 2
DAO aims at directly solving for aperture shape and intensities.

DAO plans employ fewer apertures and shorter beam-on times compared to two-stage method to obtain similar dose conformity.

- see [Ludlum and Xia, 2008, Men et al., 2007]

We discussed two major DAO approaches:
- column generation method
- leaf refinement method
References I


Aperture shape optimization for IMRT treatment planning.

A new algorithm for optimal multileaf collimator field segmentation.

Direct machine parameter optimization with raymachine® in pinnacle3®.
*RaySearch White Paper. Stockholm, Sweden: RaySearch Laboratories AB.*
Comparison of IMRT planning with two-step and one-step optimization: a way to simplify IMRT.

An exact approach to direct aperture optimization in IMRT treatment planning.

A column generation approach to radiation therapy treatment planning using aperture modulation.

*Intensity-modulated radiation therapy.*

Institute of Physics Publishing.