

A Minimum Cost Flow Formulation for Approximated MLC Segmentation*

Thomas Kalinowski

Abstract

Shape matrix decomposition is a subproblem in radiation therapy planning. A given fluence matrix A has to be written as a sum of shape matrices corresponding to homogeneous fields that can be shaped by a multileaf collimator (MLC). We solve the problem of finding an approximation B of A satisfying prescribed upper and lower bounds for each entry. The approximation B is determined such that the corresponding fluence can be realized with a prescribed delivery time using a multileaf collimator with an interleaf collision constraint, and under this condition the distance between A and B is minimized.

Keywords: intensity modulated radiation therapy, multileaf collimator, minimum cost flows

1 Introduction

Radiation therapy is an important method in cancer treatment. Basically, the aim is to destroy the tumor while minimizing the damage to the healthy tissue, in particular to sensitive structures or organs at risk. In clinical practice it is common to use a linear accelerator which can release radiation from different directions. In addition, a multileaf collimator (MLC) can be used to cover certain parts of the irradiated area. An MLC consists of two banks of metal leaves that are arranged pairwise such that each pair consists of a left leaf and a right leaf which can be moved into the radiation beam from their respective sides. Figure 1 illustrates how an MLC can be used to modulate the intensity. The subject of the present paper is one step of the treatment planning for an MLC in the *step-and-shoot mode*. That term means the radiation is switched off while the leaves are moving, so the task is to determine finitely many fields such that their superposition yields the required fluence. The two main optimization goals considered in the literature are the minimization of the delivery time (DT) and minimization of the number of used shapes. This problem has been considered by many authors (see [8] and the references therein). There are many algorithms for this task, using different reformulations of the problem and including different technological constraints, such as the interleaf collision constraint (ICC) and the tongue-and-groove constraint. In [5] it was suggested to decompose an approximation of A . This might be necessary if the delivery time for an exact decomposition of A is prohibitively large. It is

* *Networks* **57**, 135–140, 2011, doi:10.1002/net.20394

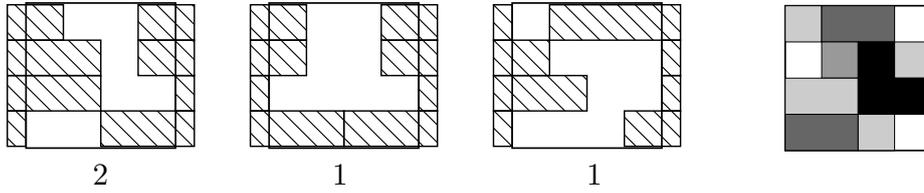


Figure 1: Generating an intensity modulated radiation field by superimposing three homogeneous fields shaped by an MLC. The shaded areas are the regions covered by MLC leaves, the numbers indicate for how long the corresponding field is irradiated, and the greyscales in the rightmost square show the total fluence distribution.

further justified by the fact that A is a result of numerical computations based on simplified models, so there should be an error interval attached to each entry. There are two natural objectives for this approximation problem. First, the delivery time for the approximation matrix should be small and second, a given delivery time should be realized by changing A as little as possible. Both of these problems were solved for single row matrices (and consequently in general for MLCs with independent rows) in [5]. In this paper we consider MLCs that have an interleaf collision constraint, which means that an overlap between opposite leaves in consecutive rows is not allowed. Given a fluence matrix A and upper and lower bounds for the entries of the approximation matrix, the minimal possible delivery time of an approximation was determined in [9], where the authors also described a heuristic method for the reduction of the distance between A and the approximation matrix B (to be defined later). The main result of the present paper is a minimum cost flow formulation of the exact minimization of this distance. In Section 2 we give a precise formulation of the approximation problem and we introduce some notation. Section 3 contains our main result: the approximation problem is dual to a minimum cost flow problem. Finally, Section 4 contains some computational results.

2 Problem formulation

Throughout the paper we use the standard notation

$$[k] = \{1, 2, \dots, k\}, \quad [k, l] = \{k, k + 1, \dots, l\}$$

for integers k and l with $k \leq l$. As in [9], we start with a fluence matrix A of size $m \times n$, and two matrices \underline{A} and \overline{A} containing the lower and upper bounds for the entries:

$$0 \leq \underline{a}_{ij} \leq a_{ij} \leq \overline{a}_{ij} \quad (i, j) \in [m] \times [n].$$

Definition 1 (Feasible Approximation). Any integer matrix B with

$$\underline{a}_{ij} \leq b_{ij} \leq \overline{a}_{ij} \quad ((i, j) \in [m] \times [n])$$

is called *feasible approximation* of A . The *total change* $TC(B)$ of a feasible approximation B is defined by

$$TC(B) = \sum_{i=1}^m \sum_{j=1}^n |b_{ij} - a_{ij}|.$$

The homogeneous fields that can be shaped by the MLC are described by binary matrices of size $m \times n$ which we call *shape matrices*.

Definition 2 (Shape matrix). An $m \times n$ matrix S is a *shape matrix* if there are pairs of integers (l_i, r_i) ($i \in [m]$), such that the following conditions are satisfied:

1. $s_{ij} = \begin{cases} 1 & \text{if } l_i < j < r_i, \\ 0 & \text{otherwise.} \end{cases}$
2. $l_i < r_{i+1}$ and $r_i > l_{i+1}$ for all $i \in [m - 1]$.

The first condition is just stating that for each row i , there are a left leaf covering bixels $(i, 1), \dots, (i, l_i)$ and a right leaf covering bixels $(i, r_i), \dots, (i, n)$, while the bixels $(i, l_i + 1), \dots, (i, r_i - 1)$ are exposed to radiation. The second condition is called the interleaf collision constraint (ICC). It ensures the left leaf of row i and the right leaf of row $i \pm 1$ do not overlap, which is required by some widely used MLCs. An MLC leaf sequence for A corresponds to a representation of A as a weighted sum of shape matrices.

Definition 3 (Shape matrix decomposition). A *shape matrix decomposition* of A is a representation of A as a positive integer linear combination of shape matrices

$$A = \sum_{t=1}^k u_t S^{(t)}.$$

The *delivery time* (DT) of this decomposition is just the sum of the coefficients,

$$DT = \sum_{t=1}^k u_t.$$

Example 1. For the shape matrix decomposition

$$\begin{pmatrix} 1 & 3 & 3 & 0 \\ 0 & 2 & 4 & 1 \\ 1 & 1 & 4 & 4 \\ 3 & 3 & 1 & 0 \end{pmatrix} = 2 \cdot \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix},$$

corresponding to Figure 1, we have $DT = 4$.

There are 3 natural optimization problems [9].

MinDT. Find a shape matrix decomposition $A = \sum_{t=1}^k u_t S^{(t)}$ such that $DT = \sum_{t=1}^k u_t$ is minimal.

Approx-MinDT. Find a feasible approximation B and a shape matrix decomposition $B = \sum_{t=1}^k u_t S^{(t)}$ such that $DT = \sum_{t=1}^k u_t$ is minimal.

Approx-MinTC. For a given delivery time \tilde{c} , find a feasible approximation B and a shape matrix decomposition

$$B = \sum_{t=1}^k u_t S^{(t)} \quad (1)$$

such that $\sum_{t=1}^k u_t \leq \tilde{c}$, and under this condition $TC(B)$ is minimal.

The first problem **MinDT** is the exact decomposition problem which can be solved by several efficient algorithms [3, 7, 10]. The second problem **Approx-MinDT** was solved in [9]. In the present paper we consider the third problem **Approx-MinTC**. Throughout the paper, we will always assume that the problem is feasible. In practice that can be realized by solving **Approx-MinDT** first. This yields the minimal possible value for \tilde{c} , and for each value as least as large **Approx-MinTC** is feasible.

3 A solution of the problem Approx-MinTC

We start by formulating an LP model for **Approx-MinTC**. Since we are only interested in the sum of the coefficients, we may assume that all the coefficients u_t in (1) are equal to 1 (allowing the same shape matrix $S^{(t)}$ for different values of t). We introduce variables L_{ij} and R_{ij} for $(i, j) \in [m] \times [n]$. Formally, if the shape matrix $S^{(t)}$ in the decomposition $B = \sum_{t=1}^k S^{(t)}$ is determined by the parameters $(l_i^{(t)}, r_i^{(t)})$ for $i \in [m]$, our variables are

$$L_{ij} = \left| \left\{ t : l_i^{(t)} < j \right\} \right|, \quad \text{and} \quad R_{ij} = \left| \left\{ t : r_i^{(t)} \leq j \right\} \right|.$$

In other words, the variable L_{ij} is the number of shapes where bixel (i, j) is not covered by the left leaf, while R_{ij} counts the shapes where bixel (i, j) is covered by the right leaf. Obviously, this gives $b_{ij} = L_{ij} - R_{ij}$. In addition, we introduce the variables $x_{ij} = |a_{ij} - b_{ij}|$ for $(i, j) \in [m] \times [n]$. Now we can formulate **Approx-MinTC** for a given delivery time \tilde{c} as an LP. To clarify our notation we also write down the dual variables for the constraints.

$$\min \sum_{i=1}^m \sum_{j=1}^n x_{ij} \quad (2)$$

$$L_{i,j+1} - L_{ij} \geq 0 \quad \alpha_{ij} \geq 0 \quad ((i, j) \in [m] \times [n-1]), \quad (3)$$

$$-L_{in} \geq -\tilde{c} \quad \alpha_{in} \geq 0 \quad (i \in [m]), \quad (4)$$

$$R_{i,j+1} - R_{ij} \geq 0 \quad \beta_{ij} \geq 0 \quad ((i, j) \in [m] \times [n-1]), \quad (5)$$

$$R_{i1} \geq 0 \quad \beta_{i0} \geq 0 \quad (i \in [m]) \quad (6)$$

$$R_{ij} - L_{ij} \geq -\bar{a}_{ij} \quad y_{ij} \geq 0 \quad ((i, j) \in [m] \times [n]), \quad (7)$$

$$L_{ij} - R_{ij} \geq \underline{a}_{ij} \quad z_{ij} \geq 0 \quad ((i, j) \in [m] \times [n]), \quad (8)$$

$$L_{ij} - R_{i+1,j} \geq 0 \quad u_{ij} \geq 0 \quad ((i, j) \in [m-1] \times [n]), \quad (9)$$

$$L_{ij} - R_{i-1,j} \geq 0 \quad v_{ij} \geq 0 \quad ((i, j) \in [2, m] \times [n]), \quad (10)$$

$$R_{ij} - L_{ij} + x_{ij} \geq -a_{ij} \quad p_{ij} \geq 0 \quad ((i, j) \in [m] \times [n]), \quad (11)$$

$$L_{ij} - R_{ij} + x_{ij} \geq a_{ij} \quad q_{ij} \geq 0 \quad ((i, j) \in [m] \times [n]). \quad (12)$$

By definition, the variables L_{ij} and R_{ij} should be nonnegative. We do not want to require this explicitly in the LP since we want to have equality constraints in the dual, but note that nonnegativity is implied: constraints (6) together with (5) force all the variables R_{ij} to be nonnegative, and from (8) it follows that also $L_{ij} \geq 0$ for all $(i, j) \in [m] \times [n]$. Constraints (3), (5) and (9), (10) are consequences of the inclusions

$$\begin{aligned} \{t : l_i^{(t)} < j\} &\subseteq \{t : l_i^{(t)} < j + 1\}, & \{t : r_i^{(t)} \leq j\} &\subseteq \{t : r_i^{(t)} \leq j + 1\} \\ \{t : r_{i+1}^{(t)} \leq j\} &\subseteq \{t : l_i^{(t)} \leq j\}, & \{t : r_{i-1}^{(t)} \leq j\} &\subseteq \{t : l_i^{(t)} \leq j\}, \end{aligned}$$

where the inclusions in the second row follow from the interleaf collision constraint. The constraints (7) and (8) ensure that $\underline{a}_{ij} \leq b_{ij} \leq \bar{a}_{ij}$, while (4) is the constraint that the total number of shapes is at most \tilde{c} . Finally, constraints (11) and (12) are equivalent to $x_{ij} \geq |a_{ij} - b_{ij}|$, and the objective is to minimize the sum of all the deviations $|a_{ij} - b_{ij}|$.

We remark that the values L_{ij} and R_{ij} do not uniquely determine the shape matrix decomposition, because in the transformation from the shape matrices $S^{(t)}$ to the cardinalities L_{ij} and R_{ij} we lost some information. It is even not completely obvious that a solution of the LP always yields a feasible decomposition. But fortunately, a natural approach to construct appropriate shape matrices works: we define shape matrices $S^{(t)}$ such that the leaves move only from left to right as t increases. More precisely, for a given solution of the problem (2)–(12) we consider the sets $\mathcal{L}_{ij} = [L_{ij}]$ and $\mathcal{R}_{ij} = [R_{ij}]$ for $(i, j) \in [m] \times [n]$ and put

$$s_{ij}^{(t)} = \begin{cases} 1 & \text{if } t \in \mathcal{L}_{ij} \setminus \mathcal{R}_{ij} \\ 0 & \text{otherwise} \end{cases} \quad ((i, j) \in [m] \times [n], t \in [L]),$$

where $L = \max\{L_{in} : i \in [m]\}$. These matrices have the first property required in Definition 2 with parameters

$$l_i^{(t)} = 0 \text{ for } t \leq L_{i1} \quad (i \in [m]), \quad (13)$$

$$r_i^{(t)} = 1 \text{ for } t \leq R_{i1} \quad (i \in [m]), \quad (14)$$

$$l_i^{(t)} = j - 1 \text{ for } L_{i,j-1} < t \leq L_{ij} \quad ((i, j) \in [m] \times [2, n]), \quad (15)$$

$$r_i^{(t)} = j \text{ for } R_{i,j-1} < t \leq R_{ij} \quad ((i, j) \in [m] \times [2, n]). \quad (16)$$

For $t > L_{in}$, there is a zero row in the i -th row of $S^{(t)}$, which can be realized by parameters $l_i^{(t)} = n$ and $r_i^{(t)} = n + 1$. The interleaf collision constraint $l_i^{(t)} < r_{i+1}^{(t)}$ is satisfied for every $t \in [L]$ and every $i \in [m - 1]$. If $l_i^{(t)} = 0$ this is obvious. Otherwise we have $l_i^{(t)} = j - 1$ for some $j \in [2, n + 1]$. Then $t > L_{i,j-1} \geq R_{i+1,j-1}$, and consequently $r_{i+1}^{(t)} \geq j$. The interleaf collision constraint $l_i^{(t)} < r_{i-1}^{(t)}$ is proved similarly. Finally, using (7) and (8) we have

$$b_{ij} = \sum_{t=1}^L s_{ij}^{(t)} = |\mathcal{L}_{ij} \setminus \mathcal{R}_{ij}| = L_{ij} - R_{ij} \in [\underline{a}_{ij}, \bar{a}_{ij}].$$

Now we dualize the LP (2)– (12) to obtain the problem **TC-Dual**:

$$\max \quad \sum_{i=1}^m \sum_{j=1}^n (-a_{ij}p_{ij} + a_{ij}q_{ij} + \underline{a}_{ij}z_{ij} - \bar{a}_{ij}y_{ij}) - \tilde{c} \sum_{i=1}^m \alpha_{in} \quad (17)$$

$$L_{ij} : \quad \begin{aligned} -\alpha_{ij} + \alpha_{i,j-1} + q_{ij} - p_{ij} + z_{ij} \\ -y_{ij} + u_{ij} + v_{ij} = 0 \end{aligned} \quad ((i, j) \in [m] \times [2, n]), \quad (18)$$

$$L_{i1} : \quad \begin{aligned} -\alpha_{i1} + q_{i1} - p_{i1} + z_{i1} - y_{i1} \\ +u_{i1} + v_{i1} = 0 \end{aligned} \quad (i \in [m]), \quad (19)$$

$$R_{ij} : \quad \begin{aligned} -\beta_{ij} + \beta_{i,j-1} - q_{ij} + p_{ij} - z_{ij} \\ +y_{ij} - u_{i-1,j} - v_{i+1,j} = 0 \end{aligned} \quad ((i, j) \in [m] \times [n-1]), \quad (20)$$

$$R_{in} : \quad \begin{aligned} \beta_{i,n-1} - q_{in} + p_{in} - z_{in} + y_{in} \\ -u_{i-1,n} - v_{i+1,n} = 0 \end{aligned} \quad ((i, j) \in [m] \times [n-1]), \quad (21)$$

$$x_{ij} : \quad p_{ij} + q_{ij} \leq 1 \quad ((i, j) \in [m] \times [n]). \quad (22)$$

Formally, we would have to write down several of the constraints for $i = 1$ and $i = m$ separately, since in these cases the variables $u_{i-1,j}$ and v_{ij} (resp. u_{ij} and $v_{i+1,j}$) are missing. In order avoid an unnecessary blowup of the formalism, we use the convention that

$$u_{0j} = u_{mj} = v_{1,j} = v_{m+1,j} = 0 \quad (j \in [n]).$$

Clearly, the objective (17) is equivalent to minimizing

$$\sum_{i=1}^m \sum_{j=1}^n (a_{ij}p_{ij} - a_{ij}q_{ij} - \underline{a}_{ij}z_{ij} + \bar{a}_{ij}y_{ij}) + \tilde{c} \sum_{i=1}^m \alpha_{in}, \quad (23)$$

and we will see that **TC-Dual** is equivalent to the problem of finding a $Q - S$ -flow of minimum cost in the following network N . The node set V of the underlying digraph consists of two nodes $(i, j, 0)$ and $(i, j, 1)$ for each bixel $(i, j) \in [m] \times [n]$ and two additional nodes Q and S :

$$V = \{Q, S\} \cup \{(i, j, k) : (i, j) \in [m] \times [n], k \in \{0, 1\}\}.$$

The arc set E is constructed corresponding to the variables in **TC-Dual**. From node Q , there is an outgoing arc to every node $(i, 1, 0)$ with corresponding flow variable β_{i0} . For the nodes $(i, j, 1)$ with $(i, j) \in [m] \times [n-1]$, we have an outgoing arc to $(i, j+1, 1)$ with corresponding flow variable α_{ij} and two outgoing arcs to $(i, j, 0)$ corresponding to flow variables p_{ij} and y_{ij} . Similarly, the nodes $(i, n, 1)$ have two outgoing arcs to $(i, n, 0)$ with flow variables p_{in} and y_{in} , but their outgoing arc with flow variable α_{in} goes to S . From a node $(i, j, 0)$ with $(i, j) \in [m] \times [n-1]$, we have an arc to $(i, j+1, 0)$ with flow variable β_{ij} and two arcs to $(i, j, 1)$ with flow variables q_{ij} and z_{ij} . For nodes $(i, n, 0)$ with $i \in [m]$ the arc to the sink which would correspond to β_{in} is missing, but we still have the two arcs to $(i, n, 1)$ with flow variables q_{in} and z_{in} . In addition, for $(i, j, 0)$, if $i < m$, there is an arc to

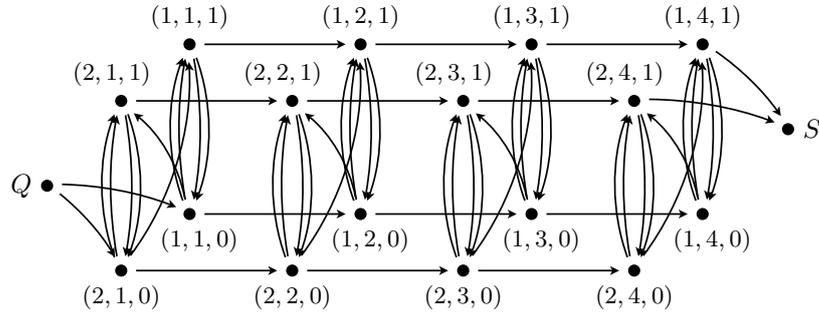


Figure 2: The digraph for the network model of the problem **TC-Dual** for a 2×4 -matrix.

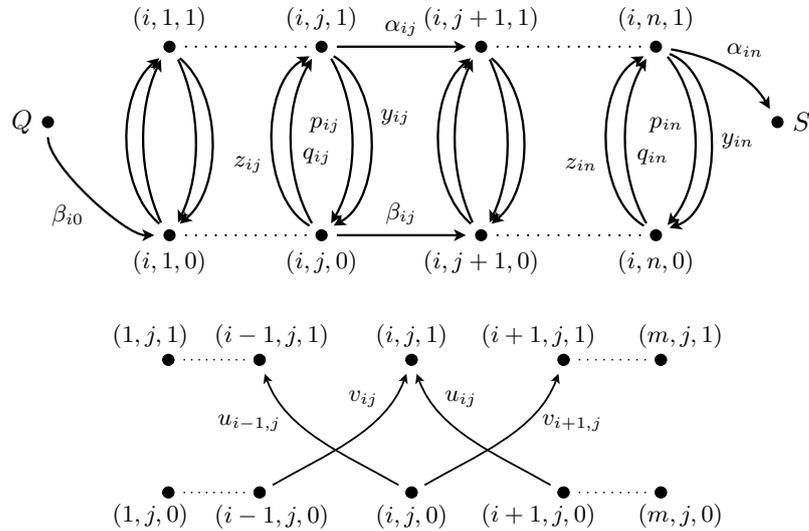


Figure 3: The structure of the digraph for the network model of the problem **TC-Dual**.

$(i + 1, j, 1)$ with flow variable $v_{i+1,j}$, and for $i > 1$ there is an arc to $(i - 1, j, 1)$ with flow variable $u_{i-1,j}$.

This digraph is illustrated in Figures 2 and 3. The capacity function is very simple: The arcs with flow variables p_{ij} and q_{ij} have capacity 1, and all the other arcs have infinite capacities. The cost function is defined such that the cost of a flow is precisely the objective function (23). Identifying the arcs with their corresponding flow variables, this can be described as follows. For $(i, j) \in [m] \times [n]$, the costs of the arcs p_{ij} , q_{ij} , z_{ij} and y_{ij} are a_{ij} , $-a_{ij}$, $-\underline{a}_{ij}$ and \bar{a}_{ij} , respectively. For $i \in [m]$, the cost of arc α_{in} is \tilde{c} . All the other arcs have zero cost.

Since the arcs p_{ij} and q_{ij} form a cycle of zero cost, we may assume that $p_{ij}q_{ij} = 0$ for every $(i, j) \in [m] \times [n]$. Under this assumption constraints (22) correspond to the capacity constraints for the arcs p_{ij} and q_{ij} , while the constraints (18)–(21) are the flow conservation constraints at the nodes $x \in V \setminus \{Q, S\}$. So we have proved the following theorem.

Theorem 1. *The problem **TC-Dual** is equivalent to the minimum cost $Q-S$ -flow problem*

in the network N .

By standard results from network flow theory [1], we obtain a solution of **Approx-MinTC** from a flow $\phi : E \rightarrow \mathbb{N}$ of minimum cost as follows. The residual network on the node set V with arc set E' and cost function $\text{cost}' : E' \rightarrow \mathbb{Z}$ is defined by

$$\begin{aligned} \phi(xy) < \text{capacity}(xy) &\implies xy \in E', \text{cost}'(xy) = \text{cost}(xy), \\ \phi(xy) > 0 &\implies yx \in E', \text{cost}'(yx) = -\text{cost}(xy). \end{aligned}$$

Recall that L_{ij} and R_{ij} are the dual variables of the flow conservation constraints in $(i, j, 1)$ and $(i, j, 0)$, respectively, so we can determine them as the negative distances (with respect to cost') from Q to $(i, j, 1)$ and $(i, j, 0)$, respectively. We obtain the approximation matrix B by $b_{ij} = L_{ij} - R_{ij}$, and a shape matrix decomposition $B = \sum_{t=1}^k S^{(t)}$ with

$$s_{ij}^{(t)} = \begin{cases} 1 & \text{if } l_i^{(t)} < j < r_i^{(t)}, \\ 0 & \text{otherwise,} \end{cases} \quad ((i, j) \in [m] \times [n])$$

where the parameters $l_i^{(t)}$ and $r_i^{(t)}$ are determined according to (13)–(16).

Example 2. We illustrate the method for $m = 1, n = 6$. Suppose we are given the following matrices A, \underline{A} and \overline{A} :

$$(5 \ 3 \ 3 \ 1 \ 5 \ 5), (4 \ 2 \ 2 \ 0 \ 4 \ 4), (6 \ 4 \ 4 \ 2 \ 6 \ 6).$$

For matrix A the minimal delivery time is 9, and we want to have an approximation matrix B with a delivery time of 6. The network is shown in Figure 4. The labels on the arcs are the nonzero costs, so a unit flow along the dashed path has cost -3 , while a unit flow along the dotted path costs -1 . The sum of these two unit flows has a cost of -4 and is optimal.

A shortest path tree in the residual network is shown in Figure 5, and the corresponding

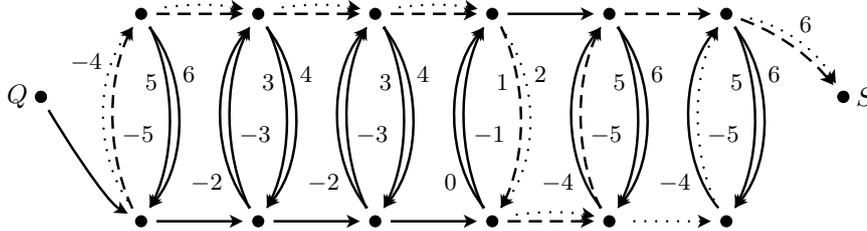


Figure 4: The network for the example.

approximation is

$$\begin{aligned} B = (4 \ 3 \ 3 \ 2 \ 4 \ 4) &= (1 \ 0 \ 0 \ 0 \ 0 \ 0) + (1 \ 1 \ 1 \ 0 \ 0 \ 0) \\ &\quad + 2(1 \ 1 \ 1 \ 1 \ 1 \ 1) + 2(0 \ 0 \ 0 \ 0 \ 1 \ 1) \end{aligned}$$

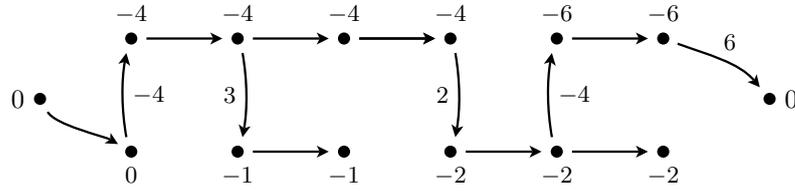


Figure 5: A shortest path tree in the residual network.

We conclude this section with a quick complexity analysis. We have reduced the total change optimal approximation of a matrix of size $m \times n$ to a minimum cost flow problem in a network with $2mn + 2$ nodes and $8mn - 2n$ arcs. Thus, according to [12, 13] the running time of the resulting algorithm is bounded by $O((mn)^2 \log^2(mn))$. For comparison, without the interleaf collision constraint the approximation problem can be solved in time $O(mn^2)$ if the differences $\bar{a}_{ij} - \underline{a}_{ij}$ are bounded [5].

4 Test results

We did some computational experiments with a C++-implementation of our algorithm. We did not implement the minimum cost flow algorithm with the theoretically optimal complexity bound. Instead we used the implementation of a primal network simplex method from [11].

We generated matrices of sizes 15×15 and 30×30 with random entries from $\{0, 1, \dots, L\}$ (independent, uniformly distributed) for $L \in \{8, 12, 16\}$. The lower and upper bounds were chosen such that a maximal change of ± 2 is possible for each entry, in other words, we put

$$\underline{a}_{ij} = \max\{0, a_{ij} - 2\}, \quad \bar{a}_{ij} = a_{ij} + 2$$

for all $(i, j) \in [m] \times [n]$. The results are shown in Table 1, where we averaged over 1000

L	$m = n = 15$					$m = n = 30$				
	DT_1	DT_2	TC_1	TC_2	time	DT_1	DT_2	TC_1	TC_2	time
8	35.7	14.5	188.7	165.3	2	67.7	24.5	837.2	713.9	11
12	51.9	29.3	140.8	125.8	2	97.9	51.4	651.3	559.5	15
16	67.6	44.3	112.8	102.0	3	127.8	79.9	505.4	430.7	17

Table 1: Test results for random matrices.

matrices for each triple (m, n, L) . Columns ‘ DT_1 ’ and ‘ DT_2 ’ contain the average decomposition times for the exact and the approximated decomposition, respectively. For the minimal possible delivery time \tilde{c} , we computed the total change using the heuristic approach from [9] (column ‘ $TC1$ ’) and with our exact method (column ‘ $TC2$ ’). The final

column ‘time’ contains the computation time (in seconds) for the approximated decomposition of 1000 matrices on a 3GHz workstation with 16GB RAM. We also tested our algorithm for two sets of practical matrices that were used in [2] and [4] and can be found online [6]. The first set contained 20 matrices of size 20×20 with entries from $\{0, 1, \dots, 15\}$, and the second one consisted of 20 matrices of size 40×40 with entries from $\{0, 1, \dots, 10\}$. The averaged results are shown in Table 2, where the computation time for the whole table

set	DT_1	DT_2	TC_1	TC_2
1	83.6	51.0	227.0	204.4
2	108.9	47.9	1387.4	1180.7

Table 2: Test results for real-world matrices.

was less than a second. Figure 6 illustrates the tradeoff between delivery time and total

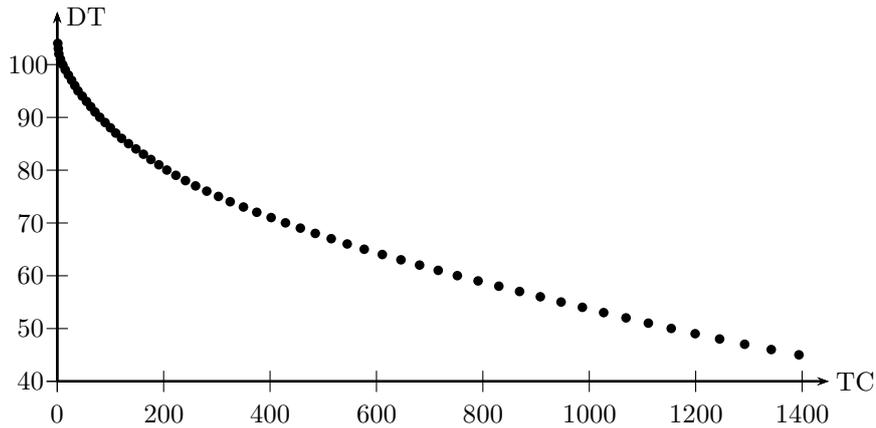


Figure 6: The tradeoff between delivery time and total change.

change for one of the 40×40 -matrices from [6].

5 Summary and discussion

We formulated the approximated MLC shape matrix decomposition with minimal total change as a minimum cost flow problem. This formulation allows us to include the interleaf collision constraint into the model. We demonstrated that this problem can be solved very efficiently using a standard implementation of the network simplex algorithm.

We want to conclude the paper with a short discussion of the relevance of our approximation approach. In some sense, the shape matrix decomposition problem could be considered as solved, since there are many efficient algorithms, even including additional technological constraints. But of course, there is room for improvement. We suggest the following two problems that might arise from a practical point of view.

1. What happens if the delivery time for a leaf sequence obtained by some exact algorithm is considered to be too large for clinical practice?
2. There are certain dosimetric effects of small or narrow fields which are not captured in the mathematical model underlying the standard algorithms. This could lead to significant differences between the planned and the delivered fluence.

We think that the “smoothing” of the fluence that is achieved by our approximation has a positive effect in both of these contexts. It has already been shown that the delivery time can be reduced considerably. Intuitively, the approximation also reduces the number of necessary small shapes with bad dosimetric properties. This deserves further investigations.

References

- [1] R.K. Ahuja, T.L. Magnanti and J.B. Orlin. *Network flows*. Englewood Cliffs, NJ: Prentice Hall, 1993.
- [2] D. Baatar, N. Boland, S. Brand and P. Stuckey. “Minimum cardinality matrix decomposition into consecutive-ones matrices: CP and IP approaches”. In: *Proc. 4th CPAIOR 2007*. Ed. by P. Van Hentenryck and L. Wolsey. Vol. 4510. LNCS. Springer, 2007, pp. 1–15. DOI: 10.1007/978-3-540-72397-4.
- [3] D. Baatar, H.W. Hamacher, M. Ehrgott and G.J. Woeginger. “Decomposition of integer matrices and multileaf collimator sequencing”. In: *Discr. Appl. Math.* 152.1-3 (2005), pp. 6–34. DOI: 10.1016/j.dam.2005.04.008.
- [4] H. Cambazard, E. O’Mahony and B. O’Sullivan. “A shortest path-based approach to the multileaf collimator sequencing problem”. In: *Proc. 6th CPAIOR 2009*. Ed. by W.-J. van Hoesve and J.N. Hooker. Vol. 5574. LNCS. Springer, 2009, pp. 41–55. DOI: 10.1007/978-3-642-01929-6_5.
- [5] K. Engel and A. Kiesel. “Approximated matrix decomposition for IMRT planning with multileaf collimators”. In: *OR Spectrum* 33.1 (2011), pp. 149–172. DOI: 10.1007/s00291-009-0168-5.
- [6] A. Holder. *Intensity map repository*. online repository. <http://holderfamily.dot5hosting.com/aholder/oncology/software/IntensityMaps> (12 March 2013). 2009.
- [7] T. Kalinowski. “A duality based algorithm for multileaf collimator field segmentation with interleaf collision constraint”. In: *Discr. Appl. Math.* 152.1-3 (2005), pp. 52–88. DOI: 10.1016/j.dam.2004.10.008.
- [8] T. Kalinowski. “Realization of intensity modulated radiation fields using multileaf collimators”. In: *Information Transfer and Combinatorics*. Ed. by R. Ahlswede *et al.* Vol. 4123. LNCS. Springer-Verlag, 2006, pp. 1010–1055. DOI: 10.1007/11889342_65.
- [9] T. Kalinowski and A. Kiesel. “Approximated MLC shape matrix decomposition with interleaf collision constraint”. In: *Algorithmic Operations Research* 4.1 (2009), pp. 49–57.

-
- [10] S. Kamath, S. Sahni, J. Li, J. Palta and S. Ranka. “Leaf sequencing algorithms for segmented multileaf collimation”. In: *Phys. Med. Biol.* 48.3 (2003), pp. 307–324. DOI: 10.1088/0031-9155/48/3/303.
 - [11] A. Löbel. *MCF 1.3 – A network simplex implementation*. Available for academic use free of charge at <http://www.zib.de>. 2004.
 - [12] J.B. Orlin. “A faster strongly polynomial minimum cost flow algorithm”. In: *Proc. 20th ACM symposium on Theory of computing, STOC 1988*. ACM. 1988, pp. 377–387. DOI: 10.1145/62212.62249.
 - [13] J.B. Orlin. “A faster strongly polynomial minimum cost flow algorithm”. In: *Operations research* 41.2 (1993), pp. 338–350. DOI: 10.1287/opre.41.2.338.