Minimizing Beam-On Time in Radiation Treatment using Network Flows

Ravindra K. Ahuja
Department of Industrial & Systems Engineering,
University of Florida, Gainesville, FL 32611, USA
ahuja@ufl.edu

Horst W. Hamacher
Fachbereich Mathematik
Universitaet Kaiserslautern
Kurt Schumacher Str. 26
D-67653 Kaiserslautern,
Germany
hamacher@mathematik.uni-kl.de

(May, 2002)

Minimizing Beam-On Time in Radiation Treatment using Network Flows

Ravindra K. Ahuja[1] and Horst W. Hamacher[2]

Abstract

In this paper, we study the modulation of intensity matrices arising in cancer radiation therapy using multileaf collimators. This problem can be formulated as decomposing a given m×n integer matrix into a positive linear combination of (0,1) matrices with the strict consecutive 1's property in rows. We consider a special case when the rows of the intensity matrix can be independently decomposed, in which case the problem is equivalent to m independent problems of decomposing an intensity row into a positive linear combination of (0, 1) rows with the consecutive 1's property. We show that this problem can be transformed into a minimum cost flow problem in a directed network which has the following special structures: (i) the network is a complete acyclic graph (that is, there is an arc (i, j) whenever i < j); (ii) each arc cost is 1; and (iii) each arc is uncapacitated (that is, has infinite capacity). We show that using this special structure, the minimum cost flow problem can be solved in O(n) time. Since we need to solve m such problems, the total running time of our algorithm is O(nm) time, which is an optimal algorithm to decompose a given m×n integer matrix into a positive linear combination of (0,1) matrices.

[1] Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611
[2] Fachbereich Mathematik, Universitaet Kaiserslautern, D-67653 Kaiserslautern, Germany

1. Introduction

In this paper, we study the modulation of intensity matrix arising in cancer radiation therapy using multileaf collimators. Using other techniques, we will determine gantry angles and the intensity function at each gantry angles. The radiation head at any gantry angle is assumed to be a rectangle which is partitioned into mxn equidistant cells, called *bixels*. At each stop of the gantry, the intensity function can be represented as a two-dimensional mxn array I representing the amount of time uniform radiation needs to be sent off in each bixel in the gantry. We assume that I is an integer valued matrix. For example, if we choose to discretize the beam head into a 5x6 grid, then one possible intensity matrix is:

$$I = \{I_{ij}\} = \begin{bmatrix} 4 & 4 & 3 & 0 \\ 1 & 6 & 3 & 0 \\ 3 & 4 & 1 & 0 \\ 4 & 4 & 3 & 0 \\ 3 & 6 & 4 & 3 \end{bmatrix} \tag{1}$$

In order to generate I, we can use multileaf collimator. Each row of I, called a *channel*, has an associated pair of leaves – a *left leaf* and a *right leaf* and the radiation can pass in between left and right leaves. If I has n columns 1, 2, ..., n, then for each row i, there are n+1 positions, 1, 2, … , n, n+1, at which the left and right leaves can be positioned. If the left leaf is at position k and the right leaf is at position *l*, then the radiation will pass through the bixels numbered k, k+1, …, *l*-1. (See, for example, Figure 1.)

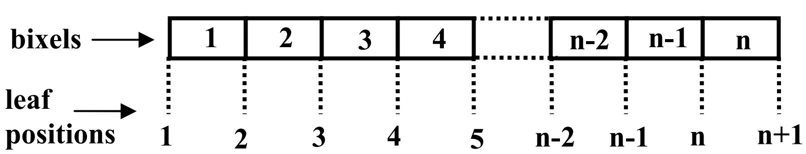

Figure 1: Each channel (row) of the multileaf collimeter.

Each choice of the left and right leaves in all rows in characterized by a 0-1 matrix; this matrix is called a *shape matrix*. If S_1, S_2, ... S_K are shape matrices and x_1, x_2, ..., x_k is the time the linear accelerator is opened to release the radiation for the corresponding shape matrix, respectively, then an intensity of $\sum_{k=1}^{K} S_k x_k$ is released. For example, if I is given by (1), then I = $3S_1 + 1S_2 + 2\,S_3$, where

$$S_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \quad S_2 = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix} \quad S_3 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \tag{2}$$

We want to select the shape matrices S_k's and the times x_k's are so that the beam-on time given by $\sum_{k=1}^{K} x_k$ plus the time needed to setup the leaves is minimum. We can define three optimization problems for the intensity modulation.

1. **Minimize beam-on time**. This problem minimizes the time during which radiation is released and ignores the setup time needed to go from one shape matrix to another shape matrix. This problem can be formulated as the following mathematical program:

$$\text{Minimize } z^* = \sum_{k=1}^{K} x_k \tag{3a}$$

subject to

$$\sum_{k=1}^{K} S_k x_k = I \tag{3b}$$

$$x_k \geq 0 \quad \text{for all } k = 1, 2, \ldots, K. \tag{3c}$$

2. **Minimize the sum of beam-on time and constant set-up time**: In this problem, we minimize the sum of the beam-on time and the set-up time, assuming that the set-up time needed to go from one shape matrix to another shape matrix is constant. This problem can be formulated as the following mathematical program:

$$\text{Minimize } z^* = \sum_{k=1}^{K} x_k \tag{4a}$$

subject to (3b) and (3c). (4b)

3. **Minimize the sum of beam-on time and variable set-up time**: This problem is a generalization of the problem in (4) where we do not assume the set-up time to be constant but allow it to depend upon the shape matrices. This problem can be formulated as:

$$\text{Minimize } z^* = \sum_{k=1}^{K} x_k + \sum_{k=1}^{K-1} c(S_k, S_{k+1}) \tag{5a}$$

subject to (3b) and (3c), (5b)

where $c(S_k, S_{k+1})$ is the time it takes to go from the shape matrix S_k to the shape matrix S_{k+1}.

The above three problems are well researched problems. It can be shown that Problems (2) and (3) are NP-complete (Orlin [2002]); whereas the Problem (1) is polynomially solvable (Boland et al. [2002]). Due to the difficulty of the Problems (2) and (3), most of the proposed algorithms are heuristic algorithms. Some selected references devoted to this problem are: Bortfeld et al. [1994], Dai and Hu [1999], Evans et al. [1997], Galvin et al. [1993], Que [1999], Siochi [1999], Webb [1998], Wu and Zhu [2001], Xia and Verhey [1998], Yu et al. [1995a, 19995b]. The paper by Boland [2002] presents a brief survey of literature devoted to the study of these problems and also describes a polynomial time algorithm to solve Problem (1).

Our paper is motivated by the research done by Boland et al. [2002]. We consider a special case when the rows of the intensity matrix can be independently decomposed, in which case the problem is equivalent to m independent problems of decomposing an intensity row into a positive linear combination of (0, 1) rows with the consecutive 1's property. We show that this problem can be transformed into a minimum cost flow problem in a directed network which has the following special structures (i) the network is a complete acyclic graph (that is, there is an arc (i, j) whenever $i < j$); (ii) each arc cost is 1; and (iii) each arc is uncapacitated (that is, has infinite capacity). We show that using this special structure, the minimum cost flow problem can be solved in O(n) time. Since we need to solve m such problems, the total running time of our algorithm is O(nm) time which is an optimal algorithm to decompose a given $m \times n$ integer matrix into a positive linear combination of (0,1) matrices.

2. A Special Case

Some radiation therapy machines require that the shape matrix must satisfy *interleaf motion constraints*. Interleaf motion constraints state that the left leaf in one row should not be to the right of the right leaf in an adjacent row and vice versa; otherwise crashes between the two leaves will occur. Hence, adjacent rows in a shape matrix must satisfy some additional constraints in order to be a valid shape matrix. However, the more recent and modern radiation therapy machines do not require that the shape matrices to satisfy the interleaf motion constraints. We will henceforth assume that these are no interleaf motion constraints and will develop methods to solve the minimum beam-on time problem (2) in the absence of such constraints.

In the absence of interleaf motion constraints each row of the shape matrix can be determined independently of other rows. Hence, the problem of determining shape matrices of size mxn can be decomposed into m independent problems of determining *shape rows* of size 1xn. We can determine the minimum beam-on time for each row in the intensity matrix I independent of other rows, and then use the row solutions to construct the minimum beam-on time solution for the intensity matrix. The minimum beam-on time problem for the i^{th} row of the intensity matrix can be stated as:

$$\text{Minimize } z_i^0 = \sum_{k=1}^{K'} x_{ik} \tag{6a}$$

subject to

$$\sum_{k=1}^{K'} R_k x_{ik} = b_i \tag{6a}$$

$$x_{ik} \geq 0 \quad \text{for all } k = 1, 2, \ldots, K'. \tag{6c}$$

where b_i denotes the i^{th} row of the intensity matrix I; R_k, $1 \leq k \leq K'$ denote the set of all possible shape row vectors. In this formulation, x_{ik}'s are decision variables and determines the time for which the shape row R_k can be exposed for radiation. The following lemma relates the optimal solution of (6) with the optimal solution of (3):

Lemma 1: *Let $z_{max} = max\{ z_i^0 : 1 \leq i \leq m\}$. Then, $z^* = z_{max}$ is an optimal solution of (3).*

Proof: It is easy to see that each z_i^0 is a lower bound on the z*, the optimal objective function value of (3). Hence, z_{max} is a valid lower bound an z*. It can also be shown that we can construct a sequence of shape matrices for which the beam-on time will equal z_{max}; hence, z_{max} is a valid upper bound on z*. It follows that $z_{max} = z^*$. ♦

We will henceforth focus on solving (6). For the sake of simplicity, we first eliminate the subscript i (representing the i^{th} row). We will thus refer to x_{ik} by x_k and b_i by b. We also convert the row vector R_k's and b into column vectors by taking their transpose and, for simplicity, use the same notation to represent the corresponding column vectors.

Each (column) vector R_k is a 0-1 vector and corresponds to a feasible (non-zero) exposure provided by a pair of left and right leaves. The feasibility of the exposure dictates that all the 1's in each R_k consists of a (possibly null) sequence of 0's, followed by a sequence of 1's, followed by another (possibility null) sequence of 0's. We will henceforth refer to the vector R_k by R_{uv} if in the vector R_k row u is the least index row with element 1 and v is the highest index row with element 1. Let R_{uv}, $(u, v) \in A$ denote the set of all possible column vectors. Since each column vector corresponds to a feasible pair of left and right leaves, the left leaf can take the positions u = 1, 2, 3, …, n+1, and the right leaf can take the position v = u+1, u+2, …, n+1, it follows that $A = \{(u, v): 1 \leq u \leq n+1, u+1 \leq v \leq n+1\}$. Observe that $|A| = n + (n-1) + (n-2) + \ldots + 1 = n(n+1)/2$. We can now restate (6) as:

$$\text{Minimize } z^0 = \sum_{(u,\, v)\in A} x_{uv} \tag{7a}$$

subject to

$$\sum_{(u,v)\in A} R_{uv} x_{uv} = b_i \tag{7b}$$

$$x_{uv} \geq 0, \quad \text{for all } (u, v) \in A \tag{7c}$$

We illustrate (7) using a numerical example. Suppose that n = 4. In this case, A = {(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)} and the formulation (7) becomes:

$$\text{Minimize } x_{12} + x_{13} + x_{14} + x_{23} + x_{24} + x_{34} \tag{8a}$$

subject to

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{12} \\ x_{13} \\ x_{14} \\ x_{23} \\ x_{24} \\ x_{34} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ 0 \end{bmatrix} \tag{8b}$$

$$x_{12}, x_{13}, x_{14}, x_{23}, x_{24}, x_{34} \geq 0,$$

where we have added a zero row (0x = 0) to the constraints (8b). Now observe that each column R_{uv} has consecutive 1's in the rows u, u+1, ... , v. A linear program where each column vectors is a vector of 0's and 1's, and all the 1's are consecutive is called a *linear program* (*LP) with consecutive 1's in columns*. This transformation consists of adding a row of zeros ($n+1^{th}$ row) to the constraint matrix and subtracting each row u from the row (u+1) for each u = n, n-1, .., 1, in this stated order. These row operations give us an equivalent LP where each column R_{uv} has one +1 in the u^{th} row, one -1 in the $(v+1)^{th}$ row, and all other values are zero. Let

$$\text{Minimize } z^0 = \sum_{(u,\, v)\in A} x_{uv} \tag{9a}$$

subject to

$$\sum_{(u,v)\in A} R'_{uv} x_{uv} = b'_i \tag{9b}$$

$$x_{uv} \geq 0, \quad \text{for all } (u, v) \in A \tag{9c}$$

denote the modified linear program. For example, if we apply these row operations to (8), we get the following matrix:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 & 0 \\ 0 & -1 & 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} x_{12} \\ x_{13} \\ x_{14} \\ x_{23} \\ x_{24} \\ x_{34} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 - b_1 \\ b_3 - b_2 \\ -b_3 \end{bmatrix} \tag{10}$$

Next observe that the right-hand side b′ is related to b in the following manner:

Property 1: $b'_1 = b_1$ *and* $b'_i = b_i - b_{i-1}$ *for all* $2 \leq i \leq n+1$.

This property implies that $\sum_{i=1}^{n+1} b'_i = 0$. It is well known that a linear programming problem where each column has one +1 and one -1 and rest of the elements as zero is a minimum cost flow problem. The minimum cost flow problem is one of the fundamental network flow problem and can be solved efficiently (see, for example, Ahja, Magnanti and Orlin [1993]). The minimum cost flow problem (9) is feasible only if $\sum_{i=1}^{n+1} b'_i = 0$. This condition obviously holds for our formulation. Let G = (N, A) denote the underlying network where N = {1, 2, 3, …, n+1}, denotes the node set and A = {(u, v): u = 1, 2, …, n+1, and v = u+1, u+2, …, n+1} denotes the arc set. In the minimum cost flow problem, the cost of flow on each arc equals 1. For example, The LP shown in (9) is a minimum cost flow problem in the network shown in Figure 2, where the number next to each node indicates the supply/demand of the node.

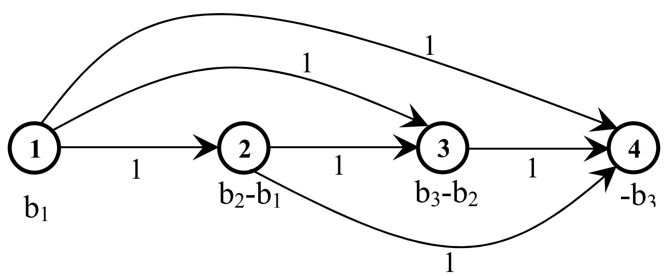

Figure 2. The minimum cost flow formulation of the LP in (10).
The number besides each arc (node) represents its cost (supply/demand).

The minimum cost flow problem (9) has a very special structure and this structure allows us to solve the minimum cost flow problem very efficiently. The minimum cost flow problem (9) has the following properties:

(i) the network is acyclic;
(ii) the network is complete (that is, it contains an arc (i, j) for ever i < j);
(iii) each arc cost is 1; and
(iv) each arc is uncapacitated (that is, it has no capacity restriction).

These properties allow us to solve the minimum cost flow problem in O(n) time using the well known successive shortest path problem for the minimum cost flow problem (see, Ahuja, Magnanti, and Orlin [1993] for a description of this algorithm). The successive shortest path algorithm starts with a zero flow and at each iteration augments flow from a supply node u (for which $b'_u > 0$) to a demand node v (for which $b'_v < 0$) along a shortest path until all node supplies/demands are satisfied. In fact, the algorithm always augments flow along the arc (u, v) (which is a path comprising of a single arc form node to node) where u is a supply node, and v is a demand node, and u < v. There is always such a single arc path because in our network there is an uncapacitated arc from every node u to every node v if u < v. Further, this single arc path (u, v) is a shortest path from node u to node v because this path has cost 1 and a shorter path cannot exist (as each arc cost is 1 and a path must have at least one arc). This result gives us the highly simplified algorithm to solve (9), which we state in Figure 3.

algorithm *min-cost-flow*;
begin
  $e(u) := b'_u$ for all $u \in N$;
  $u := \min\{r : e(r) > 0\}$;
  $v := \min\{r : e(r) < 0\}$;
  while u and v exist **do**
  begin
    $\delta := \min\{e(u), -e(v)\}$;
    $x_{uv} := \delta$;
    $e(u) := e(u) - \delta$;
    $e(v) := e(v) + \delta$;
    if $e(u) = 0$ **then** update u;
    if $e(v) = 0$ **then** update v;
  end;
  return the solution x;
end;

Figure 3. Algorithm for solving the minimum cost problem in (9).

During the execution of the algorithm, we call a node u in G to be an *excess node* if $e(u) > 0$ and a *deficit node* if $e(v) < 0$. Initially, each supply node is an excess node and each demand node is a deficit node. The algorithm always selects node u as the least index excess node and node v as the least excess deficit node. (We need to show that $u < v$.) Since arc (u, v) is present in the network and is uncapacitated, we send $\delta = \min\{e(u), -e(v)\}$ units of flow on the arc (u, v), which is a shortest path from node u to node v. This flow augmentation either reduces e(u) to zero or e(v) to zero. In the former case, we update u and in the later case we update v. To update u (or v) we simplify increment u (or v) by 1 repeatedly, and stop when $e(u) > 0$ (or $e(v) < 0$). The total time taken to update u and v over the entire algorithm is O(n). The other steps of the algorithm also take O(n) time. Hence, the following theorem:

Theorem 1. *The minimum cost flow problem (9) can be solved in O(n) time.*

3. Some Open Problems

These are several open problems that need to be investigated in the future.

Open Problem 1: We have shown that if the network G is a complete acyclic graph, then we can solve the minimum cost flow problem in it in O(n) time. Suppose that G is not a complete acyclic graph. We are given a set of arcs $A' \subset A$, and we want to solve the minimum cost flow problem in $G' = (N, A')$. This problem corresponds to a situation where not all the settings of a pair of left and right leaves are available. We are given only a specified set of leaf settings and we are allowed to use only those settings to obtain the desired column of intensities. These are two open problems:

(a) Does G′ have feasible flow? What is the fastest algorithm to determine a feasible flow if it exists?

(b) If G′ admits a feasible flow, determine a minimum cost flow. What is the fastest algorithm to determine a minimum cost flow?

Open Problem 2: We have assumed in our models so far that all arc costs are 1. This corresponds to the situation when all settings of leaf pairs are equally desirable. Suppose this is not a true and we assign a weight with each pair of leaf settings and we minimize the weighted beam-on time. This results in a variation of the minimum cost flow problem where each arc (u, v) $\in$ A has a cost c_{uv} and we minimize the objective function $\sum_{(u,\, v)\in A} c_{uv} x_{uv}$. These are two open problems:

(a) What is the fastest algorithm to solve the minimum cost flow problem in G = (N, A) when each arc (i, j) $\in$ A has cost c_{ij}?

(b) Suppose that we are given a subset of arcs A′ $\subset$ A. What is fastest algorithm to solve the minimum cost flow problem G' = (N, A′) when each arc (u, v) $\in$ A has a cost c_{uv}?

References:

Ahuja, R.K., T.L. Magnanti, and J.B. Orlin. 1993. *Network Flows: theory, algorithms, and applications*. Prentice-Hall, Englewood Cliffs, New Jersey.

Boland, N., H. W. Hamacher, and F. Lenzen. 2002. Minimizing beam-on time in cancer radiation treatment using multileaf collimators. To appear in *Networks*.

Bortfeld, T.R., D.L. Kahler, T.J. Waldron and A.L. Boyer. 1994. X-ray field compensation with multileaf collimators, *Int. J. Radiation Oncology Biol. Phys.* **28**, 723–730.

Dai, J.-R., and Y.-M. Hu. 1999. Intensity-modulation radiotherapy using independent collimators: An algorithmic study. *Medical Physics* **26**, 2562-2570.

Evans, P.M., V.N. Hansen, W. Swindell. 1997. The optimum intensities for multiple static multileaf collimator field compensation, *Medical Physics* **24**, 1147–1156.

Galvin, J.M., X.G. Chen, R.M. Smith. 1993. Combining multileaf fields to modulate fluence distributions, *Int. J. Radiation Oncology Biol. Phys.* **27**, 697–705.

Lee, E., T. Fox, and I. Crocker. 2000. Optimization of radiosurgery treatment planning via mixed integer programming. *Medical Physics* **27**, 995-1004.

Lenzen, F. 2000. An integer programming approach to the multileaf collimator problem. Master's Thesis, Department of Mathematics, University of Kaiserslautern, Germany.

Orlin, J.B. 2002. Private communications.

Que. W. 1999. Comparison of algorithms for multileaf collimator field segmentation, *Med. Phys.*, **26**, 2390–2396.

Shepard, D.M., M.C. Ferris, G.H. Olivera, and T.R.Mackie. 1999. Optimizing the delivery of radiation therapy to cancer patients. *SIAM Review* 41, 721-744.

Siochi, R.A.C. 1999. Minimizing static intensity modulation delivery time using an intensity solid paradigm, *Int. J. Radiation Oncology Biol. Phys.*, **43**, 671 – 689.

Webb, S. 1998. Configuration options for intensity-modulated radiation therapy using multiple static fields shaped by a multileaf collimator, *Physics in Medicine and Biology* **43**, 214–260.

Wu, X., and Y. Zhu. 2001. An optimization method for importance factors and beam weights based on genetic algorithms for radiotherapy treatment planning. *Physics in Medicine and Biology* **46**, 1085-1099.

Xia, P., L.J. Verhey. 1998. Multileaf collimator leaf sequencing algorithm for intensity modulated beams with multiple static segments, *Medical Physics* **25**, 1424–1434

Yu, C., M. Symons, M. Du, A. Martinez, and J. Wong. 1995a. A method for implementing dynamic photon beam intensity modulation using independent jaws and a multileaf collimator. *Physics in Medicine and Biology* **40**, 769-787.

Yu, C., D. Yan, M. Du, S. Zhou, and L. Verhey. 1995b. Optimization of leaf-positions when shaping a radiation field with a multileaf collimators. *Physics in Medicine and Biology* **40**, 305-308.