

Reducing the number of monitor units in multileaf collimator field segmentation

Thomas Kalinowski
Universität Rostock
Fachbereich Mathematik
D-18051 Rostock
Germany

thomas.kalinowski@mathematik.uni-rostock.de

August, 2004

Abstract

Multileaf collimators (MLCs) are the prevailing tool for the realization of radiation fields in intensity modulated radiation therapy (IMRT). One step in the treatment planning is to determine a set of leaf positions realizing a certain intensity modulated radiation field. In this paper we suggest two extensions in the use of the MLC that lead to considerable savings in terms of monitor units, thus potentially increasing the treatment quality. We test our method with random and with clinical sample matrices.

Key words: leaf sequencing, radiation therapy optimization, intensity modulation, multileaf collimator, IMRT

2000 MSC: 92C50, 90C90

1 Introduction

In recent years intensity modulated radiation therapy (IMRT) has become an important method in cancer therapy. The objective in the treatment planning is to irradiate the tumor as efficient as possible without damaging the organs near to it. A modern way to realize intensity modulated radiation fields is the usage of a multileaf collimator (MLC). An MLC consists of two opposite banks of metal leaves which can be shifted towards each other and so open or close certain parts of the irradiated area. In this paper we assume that the desired intensity is already determined. After discretization an intensity function can be considered as an $m \times n$ matrix A with nonnegative integer entries. We consider the usage of MLCs in the static mode (step and shoot). This means that the radiation is switched off when the leaves of the collimator are moving. In other words we have to determine a (finite) set of leaf positions with corresponding irradiation times such that the superposition of the homogeneous fields yields the given intensity matrix. The leaf positions can be described by certain $(0, 1)$ -matrices, called shape matrices or segments. Two important criteria for the quality of the segmentation are the total number of monitor units (TNMU) and the number of segments (NS) which should both be as small as possible. In the literature there are several leaf sequencing algorithms ([7, 3, 4, 13, 12, 11, 5, 2, 1, 9, 6]), some of them providing the optimal TNMU but a large NS, others reducing the NS heuristically at the price of an increased TNMU. The algorithms also differ in the extend to which they include additional machine-dependent constraints like the interleaf collision constraint. In principle both, TNMU and NS, can be optimized by integer programming [10]. But due to the NP-completeness of integer programming this is applicable only for small problem sizes. See [8] for a survey and a comparison of the different segmentation algorithms. In this paper we suggest two new forms of using MLCs. Both approaches have in common that the set of allowed segments is enlarged which makes it possible to realize a given intensity matrix with a smaller number of segments. In Section 2 we allow the MLC to be rotated about 90 between the delivery of two segments, and in Section 3 two MLCs are used whose directions of leaf movement are perpendicular to each other. In both cases we neglect machine-dependent constraints and the tongue and groove effect. As in [3] we assume that the irradiation time for each segment equals 1, i.e. we have to represent the intensity matrix as a sum of segments, and under these conditions we want to minimize the number of used segments.

2 Using the MLC in two directions

In this section we search for a realization of the given intensity matrix A in the following setup: the multileaf collimator has no interleaf collision constraint and can be rotated about 90. A *segment* is a $(0, 1)$ -matrix describing a leaf position in the sense that a 0-entry indicates a region that is covered by some leaf and a 1-entry indicates a region that receives radiation. So there are two different types of segments, called horizontal and vertical segments, according to the choice of the direction of leaf motion. This is made precise in the following definition.

Definition 1. A *horizontal segment* is an $m \times n$ -matrix $S = (s_{i,j})$, such that there exist integers l_i, r_i ($i \in [m]$) with the following properties:

$$l_i \leq r_i + 1 \quad (i \in [m]), \quad (1)$$

$$s_{i,j} = \begin{cases} 1 & \text{if } l_i \leq j \leq r_i \\ 0 & \text{otherwise} \end{cases} \quad (i \in [m], j \in [n]), \quad (2)$$

A *vertical segment* is an $m \times n$ -matrix $S = (s_{i,j})$, such that there exist integers l_j, r_j ($j \in [n]$) with the following properties:

$$l_j \leq r_j + 1 \quad (j \in [n]), \quad (3)$$

$$s_{i,j} = \begin{cases} 1 & \text{if } l_j \leq i \leq r_j \\ 0 & \text{otherwise} \end{cases} \quad (i \in [m], j \in [n]), \quad (4)$$

Now a *segmentation* of an intensity matrix A is a representation of A as a sum of segments.

Example 1. A segmentation with 4 horizontal and 2 vertical segments is given by

$$\begin{pmatrix} 1 & 4 & 2 & 5 \\ 1 & 3 & 3 & 2 \\ 1 & 3 & 2 & 5 \\ 6 & 4 & 6 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \\ + \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix}. \quad (5)$$

Observe that the minimal number of monitor units for this matrix would be 8 if the MLC was used in only one direction (horizontal or vertical) [3, 6, 9].

Obviously, in this setup the TNMU of the treatment corresponding to a segmentation is proportional to the number of segments. Now the segmentation problem is to find, for given A , a segmentation with the minimal TNMU.

In the Subsection 2.1 we derive a backtracking algorithm for the segmentation problem. Due to the this algorithm is applicable only for small problem sizes (10×10 -matrices with entries between 0 and 7), but in any case we obtain a lower bound for the TNMU by interrupting the backtracking after some time. In Subsection 2.2 we present a heuristic approach that finds a segmentation with a small TNMU. A comparison between the lower bound and heuristic results can be found in Section 4.

2.1 A lower bound

Theorem 1. *Let c_1, c_2 be nonnegative integers and put $c = c_1 + c_2$. Then a segmentation of A with c_1 horizontal and c_2 vertical segments exists iff $B := cJ - A$ (where J is the all-one-matrix of size $m \times n$) can be written as a sum of four nonnegative integer matrices $P = (p_{i,j})$, $Q = (q_{i,j})$, $R = (r_{i,j})$, $S = (s_{i,j})$ with the following properties.*

1. $p_{i,j} + q_{i,j} \leq c_1$, $r_{i,j} + s_{i,j} \leq c_2$ for $(i, j) \in [m] \times [n]$
2. $p_{i,j} \geq p_{i,j+1}$, $q_{i,j} \leq q_{i,j+1}$ for $i \in [m]$, $j \in [n-1]$
3. $r_{i,j} \geq r_{i+1,j}$, $s_{i,j} \leq s_{i+1,j}$ for $i \in [m-1]$, $j \in [n]$

Proof. “ \Rightarrow ”: By construction, $b_{i,j}$ is the number of times the bixel (i, j) has to be covered in a segmentation with c monitor units. Suppose there is given a segmentation with c_1 horizontal and c_2 vertical segments. For $(i, j) \in [m] \times [n]$, let $p_{i,j}$, $q_{i,j}$, $r_{i,j}$ and $s_{i,j}$ be the number of segments in which bixel (i, j) is covered by the left, the right, the upper and the lower leaf, respectively. This yields the desired decomposition of B .

“ \Leftarrow ”: Suppose $B = P + Q + R + S$ where P , Q , R and S satisfy the conditions of the theorem. Now we define segments $S^{(1)}, S^{(2)}, \dots, S^{(c)}$ as follows. For $1 \leq k \leq c_1$, let

$$\sigma_{i,j}^{(k)} = \begin{cases} 0 & \text{if } k \leq p_{i,j}, \\ 1 & \text{otherwise,} \end{cases} \quad \tau_{i,j}^{(k)} = \begin{cases} 0 & \text{if } k > c_1 - q_{i,j}, \\ 1 & \text{otherwise.} \end{cases}$$

For $c_1 + 1 \leq k \leq c$, let

$$\sigma_{i,j}^{(k)} = \begin{cases} 0 & \text{if } k - c_1 \leq r_{i,j} \\ 1 & \text{otherwise} \end{cases}, \quad \tau_{i,j}^{(k)} = \begin{cases} 0 & \text{if } k - c_1 > c_2 - s_{i,j} \\ 1 & \text{otherwise} \end{cases}.$$

Finally, for all $k \in [c]$, put $s_{i,j}^{(k)} = \sigma_{i,j}^{(k)} \tau_{i,j}^{(k)}$. To conclude the proof we have to check that the matrices $S^{(k)} = (s_{i,j}^{(k)})$ are segments and that

their sum is A . Condition 1 in the theorem implies that, for each k and each (i, j) , at most one of the numbers $\sigma_{i,j}^{(k)}$ and $\tau_{i,j}^{(k)}$ is 0, corresponding to the fact that a bixel can not be covered by both leaves of a leaf pair at the same time. Thus the number of indices k with $s_{i,j}^{(k)} = 0$ equals $p_{i,j} + q_{i,j} + r_{i,j} + s_{i,j} = b_{i,j}$, hence

$$\sum_{k=1}^c s_{i,j}^{(k)} = c - b_{i,j} = a_{i,j}.$$

We show that $S^{(k)}$ is a horizontal segment for $k \leq c_1$. For $k > c_1$ one obtains similarly that $S^{(k)}$ is a vertical segment. If $s_{i,j}^{(k)} = 0$, either $\sigma_{i,j}^{(k)} = 0$ or $\tau_{i,j}^{(k)} = 0$. By construction of the $\sigma_{i,j}^{(k)}$, $\tau_{i,j}^{(k)}$ and by Condition 2 in the theorem this implies $\sigma_{i,j'}^{(k)} = 0$ for all $j' < j$ or $\tau_{i,j'}^{(k)} = 0$ for all $j' > j$, and consequently $s_{i,j'}^{(k)} = 0$ for all $j' < j$ or for all $j' > j$. \square

Example 2. For $A = \begin{pmatrix} 1 & 4 & 2 & 5 \\ 1 & 3 & 3 & 2 \\ 1 & 3 & 2 & 5 \\ 6 & 4 & 6 & 0 \end{pmatrix}$ and $c = 6 = 4 + 2$ we have

$$B = \begin{pmatrix} 5 & 2 & 4 & 1 \\ 5 & 3 & 3 & 4 \\ 5 & 3 & 4 & 1 \\ 0 & 2 & 0 & 6 \end{pmatrix} = \begin{pmatrix} 3 & 2 & 2 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \end{pmatrix} + \begin{pmatrix} 2 & 0 & 2 & 1 \\ 2 & 0 & 2 & 1 \\ 2 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 2 & 0 & 2 \end{pmatrix}.$$

This yields

$$\begin{array}{lll} \sigma^{(1)} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}, & \tau^{(1)} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}, & S^{(1)} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}, \\ \sigma^{(2)} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}, & \tau^{(2)} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}, & S^{(2)} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}, \\ \sigma^{(3)} = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}, & \tau^{(3)} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}, & S^{(3)} = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}, \\ \sigma^{(4)} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}, & \tau^{(4)} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}, & S^{(4)} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}, \\ \sigma^{(5)} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}, & \tau^{(5)} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}, & S^{(5)} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}, \\ \sigma^{(6)} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}, & \tau^{(6)} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix}, & S^{(6)} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix}. \end{array}$$

Based on Theorem 1, Algorithm 1 finds the minimal TNMU c and a decomposition of $B = cJ - A$ corresponding to a segmentation with this TNMU. The step in line 7 can be realized using backtracking. Of course this method is very time-consuming. Our implementation solved the problem

Algorithm 1 Minimal TNMU for 2-directional segmentation

```
 $c := \max\{a_{i,j} : (i,j) \in [m] \times [n]\}$ 
finished:=false
while not finished do
   $B := cJ - A$ 
5:   $c_1 := c; c_2 := 0$ 
  while not finished and  $c_1 \geq 0$  do
    Construct a decomposition of  $B$  as in Theorem 1 or derive a contradiction from the assumption that such a decomposition exists
    if Construction successful then
      finished:=true
10:  else
       $c_1 := c_1 - 1; c_2 := c_2 + 1$ 
    end if
  end while
  if not finished then
15:   $c := c + 1$ 
  end if
end while
```

for 10×10 -matrices with random entries between 0 and 7 in a few seconds, but for larger problems this algorithm is not practicable. By interrupting the backtracking when some fixed time limit is reached without getting a construction or a contradiction in line 7, we still obtain a lower bound for the TNMU.

2.2 A heuristic method

For notational convenience we add a 0-th and an $(m + 1)$ -th row and a 0-th and an $(n + 1)$ -th column with

$$a_{0,j} = a_{m+1,j} = a_{i,0} = a_{i,n+1} = 0 \quad (i = 1, \dots, m; j = 1, \dots, n),$$

and we put

$$\alpha(i) = \sum_{j=1}^n \max\{0, a_{i,j} - a_{i,j-1}\} \quad (i \in [m]),$$
$$\beta(j) = \sum_{i=1}^m \max\{0, a_{i,j} - a_{i-1,j}\} \quad (j \in [n]).$$

Let also

$$c_{hor}(A) = \max_{1 \leq i \leq m} \alpha(i), \quad c_{vert}(A) = \max_{1 \leq j \leq n} \beta(j).$$

According to [6] there is a segmentation without vertical segments with $c_{hor}(A)$ monitor units, and there is a segmentation without horizontal segments with $c_{vert}(A)$ monitor units. Hence an upper bound for the minimal number of monitor units needed to realize A is $c(A) = \min\{c_{hor}(A), c_{vert}(A)\}$.

Similar to the approach in [6] we construct a segmentation by successively subtracting segments until the zero matrix is reached. For the choice of the segment S we suggest a heuristic method. First $c_{hor}(A)$ and $c_{vert}(A)$ are determined and the direction of leaf motion is chosen to be horizontal if $c_{hor}(A) \leq c_{vert}(A)$ and vertical otherwise. We describe the construction of S for the horizontal case, the vertical case is treated analogously. For the new matrix $A' := A - S$, we have two aims. Firstly, we want that

$$c_{hor}(A') = c_{hor}(A) - 1, \tag{6}$$

and secondly $c_{vert}(A')$ should be small. Let us consider the second condition first. With the segments we associate (D, D') -paths in a layered digraph $G = (V, E)$ similar to the digraph from [2].

$$V = \{D, D'\} \cup \{(i, l, r) : i \in [m], 1 \leq l \leq r + 1 \leq n + 1, a_{i,j} > 0 \text{ for } l \leq j \leq r\},$$

and E consists of all possible arcs between row i and row $i + 1$ ($i = 1, \dots, m - 1$), all arcs between D and row 1, and all arcs between row m and D' . Clearly, there is a correspondence between the paths $D, (1, l_1, r_1), \dots, (m, l_m, r_m), D'$ and the (horizontal) segments. Now we define a weight function on E and determine the segment S as a path of maximal weight in Γ . This approach seems to be natural because the values $a'_{i,j} - a'_{i-1,j}$, and thus $c_{vert}(A')$ depend on the combinations of the leaf positions in adjacent rows. Experiments have shown that for the considered range of parameters the weight function w described below works quite well.

$$w(0, (1, l, r)) = \sum_{j=l}^r \max\{1, \beta(j) + 5 - c_{vert}(A)\} \quad (1 \leq l \leq r + 1 \leq n)$$

$$w((m, l, r), 1) = 0 \quad (1 \leq l \leq r + 1 \leq n + 1)$$

In order to define the weights for the arcs $((i - 1, l, r), (i, l', r'))$ we put, for $i = 2, 3, \dots, m$ and $j = 1, 2, \dots, n$,

$$\rho_1(i, j) = \begin{cases} \max\{1, \beta(j) + 5 - c_{vert}(A)\} & \text{if } a_{i,j} \geq a_{i-1,j} \\ 0 & \text{otherwise,} \end{cases}$$

$$\rho_2(i, j) = \begin{cases} \max\{1, \beta(j) + 5 - c_{vert}(A)\} & \text{if } a_{i,j} > a_{i-1,j} \\ 0 & \text{otherwise.} \end{cases}$$

$\rho_k(i, j)$ ($k = 1, 2$) measure the influence of a potential segment with $s_{i,j} \neq s_{i-1,j}$ on $a_{i,j} - a_{i-1,j}$ and thus on $\beta(j)$. $\rho_1(i, j)$ is nonzero if a segment with $s_{i,j} = 0$ and $s_{i-1,j} = 1$ increases the value of $\max\{0, a_{i,j} - a_{i-1,j}\}$, and similarly $\rho_2(i, j)$ is nonzero if a segment with $s_{i,j} = 1$ and $s_{i-1,j} = 0$ decreases $\max\{0, a_{i,j} - a_{i-1,j}\}$. The magnitudes of the $\rho_k(i, j)$ ($k = 1, 2$) are chosen according to the idea that columns j with $\beta(j)$ close to $c_{vert}(A)$ should be considered more important than columns j with a small value of $\beta(j)$. Now we define w as follows.

$$w((i-1, l, r), (i, l', r')) = \begin{cases} -\sum_{j=l}^{l'-1} \rho_1(i, j) - \sum_{j=r'+1}^r \rho_1(i, j) & \text{if } l \leq l' \leq r'+1 \leq r+1, \\ -\sum_{j=l}^{l'-1} \rho_1(i, j) + \sum_{j=r'+1}^{r'} \rho_2(i, j) & \text{if } l \leq l' \leq r+1 \leq r', \\ -\sum_{j=l}^r \rho_1(i, j) + \sum_{j=l'}^{r'} \rho_2(i, j) & \text{if } l \leq r+1 < l' \leq r'+1, \\ \sum_{j=l'}^{l-1} \rho_2(i, j) - \sum_{j=r'+1}^r \rho_1(i, j) & \text{if } l' < l \leq r'+1 \leq r+1, \\ \sum_{j=l'}^{l-1} \rho_2(i, j) + \sum_{j=r'+1}^{r'} \rho_2(i, j) & \text{if } l' < l \leq r+1 \leq r', \\ \sum_{j=l'}^{r'} \rho_2(i, j) - \sum_{j=l}^r \rho_1(i, j) & \text{if } l' \leq r'+1 < l \leq r+1. \end{cases}$$

Example 3. Let rows $i-1$ and i of matrix A be

$$\begin{pmatrix} 2 & 4 & 1 & 4 & 3 & 4 & 3 & 3 \\ 1 & 5 & 3 & 2 & 7 & 4 & 3 & 2 \end{pmatrix},$$

and consider the arc $e = ((i-1, 4, 7), (i, 2, 6))$, corresponding to the two rows

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}.$$

If we choose a segment with these rows the corresponding part of A' is

$$\begin{pmatrix} 2 & 4 & 1 & 3 & 2 & 3 & 2 & 3 \\ 1 & 4 & 2 & 1 & 6 & 3 & 3 & 2 \end{pmatrix}$$

The relevant columns for $w(e)$ are columns number 2, 3 and 7. Assume

$$c_{vert}(A) = 23, \beta(2) = 20, \beta(3) = 21 \text{ and } \beta(7) = 17.$$

Then

$$\begin{array}{lll} \rho_1(i, 2) = 2, & \rho_1(i, 3) = 3, & \rho_1(i, 7) = 1 \\ \rho_2(i, 2) = 2, & \rho_2(i, 3) = 3, & \rho_2(i, 7) = 0 \end{array}$$

and we obtain

$$w(e) = 2 + 3 - 1 = 4.$$

Here the positive terms correspond to

$$\begin{aligned} \max\{0, a'_{i,2} - a'_{i-1,2}\} &= 0 < 1 = \max\{0, a_{i,2} - a_{i-1,2}\} \quad \text{and} \\ \max\{0, a'_{i,3} - a'_{i-1,3}\} &= 1 < 2 = \max\{0, a_{i,3} - a_{i-1,3}\}, \end{aligned}$$

while the negative term corresponds to

$$\max\{0, a'_{i,7} - a'_{i-1,7}\} = 1 > 0 = \max\{0, a_{i,7} - a_{i-1,7}\}.$$

Finally, we delete all vertices (i, l, r) that lead to segments violating (6). Fix some vertex (i, l, r) and put

$$a'_{i,j} = \begin{cases} a_{i,j} - 1 & \text{if } l \leq j \leq r, \\ a_{i,j} & \text{otherwise.} \end{cases}$$

A segment corresponding to a path through (i, l, r) can not satisfy (6) if $\alpha'(i) := \sum_{j=1}^n \max\{0, a'_{i,j} - a'_{i,j-1}\} > c_{hor}(A) - 1$. But the only terms in $\alpha'(i)$ that could be different from the corresponding terms in $\alpha(i)$ are the terms for $j = l$ and $j = r + 1$ (if $r < n$), and for these terms we have, if $l \leq r$,

$$\begin{aligned} \max\{0, a'_{i,l} - a'_{i,l-1}\} &= \begin{cases} \max\{0, a_{i,l} - a_{i,l-1}\} - 1 & \text{if } a_{i,l} > a_{i,l-1} \\ \max\{0, a_{i,l} - a_{i,l-1}\} & \text{if } a_{i,l} \leq a_{i,l-1}, \end{cases} \\ \max\{0, a'_{i,r+1} - a'_{i,r}\} &= \begin{cases} \max\{0, a_{i,r+1} - a_{i,r}\} + 1 & \text{if } a_{i,r} \leq a_{i,r+1} \\ \max\{0, a_{i,r+1} - a_{i,r}\} & \text{if } a_{i,r} > a_{i,r+1}, \end{cases} \end{aligned}$$

So

$$\alpha'(i) = \begin{cases} \alpha(i) - 1 & \text{if } l \leq r, a_{i,l} > a_{i,l-1} \text{ and } a_{i,r} > a_{i,r+1}, \\ \alpha(i) & \text{if } l = r + 1, \\ \alpha(i) & \text{if } l \leq r, a_{i,l} > a_{i,l-1} \text{ and } a_{i,r} \leq a_{i,r+1}, \\ \alpha(i) & \text{if } l \leq r, a_{i,l} \leq a_{i,l-1} \text{ and } a_{i,r} > a_{i,r+1}, \\ \alpha(i) + 1 & \text{if } l \leq r, a_{i,l} \leq a_{i,l-1} \text{ and } a_{i,r} \leq a_{i,r+1}. \end{cases}$$

Consequently, we have to delete all vertices satisfying one of the following conditions.

1. $\alpha(i) = c_{hor}(A)$ and $r = l - 1$
2. $\alpha(i) = c_{hor}(A)$ and $(a_{i,l} \leq a_{i,l-1}$ or $a_{i,r} \leq a_{i,r+1})$
3. $\alpha(i) = c_{hor}(A) - 1$ and $(l \leq r, a_{i,l} \leq a_{i,l-1}$ and $a_{i,r} \leq a_{i,r+1})$

Now we choose a segment S corresponding to a (D, D') -path of maximal weight.

3 Two orthogonal MLCs

In this section we suppose that two MLCs without ICC are arranged in such a way that the leaf pairs of the one are perpendicular to the leaf pairs of the other, and the segments are the $(0, 1)$ -matrices describing any combination of leaf positions. For instance,

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

is a segment corresponding to the leaf positions shown in Figure 1. We

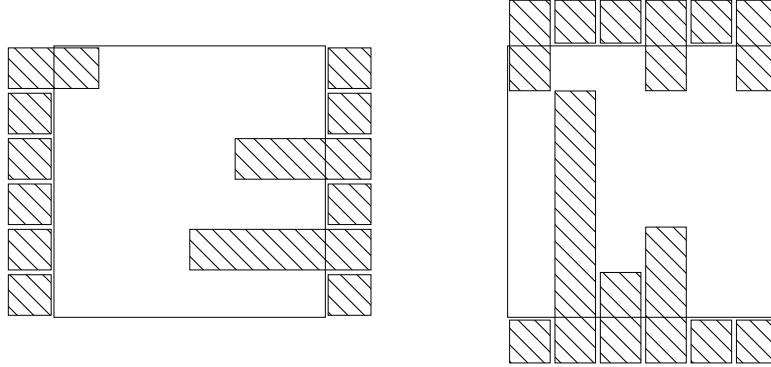


Figure 1: Leaf positions corresponding to the segment given in the text.

formulate the segmentation problem as an integer program. We associate with the segment S the set

$$T(S) = \{(i, j) : 1 \leq i \leq m, 1 \leq j \leq n, s_{i,j} = 1\},$$

and put

$$\mathcal{F} = \{T(S) : S \text{ is a segment}\}.$$

Then a segmentation $A = \sum_{i=1}^k c_i S_i$ can be considered as a function $f : \mathcal{F} \rightarrow \mathbb{N}$ by putting $f(T) = c_i$ if $T = T(S_i)$ and $f(T) = 0$ for the remaining T . Now the segmentation problem can be written as follows

$$(IP) \left\{ \begin{array}{ll} \text{minimize} & \sum_{T \in \mathcal{F}} f(T) \quad \text{subject to} \\ & f(T) \in \mathbb{N} \quad \forall T \in \mathcal{F}, \\ & \sum_{T \in \mathcal{F}: (i,j) \in T} f(T) = a_{i,j} \quad \forall (i,j) \in V. \end{array} \right.$$

An LP-relaxation of this program is

$$(P) \left\{ \begin{array}{ll} \text{minimize} & \sum_{T \in \mathcal{F}} f(T) \quad \text{subject to} \\ & f(T) \geq 0 \quad \forall T \in \mathcal{F}, \\ & \sum_{T \in \mathcal{F}: (i,j) \in T} f(T) = a_{i,j} \quad \forall (i,j) \in V. \end{array} \right.$$

The dual variables (one variable for each $(i, j) \in V$) can be considered as a function $g : V \rightarrow \mathbb{R}$ and in this formulation the dual program is

$$(D) \left\{ \begin{array}{ll} \text{maximize} & \sum_{(i,j) \in V} a_{i,j} g(i, j) \quad \text{subject to} \\ & \sum_{(i,j) \in T} g(i, j) \leq 1 \quad \forall T \in \mathcal{F}. \end{array} \right.$$

Here we suggest a heuristic segmentation method based on dual feasible solutions of a particular type. For pairs (i, j) with $2 \leq i \leq m - 1$ and $2 \leq j \leq n - 1$ we put

$$\gamma_1(i, j) = \max \left\{ \frac{1}{3} (a_{i,s} + a_{i,t} + a_{u,j} + a_{v,j} - a_{i,j}) : \begin{array}{l} 1 \leq s < j < t \leq n, \\ 1 \leq u < i < v \leq m \end{array} \right\}. \quad (7)$$

Then

$$\tilde{c}(A) = \max \{ \gamma_1(i, j) : 2 \leq i \leq m - 1, 2 \leq j \leq n - 1 \}$$

is a lower bound for the TNMU, because for any (i, j) with $2 \leq i \leq m - 1$, $2 \leq j \leq n - 1$ and any numbers s, t, u and v with $1 \leq s < j < t \leq n$ and $1 \leq u < i < v \leq m$ we can define a dual feasible solution g by putting $g(i, s) = g(i, t) = g(u, j) = g(v, j) = \frac{1}{3}$, $g(i, j) = -\frac{1}{3}$ and $g(p, q) = 0$ for all

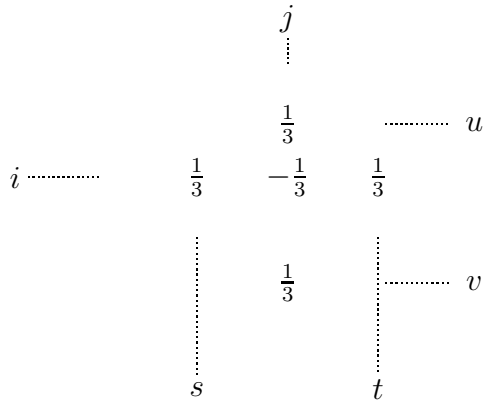


Figure 2: The structure of the dual feasible solution g , where the empty spaces are filled with zeros.

other $(p, q) \in [m] \times [n]$. This is illustrated in Figure 2. If we choose s, t, u and v so that the maximum in (7) is attained the objective value of g is $\gamma_1(i, j)$ and to see that g is feasible we just have to observe that in order to cover the bixel (i, j) we have to cover at least one of the bixels $(i, s), (i, t), (u, j), (v, j)$. Of course, $\tilde{c}(A)$ does not have to be an integer, and since we are using only integer coefficients even $\lceil \tilde{c}(A) \rceil$ can be used as a lower bound for the TNMU. Our algorithm is based on the general principle of extracting segments from A . Precisely, depending on A we determine a segment S , such that

$$A' = A - S$$

is still nonnegative, and then we iterate this step with A' until the zero matrix is reached. The main idea underlying our heuristic approach to the construction of S is that we try to decrease the value of \tilde{c} . Observe that

$$a_{i,j} = 0 \implies s_{i,j} = 0 \quad (i \in [m], j \in [n]) \quad (8)$$

is a necessary and sufficient condition for the nonnegativity of $A' = A - S$. Now we start with S equal to the all-one matrix of dimension $m \times n$ and cover successively all the zero-entries of A . The construction of S is described in Algorithm 2. Lines 2 to 9 ensure that condition (8) is satisfied for bixels (i, j) with $i \in \{1, m\}$ or $j \in \{1, n\}$. These bixels can be covered without influencing other bixels, for instance $(i, 1)$ can be covered by the left leaf of row i , so lines 2 to 9 imply no loss of generality. We still have to make precise, how we choose the bixel (i, j) in line 11 and the direction from which we cover it in line 12. For the choice of the bixel we follow the strategy to cover bixels (i, j) with a high value of $\gamma_1(i, j)$ first. Suppose we have already

Algorithm 2 Segment for two orthogonal MLCs

```
 $s_{i,j} := 1$  ( $i \in [m], j \in [n]$ )  
for  $j = 1$  to  $n$  do  
  if  $a_{1,j} = 0$  then  $s_{1,j} = 0$   
  if  $a_{m,j} = 0$  then  $s_{m,j} = 0$   
5: end for  
for  $i = 1$  to  $m$  do  
  if  $a_{i,1} = 0$  then  $s_{i,1} = 0$   
  if  $a_{i,n} = 0$  then  $s_{i,n} = 0$   
end for  
10: while (8) is violated do  
  Choose an  $(i, j)$  with  $a_{i,j} = 0$  and  $s_{i,j} = 1$   
  Choose a covering direction from {left, right, up, down}  
  Cover bixel  $(i, j)$  with the leaf from the chosen direction  
  Let  $S$  be the segment corresponding to the new leaf positions  
15: end while
```

chosen the bixel to cover. Now we have to choose the direction from which we want to cover it. For instance by covering bixel (i, j) by the left leaf we also cover all bixels (i, j') with $1 \leq j' \leq j$. So it is natural to cover bixel (i, j) from a direction with the property that the maximal entry of a bixel that is covered although the entry is nonzero, is as small as possible. Let

$$\begin{aligned}\gamma_2^{(\text{left})}(i, j) &= \max\{a_{i,j'} : 1 \leq j' \leq j\}, \\ \gamma_2^{(\text{right})}(i, j) &= \max\{a_{i,j'} : j \leq j' \leq n\}, \\ \gamma_2^{(\text{up})}(i, j) &= \max\{a_{i',j} : 1 \leq i' \leq i\}, \\ \gamma_2^{(\text{down})}(i, j) &= \max\{a_{i',j} : i \leq i' \leq m\},\end{aligned}$$

and choose the *covering direction* of bixel (i, j) ,

$$\text{dir}(i, j) \in \{\text{left, right, up, down}\}$$

to be the direction with the smallest value of $\gamma_2^{(*)}(i, j)$. To decide between directions with equal value of $\gamma_2^{(*)}(i, j)$ we consider the number of bixels (i', j') violating the condition for the segment (i.e. with $a_{i',j'} = 0$ and $s_{i',j'} = 1$) that are covered when we cover bixel (i, j) from the respective direction. To

make this precise, let

$$\begin{aligned}\gamma_3^{(\text{left})}(i, j) &= |\{(i, j') : 1 \leq j' < j, a_{i,j'} = 0 \text{ and } s_{i,j'} = 1\}|, \\ \gamma_3^{(\text{right})}(i, j) &= |\{(i, j') : j < j' \leq n, a_{i,j'} = 0 \text{ and } s_{i,j'} = 1\}|, \\ \gamma_3^{(\text{up})}(i, j) &= |\{(i', j) : 1 \leq i' < i, a_{i',j} = 0 \text{ and } s_{i',j} = 1\}|, \\ \gamma_3^{(\text{down})}(i, j) &= |\{(i, j') : i < j' \leq m, a_{i',j} = 0 \text{ and } s_{i',j} = 1\}|,\end{aligned}$$

and choose for $\text{dir}(i, j)$ the direction with in first instance minimal value of $\gamma_3^{(*)}(i, j)$, and in second instance maximal value of $\gamma_3^{(*)}(i, j)$. If there is still a tie it can be decided randomly. Finally, we put

$$\gamma_3(i, j) = \gamma_3^{(\text{dir}(i,j))}(i, j).$$

We choose among the bixels (i, j) with $a_{i,j} = 0$ and $s_{i,j} = 1$ one with in first instance maximal value of $\gamma_1(i, j)$, and in second instance maximal value of $\gamma_3(i, j)$. Now the chosen bixel (i, j) is covered by the leaf that is given by $\text{dir}(i, j)$. Precisely, we put

$$\begin{aligned}s_{i,j'} &:= 0 \quad \text{for} \quad \begin{cases} 1 \leq j' \leq j & \text{if } \text{dir}(i, j) = \text{left}, \\ j \leq j' \leq n & \text{if } \text{dir}(i, j) = \text{right}, \end{cases} \\ s_{i',j} &:= 0 \quad \text{for} \quad \begin{cases} 1 \leq i' \leq i & \text{if } \text{dir}(i, j) = \text{up}, \\ i \leq i' \leq m & \text{if } \text{dir}(i, j) = \text{down}. \end{cases}\end{aligned}$$

This covering step is repeated until (8) is satisfied.

Example 4. Consider the matrix

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

$\gamma_1(2, 2) = 5$ is the maximal value of $\gamma_1(i, j)$, and $(2, 2)$ can be covered from the right with maximal value of a covered bixel equal to 3, while from all other directions we would have to cover a bixel with entry 4. For all the other zero-bixels it is obvious how to cover them without covering a nonzero bixel. So the first segment is

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix},$$

and continuing we obtain a segmentation with 5 segments.

$$\begin{pmatrix} 1 & 4 & 3 & 1 & 5 & 1 \\ 4 & 0 & 3 & 1 & 2 & 1 \\ 2 & 1 & 5 & 4 & 0 & 0 \\ 4 & 4 & 5 & 2 & 3 & 3 \\ 2 & 2 & 2 & 1 & 1 & 1 \\ 5 & 0 & 0 & 0 & 5 & 3 \end{pmatrix} - \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 3 & 2 & 0 & 4 & 0 \\ 3 & 0 & 3 & 1 & 2 & 1 \\ 1 & 0 & 4 & 3 & 0 & 0 \\ 3 & 3 & 4 & 1 & 2 & 2 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 4 & 2 \end{pmatrix},$$

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

$$\begin{pmatrix} 0 & 2 & 1 & 0 & 3 & 0 \\ 2 & 0 & 2 & 0 & 1 & 0 \\ 0 & 0 & 3 & 2 & 0 & 0 \\ 2 & 3 & 3 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 3 & 1 \end{pmatrix} - \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 2 & 0 \\ 1 & 0 & 2 & 0 & 1 & 0 \\ 0 & 0 & 2 & 1 & 0 & 0 \\ 1 & 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 2 & 0 \end{pmatrix},$$

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 2 & 0 \\ 1 & 0 & 2 & 0 & 1 & 0 \\ 0 & 0 & 2 & 1 & 0 & 0 \\ 1 & 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 2 & 0 \end{pmatrix} - \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

4 Test results

We implemented both of the described methods in C++ on a 2GHz workstation. Tables 1 and 2 show test results for random matrices. Each entry is the average TNMU for 1000 matrices with entries from $\{0, 1, \dots, L\}$. For comparison we also included the average minimal TNMU when one MLC is used only in horizontal direction. According to [6] this minimal TNMU is given by

$$\max_{1 \leq i \leq m} \sum_{j=1}^n \max\{0, a_{i,j} - a_{i,j-1}\}.$$

The computation times for the whole heuristic columns were 172 seconds (10×10) and 926 seconds (15×15) for method 2, and 91 seconds (10×10) and 610 seconds (15×15) for method 3.

	Method 1	Method 2		Method 3	
L	TNMU	TNMU	bound	TNMU	bound
3	9.8	7.2	6.7	5.4	3.9
4	12.6	9.0	8.3	6.7	5.2
5	15.5	10.8	10.1	8.0	6.4
6	18.1	12.6	11.8	9.2	7.6
7	20.8	14.3	13.5	10.4	8.8
8	23.6	15.9	14.7	11.7	10.0
9	26.4	17.7	16.6	12.9	11.2
10	29.0	19.5	18.4	14.1	12.4
11	31.8	21.3	19.6	15.4	13.6
12	34.5	23.0	21.9	16.6	14.8
13	36.9	24.6	22.8	17.9	16.0
14	39.8	26.3	24.8	19.1	17.1
15	42.4	28.2	26.2	20.3	18.3
16	45.2	29.8	27.9	21.5	19.5

Table 1: Average TNMU for randomly generated 10×10 -matrices with entries between 0 and L . Method 1: minimal TNMU when the MLC is always used in horizontal direction. Method 2: heuristic results and lower bounds when the MLC can be rotated about 90. Method 3: heuristic results and lower bounds when two orthogonal MLCs are used.

	Method 1	Method 2		Method 3	
L	TNMU	TNMU	bound	TNMU	bound
3	14.0	10.0	8.3	7.4	4.0
4	17.9	12.4	10.2	9.0	5.3
5	21.7	14.8	12.3	10.5	6.7
6	25.6	17.2	14.5	12.0	7.9
7	29.4	19.6	16.2	13.6	9.3
8	33.2	21.9	18.0	15.1	10.5
9	37.0	24.2	20.1	16.6	11.8
10	40.9	26.6	21.8	18.1	13.0
11	44.7	29.0	24.0	19.6	14.2
12	48.5	31.4	25.7	21.1	15.7
13	52.3	33.7	27.8	22.5	16.9
14	56.2	36.1	29.5	24.1	18.1
15	59.8	38.3	31.3	25.5	19.6
16	63.3	40.7	32.6	27.0	20.4

Table 2: Average TNMU for randomly generated 15×15 -matrices with entries between 0 and L . Method 1: minimal TNMU when the MLC is always used in horizontal direction. Method 2: heuristic results and lower bounds when the MLC can be rotated about 90. Method 3: heuristic results and lower bounds when two orthogonal MLCs are used.

case no.	MU 1	MU 2	MU 3
1	16	11	11
2	16	11	11
3	20	12	11
4	19	12	11
5	15	13	10
6	17	12	12
7	18	13	11
8	22	14	13
9	26	16	13
10	22	14	14
11	22	15	14
12	23	16	14
13	23	14	14

Table 3: Test results for clinical matrices: the TNMU with one MLC in one direction (MU 1), with one MLC in two directions (MU 2) and with two orthogonal MLCs (MU 3).

References

- [1] D. Baatar and H.W. Hamacher. New LP model for multileaf collimators in radiation therapy. contribution to the conference ORP3, University of Kaiserslautern, 2003.
- [2] 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.
- [3] 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.
- [4] A.L. Boyer and C.Y. Yu. Intensity-modulated radiation therapy with dynamic multileaf collimators. *Semin. Radiat. Oncol.*, 9:48–59, 1999.
- [5] J. Dai and Y. Zhu. Minimizing the number of segments in a delivery sequence for intensity-modulated radiation therapy with a multileaf collimator. *Med. Phys.*, 28:2113–2120, 2001.
- [6] K. Engel. A new algorithm for optimal multileaf collimator field segmentation. Preprint 03/5, Fachbereich Mathematik, Uni Rostock, 2003.
- [7] 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.
- [8] T. Kalinowski. Realization of intensity modulated radiation fields using multileaf collimators. In R. Ahlswede, L. Bäumer, and N. Cai, editors, *General Theory of Information Transfer and Combinatorics*. Shannon Foundation, to be published 2004. Report on a Research Project at the ZIF (Center of interdisciplinary research) in Bielefeld Oct. 1, 2002 – August 31, 2003.
- [9] 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.
- [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] W. Que. Comparison of algorithms for multileaf collimator field segmentation. *Med. Phys.*, 26:2390–2396, 1999.

- [12] 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.
- [13] P. Xia and L. Verhey. Multileaf collimator leaf–sequencing algorithm for intensity modulated beams with multiple static segments. *Med. Phys.*, 25:1424–1434, 1998.