Algorithmic Operations Research Vol.4 (2009) 49–57

Approximated MLC shape matrix decomposition with interleaf collision constraint

Antje Kiesel and Thomas Kalinowski

Institut für Mathematik, Universität Rostock, 18051 Rostock, Germany

Abstract

Shape matrix decomposition is a subproblem in radiation therapy planning. A given fluence matrix A has to be decomposed into a sum of shape matrices corresponding to homogeneous fields that can be shaped by a multileaf collimator (MLC). We solve the problem of minimizing the delivery time for an approximation of A satisfying certain prescribed bounds, under the additional condition that the used MLC requires the interleaf collision constraint.

Key words: Intensity modulated radiation therapy (IMRT); multileaf collimator; combinatorial optimization; programming involving graphs

1. Introduction

In modern cancer therapy radiation is used to destroy the tumor tissue. At the same time one has to minimize the damage to the healthy tissue, and in particular to sensible structures or organs at risk. Intensity modulated radiation therapy was introduced in order to improve the quality of radiation treatment. In clinical practice it is common to use a linear accelerator which can release radiation from different directions (Fig. 1). In addition, a multileaf collimator (MLC) (Fig. 2) can be used to protect certain parts of the irradiated area.

For the treatment planning, the first step is to determine a set of directions (typically 3–9), from which radiation is released, given by positions of the isocenter, table angles and gantry angles [5,13]. In a second step, for each direction the fluence distribution is optimized, subject to the required dose distribution in the target. The final step is to determine, for each fluence distribution, a corresponding sequence of MLC leaf positions. Recently, there have been attempts to formulate the optimization problem more globally [5,14], but most of the widely used treatment planning systems model the three steps independently. In this paper we consider the last step for the MLC in the so called *step-and-shoot* mode. This means the radiation is switched off while the leaves are moving, and so the generated intensity modulated field is just a superposition of finitely many homogeneous fields which are shaped by the MLC. The two most important objectives in the optimization problem are the total irradiation time, or delivery time (DT), and the number of used fields, or decomposition cardinality (DC). Starting with [2] and [6] there have been proposed several algorithms for this problem [3,10,15,16], taking into account additional machine dependent constraints as the interleaf collision constraint [1,7] or the tongue-and-groove constraint [11] (see [8] or [9] for a survey).

All of these algorithms start with the given fluence matrix A and construct a sequence of leaf positions realizing this matrix. But from a practical point of view there seem to be some doubts if it is reasonable to consider every entry a_{ij} as fixed once and for all. First, the matrix A is a result of numerical computations which are based on simplified physical models of how the radiation passes through the patients body, and second, the representation of A as a superposition of homogeneous fields is also based on model assumptions which are not strictly correct, for instance the dose delivered to an exposed bixel depends on the shape of the field. So it might be sufficient, to realize (in our model) a matrix that is close to A. It is a natural question, how much the delivery time can be reduced by giving only an approximate representation of A satisfying certain minimum and maximum dose constraints. As an immediate consequence, the next problem arises: find an approximation with this optimal DT which is as close as possible to A. These questions have been answered for unconstrained MLCs in [4,12], and in the present paper

Email: Antje Kiesel [antje.kiesel@uni-rostock.de], Thomas Kalinowski [thomas.kalinowski@uni-rostock.de].

Fig. 1. A linear accelerator.

Fig. 2. Leaf pairs of a multileaf collimator.

we generalize the ideas from these references to MLCs with interleaf collision constraint.

In Section 2. we give a precise statement of the problem, Section 3. reviews an exact algorithm for shape matrix decomposition with interleaf collision constraint, in Section 4. we present our graph-theoretical characterization of the minimal DT of an approximation with a constructive proof, in Section 5. we show how the total change can be reduced heuristically, and the final Section 6. contains some test results.

2. Notation and problem formulation

Throughout the rest of the paper, for a natural number n, $[n]$ denotes the set $\{1, 2, \ldots, n\}$ and for integers $m < n$, $[m, n]$ denotes the set $\{m, m+1, \ldots, n\}$. For integers a, we also use the notation a_+ for the nonnegative part, defined by

$$a_+ = \begin{cases} a & \text{if } a \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

Our starting point is an $m \times n-$matrix A with nonnegative integer entries. The entry a_{ij} represents the desired fluence at bixel (i, j). In addition, for each entry (i, j) we have lower and upper bounds $\underline{a_{ij}}$ and $\overline{a_{ij}}$, such that

$$0 \leq \underline{a_{ij}} \leq a_{ij} \leq \overline{a_{ij}}.$$

Definition 1.[Feasible Approximation] Any integer matrix B with

$$\underline{a_{ij}} \leq b_{ij} \leq \overline{a_{ij}}$$

is called a *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) $(i = 1, \ldots, m)$, 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 in Definition 2 asserts that, in each row, there is exactly one (possibly empty) interval receiving radiation, while the rest of the row is covered either by the left or by the right leaf. The second condition is called interleaf collision constraint (ICC). It ensures that 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, for instance the Elekta MLC. 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 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*

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

we have $DT = 4$.

Now we formulate three optimization problems.

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-MinDT-TC. 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, 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 [1,7,10]. The idea underlying one of these algorithms is reviewed in the next section because it is the basis for our approach to the second problem **Approx-MinDT**. Finally, we observe that the second part of each of the problems **Approx-MinDt** and **Approx-MinDT-TC**, the search for the shape matrix decomposition, can be ignored safely, because, once the matrix B is fixed, we can apply any exact decomposition algorithm to complete the task.

3. Review of the exact decomposition

The basis of our approach is a characterization of the minimal DT of a decomposition with ICC as the maximal weight of a $q-s-$path in the following digraph $G = (V, E)$ [7,9].

$$\begin{aligned} V &= \{q, s\} \cup [m] \times [0, n+1], \\ E &= \{(q, (i, 0)) : i \in [m]\} \cup \{((i, n+1), s) : i \in [m]\} \\ &\quad \cup \{((i, j), (i, j+1)) \ : \ i \in [m],\ j \in [0, n]\} \\ &\quad \cup \{((i, j), (i+1, j)) \ : \ i \in [m-1],\ j \in [n]\} \\ &\quad \cup \{((i, j), (i-1, j)) \ : \ i \in [2, m],\ j \in [n]\}. \end{aligned}$$

In order to avoid case distinctions, we add two columns to our matrix and put

$$a_{i0} = a_{i,n+1} = 0 \qquad (i \in [m]).$$

Now we can define arc weights by

$$\begin{aligned} w(q, (i, 0)) &= w((i, n+1), s) = 0 \ (i \in [m]) \\ w((i, j-1), (i, j)) &= (a_{ij} - a_{i,j-1})_+ \\ &\qquad (i \in [m],\ j \in [n+1]) \\ w((i, j), (i+1, j)) &= -a_{ij} \ (i \in [m-1],\ j \in [n]) \\ w((i, j), (i-1, j)) &= -a_{ij} \ (i \in [2, m],\ j \in [n]). \end{aligned}$$

We call this graph the *DT-ICC-graph* for A. Fig. 3 shows the DT-ICC-graph for the matrix

$$A = \begin{pmatrix} 4 & 5 & 0 & 1 & 4 & 5 \\ 2 & 4 & 1 & 3 & 1 & 4 \\ 2 & 3 & 2 & 1 & 2 & 4 \\ 5 & 3 & 3 & 2 & 5 & 3 \end{pmatrix}.$$

Definition 4. Let A be an intensity matrix, and let G be the DT-ICC-graph for A. The maximal weight of a $q - s-$path in G is called *ICC-complexity* of A and denoted by $c(A)$. More formally,

$$c(A) = \max\{w(P) \ : \ P \text{ is a } q - s - \text{path in } G.\}.$$

Using this definition the main result of [7] can be formulated as follows.

Theorem 1 *The minimal DT of a decomposition of A with ICC equals $c(A)$.*

4. Approximation

To simplify our notation, for each $(i, j) \in [m] \times [n]$ we introduce the interval of acceptable fluence values

$$I_{ij} = \left[\underline{a_{ij}}, \overline{a_{ij}}\right], \qquad \underline{a_{ij}} \le a_{ij} \le \overline{a_{ij}}.$$

We want to find a matrix B such that

$$b_{ij} \in I_{ij} \ \text{ for } (i, j) \in [m] \times [n] \quad \text{ and } \quad c(B) \to \min.$$

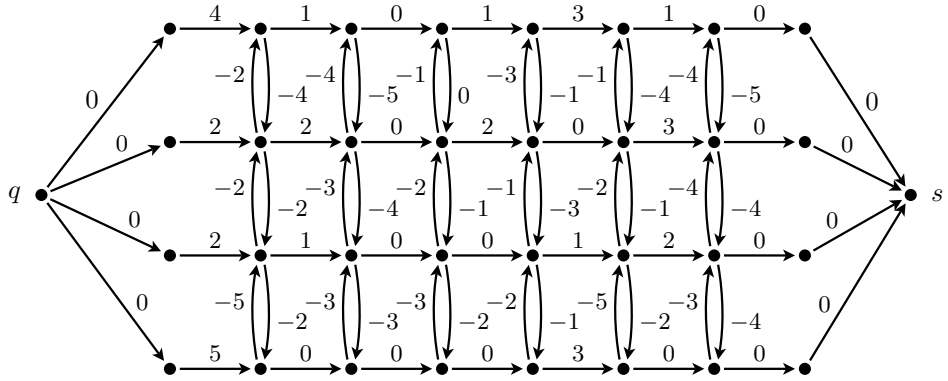

Fig. 3. The DT-ICC-graph for matrix A.

We follow an approach from [12] and replace every vertex $(i,j) \in [m] \times [n]$ by $|I_{ij}|$ copies, i.e. by the set

$$V_{ij} = \{(i,j)\} \times I_{ij}.$$

In order to avoid case distinctions in the discussion below we also replace the vertices in columns 0 and $n+1$ by

$$V_{i0} = \{(i,0,0)\} \quad \text{and} \quad V_{i,n+1} = \{(i,n+1,0)\}.$$

An arc $((i,j),(i,j+1))$ in the DT-ICC-graph G is replaced by the complete bipartite graph $V_{ij} \times V_{i,j+1}$, and similarly for the arcs $((i,j),(i \pm 1,j))$. The weights of the arcs $((i,j,k),(i,j+1,l))$ should model the approximation matrix B if we choose $b_{ij} = k$ and $b_{i,j+1} = l$, and similarly for the other arc types. Hence we define the arc weights by

$$\begin{aligned}
w(q,(i,0,0)) &= 0 \ (i \in [m]),\\
w((i,n+1,0),s) &= 0 \ (i \in [m]),\\
w((i,0,0),(i,1,k)) &= k \ (i \in [m], k \in I_{i1}),\\
w((i,n,k),(i,n+1,0)) &= 0 \ (i \in [m], k \in I_{in}),\\
w((i,j-1,k),(i,j,l)) &= (l-k)_+ \ (i \in [m], j \in [n],\\
&\qquad k \in I_{i,j-1}, l \in I_{ij}),\\
w((i,j,k),(i+1,j,l)) &= -k \ (i \in [m-1], j \in [n],\\
&\qquad k \in I_{ij}, l \in I_{i+1,j}),\\
w((i,j,k),(i-1,j,l)) &= -k \ (i \in [2,m], j \in [n],\\
&\qquad k \in I_{ij}, l \in I_{i-1,j}).
\end{aligned}$$

In order to determine the minimal complexity of an approximation matrix we compute numbers $W(i,j,k)$ such that

$$\begin{aligned}
W(i,j,k) = \max\big\{ &\min_l W(i,j-1,l) + (k-l)_+,\\
&\min_l W(i-1,j,l) - l,\\
&\min_l W(i+1,j,l) - l\big\}.
\end{aligned}$$

The intuitive idea is that for every feasible approximation B with $b_{ij} = k$, the maximal weight of a q-(i,j)-path in the DT-ICC-graph for B is at least $W(i,j,k)$. The numbers $W(i,j,k)$ can be computed efficiently (complexity $O(m^2 n \Delta^2)$, where Δ denotes any upper bound for $|I_{ij}|$) as described in Algorithm 1. Again, in order to avoid case distinctions at the boundaries, we add the values

$$W(0,j,0) = W(m+1,j,0) = a_{0j} = a_{m+1,j} = 0 \qquad (j \in [n]).$$

By construction, for any feasible approximation B with $b_{in} = k$, the DT-ICC-graph for B contains a path of weight at least $W(i,n,k)$. Hence the numbers $W(i,n,k)$ can be used to define a lower bound $\tilde{c}(A)$ for the ICC-complexity of a feasible approximation of A.

Definition 5. The *ICC-approximation complexity* of A (with respect to the given intervals I_{ij}) is defined by

$$\tilde{c}(A) = \max_i \min_k W(i,n,k).$$

We will show that this bound is sharp by an explicit construction of an approximation matrix B with this ICC-complexity. For the last column we put

$$b_{in} = \begin{cases} a_{in} & \text{if } W(i,n,a_{in}) \le \tilde{c}(A),\\ \max\{k \ : \ W(i,n,k) \le \tilde{c}(A)\} & \text{otherwise.} \end{cases}$$

Algorithm 1 (Computation of the numbers $W(i,j,k)$)
for $i \in [m]$ ***do*** $W(i,0,0) = 0$
for $j = 1$ ***to*** n ***do***
  for $i \in [m]$ ***do***
    for all k ***do***
      $W(i,j,k) = \min_l W(i,j-1,l) + (k-l)_+$
  for $i = 2$ ***to*** m ***do***
    for all k ***do***
      $W(i,j,k) = \max\{W(i,j,k), \min_l W(i-1,j,l) - l\}$
    for $i' = i-1$ ***downto*** *1* ***do***
      for all k ***do***
        $W(i',j,k) = \max\{W(i',j,k), \min_l W(i'+1,j,l) - l\}$

For $j < n$, we assume that the entries $b_{i,j+1}$ are already determined, and put

$$b_{ij} = \max\{k \ : W(i,j,k) + (b_{i,j+1} - k)_+ \\ \leq W(i,j+1,b_{i,j+1})\}.$$

Example 2 *We consider the following fluence matrix A with $c(A) = 8$.*

$$A = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 0 & 4 \end{pmatrix}$$

We choose the upper and lower bound such that $|b_{ij} - a_{ij}| \leq 1$ for every (i,j). The intervals and an optimal approximation are

$$\begin{pmatrix} [3,5] & [0,1] & [0,1] \\ [0,1] & [0,1] & [3,5] \end{pmatrix}, \qquad B = \begin{pmatrix} 3 & 1 & 0 \\ 1 & 1 & 3 \end{pmatrix}$$

with $c(B) = 4$, realized by the optimal decomposition

$$\begin{pmatrix} 3 & 1 & 0 \\ 1 & 1 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Our algorithm obtains matrix B as follows. First we compute the numbers $W(i,j,k)$, and obtain, for each (i,j), a vector

$$\left(W_{i,j,\underline{a_{ij}}}, W_{i,j,\underline{a_{ij}}+1}, \ldots, W_{i,j,\overline{a_{ij}}}\right).$$

These vectors are collected in the following array.

$$\begin{matrix} (3,4,5) & (3,3) & (3,3) \\ (0,1) & (2,2) & (4,5,6). \end{matrix}$$

Thus the optimal DT is

$$\max\{\min\{3,3\}, \min\{4,5,6\}\} = 4.$$

For the third column we choose $b_{13} = 0$ and $b_{23} = 3$. For the entry $(1,2)$ we have

$$\begin{aligned} W(1,2,0) + w((1,2,0),(1,3,0)) \\ &= W(1,2,1) + w((1,2,1),(1,3,0)) \\ &= W(1,3,0). \end{aligned}$$

We choose the maximal possible value $b_{12} = 1$. Observe that $b_{12} = 0$ is indeed not possible, since it leads to an increased DT. For entry $(2,2)$ we have

$$W(2,2,0) + w((2,2,0),(2,3,3)) = 2+3 > W(2,3,3),$$

so here $b_{22} = 1$ is the only possible choice. Similarly, we get $b_{11} = 3$ and $b_{21} = 1$. Clearly, the latter one can be replaced by 0.

In order to prove that our method is correct, we need some simple properties of the numbers $W(i,j,k)$.

Lemma 2 *For every $(i,j) \in [m] \times [n]$ and every k such that $(i,j,k), (i,j,k+1) \in V_{ij}$ we have*

$$W(i,j,k) \leq W(i,j,k+1) \leq W(i,j,k) + 1. \quad (1)$$

Furthermore, $W(i,j,k+1) = W(i,j,k) + 1$ iff

$$W(i,j,k) = W(i,j-1,l) + (k-l)_+$$

for some $l \in I_{i,j-1}$ with $l \leq k$.

Proof. Since

$$W(i,j-1,l) + (k-l)_+ \leq W(i,j-1,l) - (k+1-l)_+$$

and using the definition of the $W(i,j,k)$, we conclude $W(i,j,k) \leq W(i,j,k+1)$. On the other hand, we have

$$\begin{aligned}W(i,j,k) = \max\big\{ &\min_l W(i,j-1,l) + (k-l)_+,\\ &\min_l W(i-1,j,l) - l,\\ &\min_l W(i+1,j,l) - l\big\}\\ \geq \max\big\{ &\min_l W(i,j-1,l) + (k+1-l)_+,\\ &\min_l W(i-1,j,l) - l,\\ &\min_l W(i+1,j,l) - l\big\} - 1\\ = W(i,j,k+1) - 1,&\end{aligned}$$

where equality occurs iff $W(i,j,k) = W(i,j-1,l) + (k-l)_+$ and $k \geq l$. ■

The next lemma is the key step of our argument. It asserts that the chosen b_{ij} do not lead to conflicts inside the columns.

Lemma 3 *For all j and all $i \in [m-1]$, we have*

$$W(i,j,b_{ij}) - b_{ij} \leq W(i+1,j,b_{i+1,j}),$$

and for all j and all $i \in [2,m]$, we have

$$W(i,j,b_{ij}) - b_{ij} \leq W(i-1,j,b_{i-1,j}).$$

Proof. We only show the first statement, since the second one can be proved similarly. Suppose the statement is false, i.e.

$$W(i,j,b_{ij}) - b_{ij} > W(i+1,j,b_{i+1,j}).$$

By construction, there is some $k \in I_{ij}$ such that $W(i,j,k) - k \leq W(i+1,j,b_{i+1,j})$.

Case 1. $k < b_{ij}$. Let $\delta = b_{ij} - k > 0$. By Lemma 2 we have

$$W(i,j,k) \geq W(i,j,b_{ij}) - \delta.$$

But now we obtain

$$\begin{aligned}W(i,j,k) - k &\geq (W(i,j,b_{ij}) - \delta) - (b_{ij} - \delta)\\ &> W(i+1,j,b_{i+1,j}),\end{aligned}$$

and this is the required contradiction.

Case 2. $k > b_{ij}$. Let $\delta = k - b_{ij} > 0$. By construction of the numbers b_{ij},

$$\begin{aligned}W(i,j,b_{ij}) + (b_{i,j+1} - b_{ij})_+ &\leq W(i,j+1,b_{i,j+1}),\\ W(i,j,b_{ij}+1) + (b_{i,j+1} - (b_{ij}+1))_+ &> W(i,j+1,b_{i,j+1}).\end{aligned}$$

Using Lemma 1, this is possible only if

$$W(i,j,b_{ij}+1) = W(i,j,b_{ij}) + 1.$$

Using Lemma 1 repeatedly, we obtain

$$W(i,j,k) = W(i,j,b_{ij}) + \delta.$$

But together this implies

$$W(i,j,k) - k = W(i,j,b_{ij}) - b_{ij},$$

which is a contradiction. ■

Now let G be the DT-ICC-graph for B. Denote by $\alpha_1(i,j)$ the maximal weight of a $q-(i,j)-$path in G. Note that the numbers $\alpha_1(i,j)$ can be computed similarly to the numbers $W(i,j,k)$. Clearly, $\alpha_1(i,1) = b_{i1}$, and the procedure for column $j > 1$ is described in Algorithm 2.

Lemma 4 *For all (i,j) we have $\alpha_1(i,j) \leq W(i,j,b_{ij})$.*

Proof. We use induction on j. For $j = 1$ the claim is obvious:

$$\alpha_1(i,1) = W(i,1,b_{i1}) = b_{i1}.$$

Now let $j > 1$. After the initialization of the numbers $\alpha_1(i,j)$ in the first loop of Algorithm 2 we obtain for every i,

$$\begin{aligned}\alpha_1(i,j) &= \alpha_1(i,j-1) + (b_{ij} - b_{i,j-1})_+\\ &\leq W(i,j-1,b_{i,j-1}) + (b_{ij} - b_{i,j-1})_+\\ &\leq W(i,j,b_{ij}).\end{aligned}$$

We just have to check that this inequalities remain valid in every updating step. Suppose the first violation occurs when we replace $\alpha_1(i,j)$ by $\alpha_1(i \pm 1,j) - b_{i\pm1,j}$. In this case,

$$\begin{aligned}\alpha_1(i,j) &= \alpha_1(i \pm 1,j) - b_{i\pm1,j}\\ &\leq W(i \pm 1,j,b_{i\pm1,j}) - b_{i\pm1,j}\\ &\leq W(i,j,b_{ij}),\end{aligned}$$

where the last inequality is Lemma 3. So the statement of the lemma remains valid. ■

By Lemma 4 (and Theorem 1), matrix B allows a decomposition with $DT \leq \tilde{c}(A)$ and this implies the following theorem.

Theorem 5 *The minimal DT of a decomposition of a feasible approximation of A equals $\tilde{c}(A)$ and an approximation matrix B realizing this DT can be constructed as described above in time $O(m^2 n \Delta^2)$.*

Algorithm 2 (Computation of the numbers $\alpha_1(i,j)$ for fixed $j \geq 2$)
for $i \in [m]$ ***do***
$\alpha_1(i,j) = \alpha_1(i,j-1) + (b_{ij} - b_{i,j-1})_+$
for $i = 2$ ***to*** m ***do***
$\alpha_1(i,j) = \max\{\alpha_1(i,j), \alpha_1(i-1,j) - b_{i-1,j}\}$
for $i' = i-1$ ***downto*** *1* ***do***
$\alpha_1(i',j) = \max\{\alpha_1(i',j), \alpha_1(i'+1,j) - b_{i'+1,j}\}$

Proof. The only thing that is left to prove is the complexity statement. For this it is sufficient to note that the computation of the numbers $W(i,j,k)$ dominates the computation time, since this has complexity $O(m^2n\Delta^2)$ as can be seen immediately from Algorithm 1. But after the numbers $W(i,j,k)$ have been computed we look at every entry (i,j) only once and in order to fix b_{ij} we have to do at most $|I_{ij}|$ comparisons. So the matrix B is determined in time $O(mn\Delta)$ and this concludes the proof. ■

5. Reducing the total change

The construction described in Section 4. leads to an approximation B with minimal delivery time, but a large total change $TC(B)$. The reason is, that we put

$$b_{ij} = \max\{k \ : \ W(i,j,k) + (b_{i,j+1} - k) \leq W(i,j+1,b_{i,j+1})\},$$

even if none of the vertices (i,j,k) is critical, i.e. part of a q-s-path of maximal weight in the DT-ICC-graph of a feasible approximation of A. Thus, the aim is to find an approximation with the same delivery time, but smaller total change. Clearly, we can replace b_{ij} by a value b'_{ij} with $b_{ij} < b'_{ij} \leq a_{ij}$ in the case $b_{ij} < a_{ij}$, respectively with $a_{ij} \geq b'_{ij} > b_{ij}$ in the case $a_{ij} > b_{ij}$, if this decision does not increase the maximal weight of a q-s-path in the DT-ICC-graph.

Let therefore G be the DT-ICC-graph of B and let $\alpha_1(i,j)$ denote the maximal weight of a q-(i,j)-path in G. Similarly, let $\alpha_2(i,j)$ denote the maximal weight of an (i,j)-s-path in G. The values $\alpha_2(i,j)$ can be computed similarly as the numbers $\alpha_1(i,j)$.

Definition 6. Let B be a feasible approximation of A. For $(i,j) \in [m] \times [n]$, an integer b is called (i,j)*−feasible* (with respect to B) if the following conditions are satisfied.

(1) $b \in I_{ij}$.
(2) $\alpha_1(i,j-1) + (b - b_{i,j-1})_+ + (b_{i,j+1} - b)_+ + \alpha_2(i,j+1) \leq \tilde{c}(A)$.
(3) $i = 1$ or $\alpha_1(i,j-1) + (b - b_{i,j-1})_+ - b + \alpha_2(i-1,j) \leq \tilde{c}(A)$.
(4) $i = m$ or $\alpha_1(i,j-1) + (b - b_{i,j-1})_+ - b + \alpha_2(i+1,j) \leq \tilde{c}(A)$.
(5) $i = 1$ or $\alpha_1(i-1,j) - b_{i-1,j} + (b_{i,j+1} - b)_+ + \alpha_2(i,j+1) \leq \tilde{c}(A)$.
(6) $i = m$ or $\alpha_1(i+1,j) - b_{i+1,j} + (b_{i,j+1} - b)_+ + \alpha_2(i,j+1) \leq \tilde{c}(A)$.
(7) $i \in \{1,m\}$ or $\alpha_1(i-1,j) - b_{i-1,j} - b + \alpha_2(i+1,j) \leq \tilde{c}(A)$.
(8) $i \in \{1,m\}$ or $\alpha_1(i+1,j) - b_{i+1,j} - b + \alpha_2(i-1,j) \leq \tilde{c}(A)$.

In other words, b is $(i,j)-$feasible iff we can replace b_{ij} by b without destroying the $DT-$optimality of B. Fig 4 illustrates the different possibilities for a path to pass through vertex (i,j). Each of these possibilities corresponds to one of the conditions 2 through 8 in Definition 6.

We propose a heuristic, formally described in Algorithm 3, to reduce the total change. Clearly, the application of this algorithm can be iterated until no more changes occur.

6. Test Results

In this section we demonstrate the DT-reduction obtained by the methods from Section 4. and the total change reduction using the heuristic approach from Section 5.. We use matrices of size 15×15 and 30×30 with random entries $a_{ij} \in \{0,1,\ldots,L\}$ for $L \in \{8,12,16\}$. In our tests we choose the upper and lower bounds for the entries such that each entry is changed by at most 2, i.e. we put

$$\underline{a_{ij}} = (a_{ij} - 2)_+, \qquad \overline{a_{ij}} = a_{ij} + 2.$$

For each L, we construct decompositions of 1000 matrices, and compute the average minimal delivery time $\tilde{c}(A)$ and the total change according to our algorithm from Section 4.. Finally, we analyze the total change reduction, that can be achieved using Algorithm 3. The results are shown in Table 2 and 3. For comparison we

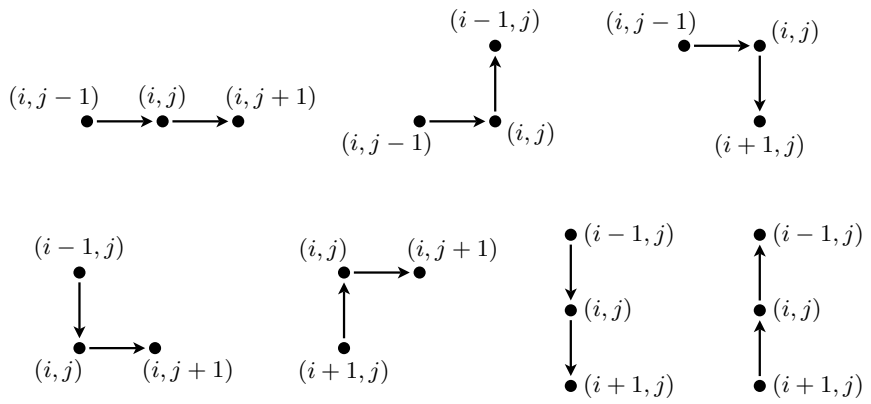

Fig. 4. The seven different types of paths that are affected by the choice of b_{ij}.

Algorithm 3 (Heuristic for total change minimization)
for $j = 1$ ***to*** n ***do***
  for $i = 1$ ***to*** m ***do***
    if $b_{ij} < a_{ij}$ ***and*** $b_{ij} + 1$ *is* (i,j)*−feasible* ***then*** $b_{ij}++$
    if $b_{ij} > a_{ij}$ ***and*** $b_{ij} - 1$ *is* (i,j)*−feasible* ***then*** $b_{ij}--$
    Update the numbers $\alpha_1(k,l)$ *and* $\alpha_2(k,l)$

include the minimal DT for exact decomposition with ICC [7]. Columns 'DT_1' and 'DT_2' contain the average delivery times for the exact and for the approximated decomposition, respectively. Columns 'TC_1' and 'TC_2' contain the total change values before and after the application of Algorithm 3. Our algorithms are completely

Table 1

L	DT_1	DT_2	TC_1	TC_2
8	35.7	14.6	329.1	188.7
12	51.8	29.2	358.3	140.8
16	67.7	44.6	373.9	112.8

Test results for $m = n = 15$.

L	DT_1	DT_2	TC_1	TC_2
8	67.7	24.5	1360.0	837.2
12	97.9	51.4	1484.3	651.3
16	127.7	79.9	1546.2	505.4

Test results for $m = n = 30$.

practicable. On a 3GHz workstation, the computations for the last row, i.e. for the decomposition of 1000 matrices of size 15×15 with entries from $\{0, 1, \ldots, 16\}$ took only 5 seconds for $m = n = 15$ and less than a minute for $m = n = 30$. Basically, we can draw two conclusions from our results.

(1) The approximation approach leads to a significant DT-reduction: for $L = 16$, allowing a change of at most 2 for each entry reduces the DT by more than 30%.
(2) Our heuristic leads to a large total change reduction: for $L = 16$ the total change can be reduced by almost 60%.

7. Summary and discussion

We presented an efficient method to minimize exactly the decomposition time in approximated MLC shape matrix decomposition with interleaf collision constraint. We also described a heuristic for reducing the total approximation error, and demonstrated the proposed algorithms on randomly generated matrices. The obvious next problem, which is the subject of ongoing research, is to find an exact algorithm for the minimization of the total change.

References

[1] D. Baatar, H.W. Hamacher, M. Ehrgott, and G.J. Woeginger. Decomposition of integer matrices and multileaf collimator sequencing. *Discrete Appl. Math.*, 152(1-3):6–34, 2005.

[2] T.R. Bortfeld, D.L. Kahler, T.J. Waldron, and A.L. Boyer. X–ray field compensation with multileaf collimators. *Int. J. Radiat. Oncol. Biol. Phys.*, 28:723–730, 1994.

[3] K. Engel. A new algorithm for optimal multileaf collimator field segmentation. *Discrete Appl. Math.*, 152(1-3):35–51, 2005.

[4] K. Engel and A. Kiesel. Approximated matrix decomposition for IMRT planning with multileaf collimators. To appear in *OR Spectrum*, 2008.

[5] K. Engel and E. Tabbert. Fast simultaneous angle, wedge, and beam intensity optimization in inverse radiotherapy planning. *Optimization and Engineering*, 6(4):393–419, 2005.

[6] J.M. Galvin, X.G. Chen, and R.M. Smith. Combining multileaf fields to modulate fluence distributions. *Int. J. Radiat. Oncol. Biol. Phys.*, 27:697–705, 1993.

[7] T. Kalinowski. A duality based algorithm for multileaf collimator field segmentation with interleaf collision constraint. *Discrete Appl. Math.*, 152(1-3):52–88, 2005.

[8] T. Kalinowski. Realization of intensity modulated radiation fields using multileaf collimators. In R. Ahlswede et al., editor, *Information Transfer and Combinatorics*, volume 4123 of *LNCS*, pages 1010–1055. Springer-Verlag, 2006.

[9] T. Kalinowski. Multileaf collimator shape matrix decomposition. In G.J. Lim, editor, *Optimization in Medicine and Biology*, pages 249–282. Auerbach Publishers Inc., 2008.

[10] S. Kamath, S. Sahni, J. Li, J. Palta, and S. Ranka. Leaf sequencing algorithms for segmented multileaf collimation. *Phys. Med. Biol.*, 48(3):307–324, 2003.

[11] S. Kamath, S. Sartaj, J. Palta, S. Ranka, and J. Li. Optimal leaf sequencing with elimination of tongue–and–groove underdosage. *Phys. Med. Biol.*, 49:N7–N19, 2004.

[12] A. Kiesel. Approximated matrix decomposition for IMRT planning with multileaf collimators. Master's thesis, Universität Rostock, 2007.

[13] J. Lim, M.C. Ferris, S.J. Wright, D.M. Shepard, and M.A. Earl. An optimization framework for conformal radiation treatment planning. *INFORMS Journal on Computing*, 19(3):366–380, 2007.

[14] H.E. Romeijn, R.K. Ahuja, J.F. Dempsey, and A. Kumar. A column generation approach to radiation therapy treatment planning using aperture modulation. *SIAM J. on Optimization*, 15(3):838–862, 2005.

[15] R.A.C. Siochi. Minimizing static intensity modulation delivery time using an intensity solid paradigm. *Int. J. Radiat. Oncol. Biol. Phys.*, 43:671–680, 1999.

[16] P. Xia and L. Verhey. Multileaf collimator leaf–sequencing algorithm for intensity modulated beams with multiple static segments. *Med. Phys.*, 25:1424–1434, 1998.

Received 26 May 08; revised 21 Nov 08; accepted 28 Nov 08