The Matrix Orthogonal Decomposition Problem in Intensity-Modulated Radiation Therapy*

Xin Dou[1], Xiaodong Wu[1,2], John E. Bayouth[2], and John M. Buatti[2]

[1] Dept. of Electrical and Computer Engineering, University of Iowa, Iowa City, Iowa 52242, USA
{xdou, xiaodong-wu}@engineering.uiowa.edu

[2] Dept. of Radiation Oncology, University of Iowa, Iowa City, Iowa 52242, USA
{john-bayouth, john-buatti}@uiowa.edu.

Abstract. In this paper, we study an interesting matrix decomposition problem that seeks to decompose a "complicated" matrix into two "simpler" matrices while minimizing the sum of the horizontal complexity of the first sub-matrix and the vertical complexity of the second sub-matrix. The matrix decomposition problem is crucial for improving the "step-and-shoot" delivery efficiency in Intensity-Modulated Radiation Therapy, which aims to deliver a highly conformal radiation dose to a target tumor while sparing the surrounding normal tissues. Our algorithm is based on a non-trivial graph construction scheme, which enables us to formulate the decomposition problem as computing a minimum *s*-*t* cut in a 3-D geometric multi-pillar graph. Experiments on randomly generated intensity map matrices and on clinical data demonstrated the efficiency of our algorithm.

1 Introduction

In this paper, we study an interesting *matrix orthogonal decomposition* problem arising in *intensity-modulated radiation therapy* (IMRT) [15]. IMRT is a modern cancer therapy technique that aims to deliver a highly conformal radiation dose to a target tumor while sparing the surrounding normal tissues. The prescribed dose distribution of radiation is commonly described by an *intensity map* (IM), which is specified by a set of nonnegative integers on a 2-D grid (see Figure 1(a)). The number in a grid cell indicates the amount (in unit) of radiation to be delivered. The delivery is done by a set of cylindrical radiation beams orthogonal to the IM grid.

An advanced tool today for IM delivery is the multileaf collimator (MLC) [15]. An MLC consists of many pairs of tungsten alloy leaves of the same rectangular

* This research was supported in part by a faculty start-up fund from the University of Iowa, and in part by a fund from the American Cancer Society through an Institutional Research Grant to the Holden Comprehensive Cancer Center, the University of Iowa, Iowa City, IA, USA.

D.Z. Chen and D.T. Lee (Eds.): COCOON 2006, LNCS 4112, pp. 156–165, 2006.

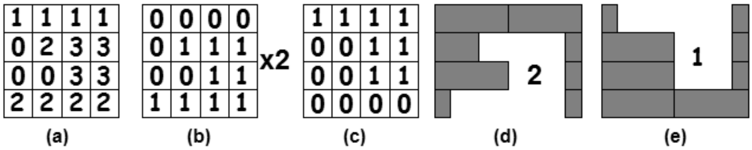

Fig. 1. (a) An example of intensity map. (b) and (c) MLC apertures used to deliver the IM in (a). (d) and (e) The corresponding collimator configurations in (b) and (c).

shape and size. The leaves can move left and right to form a rectilinear region, called an *MLC-aperture*. Each MLC-aperture is associated with an integer representing the radiation units delivered by its radiation beam.

One of the most popular IMRT delivery technique [14] is called the *static leaf sequencing* (SLS) or *step-and-shoot* approach [5, 15, 18] Mathematically, the "step-and-shoot" delivery planning can be viewed as the following matrix decomposition problem: Given an intensity map M (i.e., a matrix), decompose M into the form of $M = \sum_{i=1}^{\kappa} \alpha_i S_i$, where S_i is a special 0-1 matrix specifying an MLC-aperture, α_i is the amount of radiation delivered through S_i, and κ is the number of MLC-apertures used to deliver M (see Figure 1). (The reader is referred to [18, 1, 7, 4] for more details on the step-and-shoot IMRT technique.) There are two obvious measures for the quality of the step-and-shoot delivery: (1) the *beam-on time* which is given by $\sum_{i=1}^{\kappa} \alpha_i$, and (2) the number of MLC-apertures used. The beam-on time is the actual time that the patient is exposed under the radiation beams. Minimizing beam-on time is crucial to reduce the patient's risk under irradiation and to reduce the delivery error caused by the tumor motion [1]. On the other hand, minimizing the number of MLC-apertures used for each IM (hence, minimizing the treatment time of each IM) is also important because it not only lowers the treatment cost for each patient but also enables hospitals to treat more patients [4].

To deliver the IMs, in current SLS method MLC leaves move along one direction (say, horizontally or vertically) during the entire delivery process. This uni-direction delivery may not fully utilize the capacity of the advanced MLC, which is rotatable. In fact, in order to improve the efficiency of the IMRT delivery, it was proposed recently to rotate the MLC between the delivery of the MLC-apertures for an IM [9, 2, 8].

In this paper, we propose to use two orthogonal directions to deliver an IM (i.e., horizontal and vertical) and formulate the following **matrix orthogonal decomposition (MOD)** problem: Given an $m \times n$ non-negative integer matrix $A = (a_{i,j}) \in {\mathbb{Z}^+}^{m \times n}$ (i.e., an IM) and an integer $\lambda \geq 1$, find two matrices (i.e., sub-IMs) $Q = (q_{i,j}), R = (r_{i,j}) \in {\mathbb{Z}^+}^{m \times n}$ such that:

(1) $A = \lambda Q + R$,
(2) the sum of the *horizontal complexity* $C_H(Q)$ of Q and the *vertical complexity* $C_V(R)$ of R is minimized, where

$$C_H(Q) = \sum_{i=1}^{m} \left(q_{i,1} + \sum_{j=2}^{n} \max(0, q_{i,j} - q_{i,j-1}) \right)$$
$$C_V(R) = \sum_{j=1}^{n} \left(r_{1,j} + \sum_{i=2}^{m} \max(0, r_{i,j} - r_{i-1,j}) \right) \quad (1)$$

Then, the sub-IM Q and R are delivered in two orthogonal directions. The rational behind this decomposition is based on the following observations. The beam-on time $T_{bot}(B[i])$ for delivering each row $B[i]$ of B equals to $\left(b_{i,1} + \sum_{j=2}^{n} \max(0, b_{i,j} - b_{i,j-1})\right)$ [7]. The horizontal complexity $C_H(Q)$ measures the total beam-on time of all rows of the IM Q when it is delivered horizontally, while the vertical complexity $C_V(R)$ is the total beam-on time of all columns of R when it is delivered vertically. Hence, the complexity of an IM that we use is closely related to the beam-on time of the IM. It is helpful to note that two IMs A and B with $A = \lambda \cdot B$ for some integer $\lambda > 1$, can be delivered by the same set of MLC-apertures. By adding the factor λ, it is very likely to reduce the total number of MLC-apertures since this can reduce the elements in R and thus the number of MLC-apertures used to deliver R. Most of current approaches for the SLS problem are based on a method for reducing the intensity level of IM matrices, then compute a set of MLC-apertures for the IM matrices with a smaller maximum intensity level [18, 4, 11, 12, 13, 3]. Our decomposition results in two "simpler" sub-IMs with smaller maximum intensity level, which, in turn, yields a more efficient delivery plan using fewer MLC-apertures and/or less total beam-on time.

We model the MOD problem as a minimum s-t cut problem. As an approach of partitioning, the minimum s-t cut has been extensively used. For example, several medical image segmentation techniques based on minimum s-t cuts were developed by, to name a few, Boykov and Jolly [19], Kim and Zabih [20], and Wu and Chen [16].

To our best knowledge, no previous work specifically for solving the matrix orthogonal decomposition problem discussed in this paper was known before. The closely related work is Chen *et al.*'s optimal linear time algorithm [3] for partitioning an IM matrix A into two sub-IMs of the form $\lambda \cdot Q + R$, without introducing new delivery error while minimizing the maximum intensity level of the sub-IM R.

In this paper, we develop an $T(mn\lfloor \frac{H}{\lambda} \rfloor, mn\lfloor \frac{H}{\lambda} \rfloor)$ time algorithm for the IM matrix orthogonal decomposition problem, where $T(n', m')$ is the time for computing a minimum s-t cut in an edge-weighted directed graph with $O(n')$ vertices and $O(m')$ edges. Our algorithm is based on a non-trivial graph construction scheme, which enables us to formulate the decomposition problem as computing a minimum s-t cut in a 3-D geometric multi-pillar graph (defined in Section 2.1). Experiments on randomly generated IM matrices and on clinical data are performed.

2 Our Algorithm for the IM Orthogonal Decomposition Problem

This section presents our efficient IM matrix orthogonal decomposition (MOD) algorithm. We model the MOD problem as a minimum s-t cut problem on a 3-D geometric multi-pillar graph by a complicated graph transformation scheme.

2.1 Modeling the MOD Problem

We define a 3-D geometric *multi-pillar graph* $G = (V, E)$ on a 2-D $m \times n$ grid Γ from the given IM matrix $A = (a_{i,j})_{m\times n}$ and the integer $\lambda > 0$, as follows.

Let $g(i, j)$ ($0 < i \leq m$ and $0 < j \leq n$) denote a grid point in Γ. For each grid point $g(i, j) \in \Gamma$, there is a set $Col(i, j)$ of $\lfloor \frac{a_{i,j}}{\lambda} \rfloor + 2$ (defined as *height* $h_{i,j}$ of the pillar) vertices in G corresponding to $a_{i,j}$ of the IM matrix A; $Col(i, j) = \{g(i, j, k) \mid k = 1, 2, \ldots, h_{i,j}\}$, called the (i, j)-*pillar* of G (see Figure 2(a) and (b) for an example). In addition, we add two dumbing vertices, a source s and a sink t, in G since we want to formulate our MOD problem as computing a minimum s-t cut in G.

For the ease to introducing edges in G, we here give some notation. We say that two pillars $Col(i, j)$ and $Col(i', j')$ are *adjacent* to each other if $|i-i'|+|j-j'| = 1$. For each pillar $Col(i, j)$, $g(i, j, 1)$ (resp., $g(i, j, h_{i,j})$) is called the *base* (resp., *top*) vertex of the pillar. For every vertex $g(i, j, k)$ in G with $i < m$ and $0 < k < h_{i,j}$, we define its *lower neighbor* and its *upper neighbor* on the pillar $Col(i + 1, j)$: (1) if $1 \leq (\lfloor \frac{a_{i+1,j}-a_{i,j}}{\lambda} \rfloor + k) \leq h_{i+1,j} - 1$, the lower neighbor of $g(i, j, k)$ is $g(i + 1, j, \lfloor \frac{a_{i+1,j}-a_{i,j}}{\lambda} \rfloor + k)$; (2) if $g(i + 1, j, k')$ is the lower neighbor of $g(i, j, k)$ and $k' < h_{i+1,j}$, the upper neighbor of $g(i, j, k)$ is $g(i + 1, j, k' + 1)$. Intuitively, the upper neighbor of $g(i, j, k)$ is the vertex on $Col(i+1, j)$ immediately "above" the lower neighbor of $g(i, j, k)$.

We are now ready to put directed edges in G. We introduce four subsets, E_{vt}, E_{hz}, E_q, and E_r, of directed edges into G, which are used to realize different parts of the complexity equation.

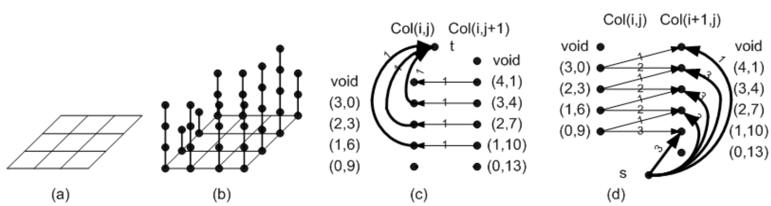

Fig. 2. (a) A 2-D grid. (b) Multi-pillar vertices of the IM in Figure 1. (c) Illustrating E_{hz} (thin edges) and E_q (thick edges) of the case $a_{i,j} = 9$, $a_{i,j+1} = 13$, and $\lambda = 3$. (d) Illustrating E_{vt} (thin) and E_r (thick) of the case $a_{i,j} = 9$, $a_{i+1,j} = 13$, and $\lambda = 3$.

- *The edges in* E_{vt}: Consider pillar $Col(i,j)$ and $Col(i+1,j)$, for $0 < i < m$ and $0 < j \leq n$. For each non-base vertex $g(i,j,k)$, two directed edges are put in E_{vt}: (1) a *lower edge* to its lower neighbor, and (2) an *upper edge* to its upper neighbor (see Figure 2(d)). The weight of the lower edge is $(\lambda - [a_{i+1,j} - a_{i,j}]\%\lambda)$ (note that "%" denotes a modulate operation), and the weight of the upper edge is $([a_{i+1,j} - a_{i,j}]\%\lambda)$. Meanwhile, for the base vertex $g(i,j,1)$, we put an *upper-base edge* with a weight of $([a_{i+1,j} - a_{i,j}]\%\lambda)$ to its upper neighbor. If the lower neighbor $g(i+1,j,lw)$ of $g(i,j,k)$ is not a base vertex, a set of directed edges (called the *lower-base edges*) from $g(i,j,k)$ to $g(i+1,j,k')$ for every $2 \leq k' \leq lw$ is introduced into E_{vt}; the weight of each of these edge is λ. Note that all the above edges are added only when the corresponding neighbor exists and the neighbor is not the base vertex.
- *The edges in* E_{hz}: Consider pillar $Col(i,j)$ and $Col(i,j+1)$, for $0 < i \leq m$ and $0 < j < n$. For each non-base vertex $g(i,j,k)$ on $Col(i,j)$, if $k < \min\{h_{i,j}, h_{i,j+1}\}$, we put an edge from $g(i,j,k)$ to $g(i,j+1,k)$ with a weight of 1. (see Figure 2(c)). If the height $h_{i,j+1}$ of $Col(i,j+1)$ is larger than the height $h_{i,j}$ of $Col(i,j)$, a directed edge of weight 1 is also introduced from each vertex $g(i,j+1,k)$ on the pillar $Col(i,j+1)$ to the top vertex $g(i,j,h_{i,j})$ of $Col(i,j)$, for $k = h_{i,j}, \ldots, h_{i,j+1} - 1$.
- *The edges in* E_q: For each non-base vertex $g(i,1,k)$ of every pillar $Col(i,1)$, $i = 1, 2, \ldots, m$, we put a directed edge of weight 1 from $g(i,1,k)$ to the sink t (see Figure 2(c) when $j = 1$).
- *The edges in* E_r: The top vertex of each pillar $Col(1,j)$, for $j = 1, 2, \ldots, n$, has a directed edge with a weight of $[a_{1,j}\%\lambda]$ from the source s. Additionally, For each non-base, non-top vertex $g(1,j,k)$ of every pillar $Col(1,j)$we add a directed edge of weight λ from the source s. Figure 2(d) shows an example for this construction when $i = 1$.

In addition, we introduce two more sets of edges, E_{mo} and E_{ad}, into G. The set of edges in E_{mo} is used to guarantee the monotonicity property of the result. While the edges in E_{ad} is employed to avoid the degeneracy of the solution.

- *The edges in* E_{mo}: On each pillar $Col(i,j)$, an edge of weight $+\infty$ is added from every vertex $g(i,j,k)$ to vertex $g(i,j,k-1)$ for $k = 2, 3, \ldots, h_{i,j}$.
- *The edges in* E_{ad}: An edge of weight $+\infty$ is put in E_{ad} from the source s to the base vertex of each pillar. Meanwhile, an edge of weight $+\infty$ is added from the top vertex of each pillar to the sink t.

Hence, the edge set E of G is $E_{vt} \cup E_{hz} \cup E_q \cup E_r \cup E_{mo} \cup E_{ad}$. We thus complete the construction of the multi-pillar graph G.

2.2 Computing an Optimal Matrix Orthogonal Decomposition

The graph G thus constructed allows us to find the optimal matrix orthogonal decomposition for the given IM matrix A, by computing a minimum-weight s-t cut in G. In order to do that, below we prove that following facts: (1) Any valid s-t cut $\mathcal{C}$ (i.e., the total edge weight $w(\mathcal{C})$ of $\mathcal{C}$ is finite) defines a feasible decomposition of A (i.e., $A = \lambda \cdot Q + R$), such that $C_H(Q) + C_V(R) = w(\mathcal{C})$; (2) any feasible decomposition of $A = \lambda \cdot Q + R$ specifies a valid s-t cut $\mathcal{C}$ in G, such that $w(\mathcal{C}) = C_H(Q) + C_V(R)$. Consequently, a valid s-t cut in G with the minimum total edge weight can be used to specify an optimal matrix orthogonal decomposition of A.

We first argue that any valid s-t cut in G corresponds to a feasible decomposition of A and any feasible decomposition of A corresponds to a valid s-t cut in G.

We have these two obvious observations:

Observation 1. *For a valid s-t cut $\mathcal{C} = (S, \bar{S})$ in G, the base vertices of all pillars are included in the source set S and the all top vertices of pillars are included in the sink set $\bar{S}$.*

Observation 2. *For a valid s-t cut $\mathcal{C} = (S, \bar{S})$ in G, if a vertex $g(i, j, k) \in Col(i, j)$ is in the source set S, each vertex $g(i, j, k')$ with $k' < k$ is also in S; if a vertex $g(i, j, k) \in Col(i, j)$ is in the sink set $\bar{S}$, every vertex $g(i, j, k')$ with $k' > k$ is also in the sink set $\bar{S}$.*

Thus, we can define a matrix $D = (d_{i,j})_{m \times n}, \quad d_{i,j} \in \mathbb{Z}^+, 1 \leq d_{i,j} \leq h_{i,j} - 1$ to describe a valid s-t cut $\mathcal{C} = (S, \bar{S})$ in G, such that for each pillar $Col(i, j)$, $S \cap Col(i, j) = \{g(i, j, k) \mid k = 1, 2, \ldots d_{i,j}\}$ and $\bar{S} \cap Col(i, j) = \{g(i, j, k) \mid k = d_{i,j} + 1, d_{i,j} + 2, \ldots h_{i,j}\}$. Then, a feasible decomposition of A, with $A = \lambda \cdot Q + R$, can be defined, as follows. For every pair (i, j) ($1 \leq i \leq m$ and $1 \leq j \leq n$), $q_{i,j} = d_{i,j} - 1$ (Note that $r_{i,j}$ is uniquely defined by $q_{i,j}$).

On the other hand, given a feasible decomposition $A = \lambda \cdot Q + R$, a valid s-t cut in G can be specified by letting $d_{i,j} = q_{i,j} + 1$ for every pair $(i, j) \in \Gamma$. Hence, the following lemma holds.

Lemma 1. *Any valid s-t cut in G has a one-to-one correspondence to a feasible decomposition of the IM matrix A.*

Next, we show that the total edge weight $w(\mathcal{C})$ of $\mathcal{C}$ equals to the complexity of the decomposition.

From Observations 1 and 2, edges in E_{mo} or in E_{ad} cannot be in $\mathcal{C}$. We thus only need to consider edges in E_{vt}, E_{hz}, E_q, and E_r. Actually, we are able to show that the total edge weight of the intersection of $\mathcal{C}$ with E_{vt}, E_{hz}, E_q, and E_r, equals to $\sum_{j=1}^{n} \sum_{i=2}^{m} \max(0, r_{i,j} - r_{i-1,j})$, $\sum_{i=1}^{m} \sum_{j=2}^{n} \max(0, q_{i,j} - q_{i,j-1})$, $\sum_{i=1}^{m} q_{i,1}$, and $\sum_{j=1}^{n} r_{1,j}$, respectively.

Lemma 2. *For a valid s-t cut $\mathcal{C} = (S, \bar{S})$ in G, the total edge weight of $\mathcal{C} \cap E_{vt}$ equals to $\sum_{j=1}^{n} \sum_{i=2}^{m} \max(0, r_{i,j} - r_{i-1,j})$.*

Proof. In the construction of the edge set E_{vt}, all edges are added between two adjacent pillars on the same column of Γ, we thus can first consider the edges that are between pillars $Col(i, j)$ and $Col(i + 1, j)$, and sum on the whole grid.

Recall our construction scheme and the constraint of range of k (the starting vertex must be in the source set and the ending vertex must be in the sink set), the number of lower edges in the cut $\mathcal{C}$ is,

$$\max\left\{0, \min\left\{h_{i+1,j} - 1 - \lfloor\frac{a_{i+1,j} - a_{i,j}}{\lambda}\rfloor, q_{i,j} + 1\right\} - \max\left\{2, q_{i+1,j} + 2 - \lfloor\frac{a_{i+1,j} - a_{i,j}}{\lambda}\rfloor\right\} + 1\right\}. \tag{2}$$

For the upper edges between $Col(i, j)$ and $Col(i+1, j)$, in a similar way, we can calculate that the number of such edges in the cut $\mathcal{C}$ is

$$\max\left\{0, \min\left\{h_{i+1,j} - 1 - \lfloor\frac{a_{i+1,j} - a_{i,j}}{\lambda}\rfloor, q_{i,j} + 1\right\} - \max\left\{1, q_{i+1,j} + 1 - \lfloor\frac{a_{i+1,j} - a_{i,j}}{\lambda}\rfloor\right\} + 1\right\}. \tag{3}$$

The number of the upper-base and lower-base edges between $Col(i, j)$ and $Col(i + 1, j)$ that are in the cut $\mathcal{C}$ is

$$\max(\lfloor\frac{a_{i+1,j} - a_{i,j}}{\lambda}\rfloor - q_{i+1,j}, 0). \tag{4}$$

When $0 \leq a_{i+1,j} - a_{i,j} < \lambda$ or $\lfloor\frac{a_{i+1,j}-a_{i,j}}{\lambda}\rfloor = 0$, number of edges in equation (2), (3), and (4) can be reduced to $\max(0, q_{i,j} - q_{i+1,j})$, $\max(0, q_{i,j} - q_{i+1,j} + 1)$, and 0. Thus the total weight of these edges can be calculated as $\max(r_{i+1,j} - r_{i,j}, 0)$.

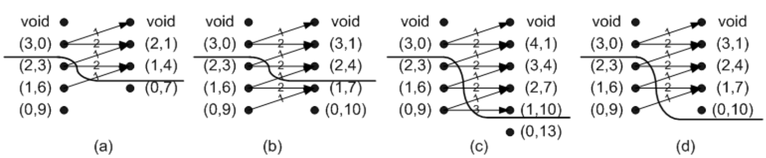

Fig. 3. Examples illustrating the proof of Lemma 2. (a) An example with $\lfloor\frac{a_{i+1,j}-a_{i,j}}{\lambda}\rfloor < 0$, wherein $a_{i,j} = 9$, $a_{i+1,j} = 7$, $q_{i,j} = 2$, $q_{i+1,j} = 0$, and $\lambda = 3$. (b) Increasing $a_{i+1,j}$ to 10 and $q_{i+1,j}$ to 1, $r_{i,j}$ will not be changed, neither are the edges across the cut. (c) An example with $\lfloor\frac{a_{i+1,j}-a_{i,j}}{\lambda}\rfloor > 0$, wherein $a_{i,j} = 9$, $a_{i+1,j} = 13$, $q_{i,j} = 2$, $q_{i+1,j} = 0$, and $\lambda = 3$. (d) Decreasing $a_{i+1,j}$ to 10 and keeping $q_{i+1,j}$ unchanged, $r_{i+1,j}$ is decreased by 3, but an edge of weight 3 can counteract this change.

When $\lfloor\frac{a_{i+1,j}-a_{i,j}}{\lambda}\rfloor > 0$, we can decrease $a_{i+1,j}$ by $\lambda\lfloor\frac{a_{i+1,j}-a_{i,j}}{\lambda}\rfloor$ and $q_{i+1,j}$ by $\min\left\{q_{i+1,j}, \lfloor\frac{a_{i+1,j}-a_{i,j}}{\lambda}\rfloor\right\}$ (to make sure that $q'_{i+1,j} \geq 0$) to $a'_{i+1,j}$ and $q'_{i+1,j}$, respectively. Observe that the case for $q_{i+1,j} \geq \lfloor\frac{a_{i+1,j}-a_{i,j}}{\lambda}\rfloor$ is the same as the case for $\lfloor\frac{a_{i+1,j}-a_{i,j}}{\lambda}\rfloor < 0$. However, if $q_{i+1,j} < \lfloor\frac{a_{i+1,j}-a_{i,j}}{\lambda}\rfloor$, the new $r'_{i+1,j} = a'_{i+1,j} - \lambda q'_{i+1,j}$ will be $(\lfloor\frac{a_{i+1,j}-a_{i,j}}{\lambda}\rfloor - q_{i+1,j}) \times \lambda$ less than the actual $r_{i+1,j}$. The term $\max\left\{\lfloor\frac{a_{i+1,j}-a_{i,j}}{\lambda}\rfloor - q_{i+1,j}, 0\right\} \times \lambda$ can then counteract the change. Hence, in this case, we again have the total weight of the edges in the intersection of the

s-t cut $\mathcal{C}$ and the edges between $Col(i,j)$ and $Col(i+1,j)$, is $\max(r_{i+1,j} - r_{i,j}, 0)$. Figure 3 (c) and (d) illustrate the essential idea using an example.

When $\lfloor \frac{a_{i+1,j} - a_{i,j}}{\lambda} \rfloor < 0$, the situation is similar and Figure 3 (a) and (b) show an example to illustrate the idea.

Taking all the above possibilities into account, we conclude that the total weight of the edges in the intersection of the s-t cut $\mathcal{C}$ and the edges between $Col(i,j)$ and $Col(i+1,j)$, is $\max(r_{i+1,j} - r_{i,j}, 0)$.

By considering all pairs of adjacent pillars on the same columns of Γ, we have $w(\mathcal{C} \cap E_{vt}) = \sum_{j=1}^{n} \sum_{i=2}^{m} \max(0, r_{i,j} - r_{i-1,j})$. Thus, Lemma 2 follows. □

Using a similar argument as for Lemma 2, we have the following lemmas.

Lemma 3. *For a valid s-t cut* $\mathcal{C} = (S, \bar{S})$ *in* G*, the total edge weight of* $\mathcal{C} \cap E_{hz}$ *equals to* $\sum_{i=1}^{m} \sum_{j=2}^{n} \max(0, q_{i,j} - q_{i,j-1})$.

Lemma 4. *For a valid s-t cut* $\mathcal{C} = (S, \bar{S})$ *in* G*, the total edge weight of* $\mathcal{C} \cap E_q$ *equals to* $\sum_{i=1}^{m} q_{i,1}$.

Lemma 5. *For a valid s-t cut* $\mathcal{C} = (S, \bar{S})$ *in* G*, the total edge weight of* $\mathcal{C} \cap E_r$ *equals to* $\sum_{j=1}^{n} r_{1,j}$.

Putting Lemmas 2 - 5 all together, we have the following fact.

Lemma 6. *For any valid s-t cut* $\mathcal{C}$ *in* G *and its specified decomposition of* A*, with* $A = \lambda \cdot Q + R$*, we have* $w(\mathcal{C}) = C_H(Q) + C_V(R)$.

From Lemmas 1 and 6, an minimum-weight s-t $\mathcal{C}^*$ in G can be used to define an optimal matrix orthogonal decomposition of A, with $A = \lambda \cdot Q^* + R^*$, such that $C_H(Q^*) + C_V(R^*)$ is minimized. Note that $|V| = O(mn\lfloor \frac{H}{\lambda} \rfloor)$ and $|E| = O(mn\lfloor \frac{H}{\lambda} \rfloor)$, where H is the largest intensity level in the IM matrix A. Denote by $T(n', m')$ the time for finding a minimum s-t cut in an edge-weighted directed graph with $O(n')$ vertices and $O(m')$ edge. We have our main result.

Theorem 3. *The MOD problem can be solved in* $T(mn\lfloor \frac{H}{\lambda} \rfloor, mn\lfloor \frac{H}{\lambda} \rfloor)$ *time.*

3 Experiment Results

To evaluate our algorithm, we performed some statistical studies using 1000 randomly generated 15×15 IM matrices each with intensity levels range from 4 to 64 in powers of 2. The number of MLC-apertures are computed using Xia and Verhey's algorithm [18] without considering interleaf motion constraint.

Table 1 shows percentage of IMs getting improved and the average results (both beam-on time and number of MLC-apertures) before and after performing our decomposition method (the average is calculated based only on those IMs getting improved). We observed that our MOD algorithm generated as much as 38.1% less MLC-apertures and 33.3% less beam-on time than single direction delivery.

Table 1. The average beam-on time and the number of MLC-apertures

	# of MLC-apertures			beam-on time		
	%improved	avg before	avg after	%improved	avg before	avg after
4	9%	5.98±0.60	5.78±0.44	25%	7.75±0.99	7.32±0.80
8	24%	9.07±0.64	8.46±0.66	66%	17.09±1.96	15.47±1.56
16	25%	12.04±0.71	11.44±0.58	73%	35.33±4.27	32.05±3.95
32	45%	14.91±0.85	14.20±0.69	81%	69.71±8.93	63.41±6.46
64	54%	18.16±0.94	17.11±0.72	94%	144.15±18.23	129.04±13.08

We have also experimented with some real medical data sets available to us. 77% IMs that we tested on got improved number of MLC-apertures. Our MOD algorithm produced as much as 27.3% less MLC-apertures with an average of 13.1% comparing with the SLS method using a single direction for delivery.

The experiments are performed on a Pentium-D 2.8GHz computer with 3.5GB of memory. We used a program provided by Matlab to compute the minimum *s*-*t* cut in a graph, and expected to have a much faster execution time by implementing the minimum cut algorithm using C. The average execution time of decomposition is shown in Table 2. Our experiments on randomly generated IMs and on the clinical data demonstrated the efficiency of our MOD algorithm. Although the worst cast running time of our MOD algorithm is pseudo-polynomial with respect to the maximum intensity level H of the IM matrix, its practical execution time on real medical data is expected to be quite short, since on the medical data sets used in current clinical treatments, the maximum intensity level of an IM matrix is rarely larger than 100 and is mostly about tens.

Table 2. Execution Times (in seconds)

Maximum intensity	Size 10×10		Size 15×15	
level (H)	$\lambda = 1$	$\lambda = \lfloor\sqrt{H}\rfloor$	$\lambda = 1$	$\lambda = \lfloor\sqrt{H}\rfloor$
4	0.3125	0.1955	0.9925	0.6330
8	0.6090	0.3360	2.2420	1.2815
16	1.6570	0.4135	5.7425	1.3360
32	5.1560	0.7265	19.1880	2.6635

References

[1] N. Boland, H.W. Hamacher, and F. Lenzen, Minimizing Beam-on Time in Cancer Radiation Treatment Using Multileaf Collimators, *Networks*, 43(4):226-240, 2004.
[2] Y. Chen, Q. Hou, and J.M. Galvin, A graph-searching method for MLC leaf sequencing under constraints, *Med. Phys.*, 31(2004), pp. 1504-11.
[3] D.Z. Chen, X.S. Hu, S. Luan, and C. Wang, Mountain Reduction, Block Matching, and Medical Applications. *Proc. of the 21st Annual Symposium on Computational Geometry (SoCG)*, Pisa, Italy, pp. 35-44, 2005.

[4] D.Z. Chen, X.S. Hu, S. Luan, C. Wang, and X. Wu. Geometric Algorithms for Static Leaf Sequencing Problems in Radiation Therapy. *International Journal of Computational Geometry and Applications*, 14(5):311-339, 2004.

[5] J. Dai and Y. Zhu. Minimizing the Number of Segments in a Delivery Sequence for Intensity-Modulated Ratiation Therapy with Multileaf Collimator. *Med. Phys.,* 28(10):2113-2120, 2001.

[6] N. Dogan, L.B. Leybovich, A. Sethi, and B. Emami, Automatic Feathering of Split Fields for Step-and-Shoot Intensity Modulated Radiation Therapy, *Phys. Med. Biol.*, 48:1133-1140, 2003.

[7] K. Engel, A New Algorithm for Optimal Multileaf Collimator Field Segmentation, *http://www.trinity.edu/aholder/HealthApp/oncology/papers/paperlist.html,* March 2003.

[8] B. Hardemark, H. Rehbinder, and J. Lof, Rotating the MLC Between Segments Improves Performance in step-andshoot IMRT delivery, Presentation at the AAPM 45th Meeting (1014 August 2003), *Med. Phys.*, 30(2003).

[9] T. Kalinowski, Reducing the number of monitor units in multileaf collimator field segmentation. *Physics in Medicine and Boilogy* 50:1147–1161, 2005.

[10] M. Langer, V. Thai, and L. Papiez, Improved Leaf Sequencing Reduces Segments of Monitor Units Needed to Deliver IMRT Using Multileaf Collimators. *Med. Phys.* 28:2450-2458, 2001.

[11] L.D. Potter, S.X. Chang, T.J. Cullip, and A.C. Siochi, A Quality and Efficiency Analysis of the $IMFAST^{TM}$ Segmentation Algorithm in Head and Neck "Step & Shoot" IMRT Treatments, *Med. Phys.,* 29(3)(2002), pp. 275-283.

[12] W. Que, Comparison of Algorithms for Multileaf Collimator Field Segmentation, *Med. Phys.,* 26(1999), pp. 2390-2396.

[13] R. Svensson, P. Kallman, and A. Brahme, An Analytical Solution for the Dynamic Control of Multileaf Collimation, *Phys. in Med. and Biol.,* 39(1994), pp. 37-61.

[14] S.Webb. *The Physics of Conformal Radiotherapy Advances in Technology.* Bristol, Institute of Physics Publishing, 1997.

[15] S.Webb, *Intensity-Modulated Radiation Therapy*, Institute of Cancer Research and Royal Marsden NHS Trust, Jan. 2001.

[16] X. Wu and D.Z. Chen, Optimal Net Surface Problems with Applications, *lecture Notes in Computer Science,* Vol. 2380, Springer Verlag, *Proc. of the 29th International Colloquium on Automata, Languages and Programming (ICALP),* Malaga, Spain, July 2002, pp. 1029-1042.

[17] Q. Wu, M. Arnfield, S. Tong, Y. Wu, and R. Mohan, Dynamic Splitting of Large Intensity-Modulated Fields, *Phys. Med. Biol.*, 45:1731-1740, 2000.

[18] P. Xia and L.J. Verhey., MLC Leaf Sequencing Algorithm for Intensity Modulated Beams with Multiple Static Segments. *Med. Phys.,* 25:1424-1434, 1998.

[19] Y. Boykov and M.-P. Jolly, Interactive Organ Segmentation Using Graph Cuts", *Proc. Medical Image Computing and Computer-Assisted Intervention (MICCAI),* 276-286, 2000.

[20] J. Kim and R. Zabih, A Segmentation Algorithm for Contrast-Enhanced Images, *Proc. IEEE Int'l Conf. Computer Vision,* 502-509, 2003.