An algorithm for optimal multileaf collimator field segmentation with interleaf collision constraint 2

Thomas Kalinowski
Universität Rostock
Fachbereich Mathematik
D–18051 Rostock

July, 2003

Abstract

Multileaf collimators are widely used in radiotherapy to realize intensity modulated fields as superpositions of homogeneous fields, so called segments. One important step in the planning process is the decomposition of the modulated field into a small number of segments such that the total number of monitor units is also small. In this paper we present an algorithm that is based on the results of [6] and constructs a segmentation with minimal total number of monitor units and a small number of segments, taking into account a machine–dependent constraint, that forbids leaf overtravel in adjacent rows of the multileaf collimator.

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

1 Introduction

In radiotherapy a modulated field is described by an intensity map. After discretization such an intensity map can be considered as a nonnegative integer $m \times n$–matrix $A = (a_{i,j})$. A realization of the modulated intensity with a multileaf collimator corresponds to a representation of A as a nonnegative combination of special $(0,1)$–matrices, called segments, which describe the possible leaf positions of the collimator. Throughout, for $n \in \mathbb{N}$, let $[n]$ denote the set $\{1, 2, \ldots, n\}$.

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

$$l_i \leq r_i + 1 \qquad (i \in [m]), \tag{1}$$

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

$$\text{ICC:} \quad l_i \leq r_{i+1} + 1,\ r_i \geq l_{i+1} - 1 \qquad (i \in [m-1]). \tag{3}$$

The interpretation is that $l_i - 1$ and $r_i + 1$ are the positions of the i–th left and right leaf, respectively. (3) is the formal description of the interleaf collision constraint (ICC), a technological restriction present in many widely used multileaf collimators, which forbids the overtravel of opposite leafs in adjacent rows. A segmentation of A is a representation of A as a sum of segments, i.e.

$$A = \sum_{i=1}^{k} u_i S_i \tag{4}$$

with segments S_i $(i = 1, 2, \ldots, k)$ and positive integers u_i $(i = 1, 2, \ldots, k)$. To a segmentation of A there corresponds a treatment plan realizing the intensity map given by A. There are two obvious measures for the quality of the segmentation: the total number of monitor units (TNMU) and the number of segments (NS) which are given by $\sum_{i=1}^{k} u_i$ and k, respectively. Clearly, both of these should be minimized in order to optimize the treatment plan. In the literature there are several algorithms for the construction of segmentations ([2–8]), some of them minimizing the TNMU, others reducing the NS at the cost of an increased TNMU. In [1] it is shown that the NS–minimization is NP–complete even for one row. So it might be a good strategy to look for an algorithm that minimizes the TNMU and approximately minimizes the NS.

2 The Algorithm

In order to avoid case distinctions later on we add two zero columns left and right of the matrix A, i.e. we put

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

Further let

$$d_{i,j} = a_{i,j} - a_{i,j-1} \qquad (i \in [m],\ j \in [n]).$$

In [6] it is proved that the minimal TNMU in a segmentation of A equals

$$c(A) = \max\{\delta(P)\ :\ P \text{ is a } (0,1)-\text{path in } \overrightarrow{G}\},$$

where $\overrightarrow{G} = (V \cup \{0,1\}, E)$ is a digraph with $V = [m] \times [n]$ and $E = \bigcup_{i=1}^{4} E_i$ with

$$\begin{aligned}
E_1 &= \{(0,(i,1))\ :\ i \in [m]\} \cup \{((i,n),1)\ :\ i \in [m]\},\\
E_2 &= \{((i,j),(i,j+1)\ :\ i \in [m],\ j \in [n-1])\},\\
E_3 &= \{((i,j),(i+1,j))\ :\ i \in [m-1],\ j \in [n]\},\\
E_4 &= \{((i,j),(i-1,j))\ :\ 2 \le i \le m,\ j \in [n]\},
\end{aligned}$$

and the length function δ on E is defined by

$$\begin{aligned}
\delta(0,(i,1)) &= a_{i,1} && (i \in [m]),\\
\delta((i,n),1) &= 0 && (i \in [m]),\\
\delta((i,j),(i,j+1)) &= \max\{0, d_{i,j+1}\} && (i \in [m],\ j \in [n-1]),\\
\delta((i,j),(i+1,j)) &= -a_{i,j} && (i \in [m-1],\ j \in [n]),\\
\delta((i,j),(i-1,j)) &= -a_{i,j} && (2 \le i \le m,\ j \in [n]).
\end{aligned}$$

Adopting the terminology of [4] we call the pair (u,S) of a positive number u and a segment S an *admissible segmentation pair* if

$$\begin{aligned}
A' = A - uS &\text{ is nonnegative and}\\
c(A') = c(A) - u.&
\end{aligned}$$

The essential step of our algorithm is to determine the maximal coefficient u with the property that there exists a segment S, such that (u,S) is an admissible segmentation pair. Iterating this step with $A' = A - uS$ we clearly obtain a segmentation of A with $c(A)$ monitor units. In order to derive an upper bound for the coefficient u in an admissible segmentation pair (u,S), we identify, according to [2], the set of segments with the set of paths from D to D' in the layered digraph $H = (V, E)$, constructed as follows. The vertices in the $i-$th layer correspond to the possible leaf positions in row i $(1 \le i \le m)$ and two additional vertices D and D' are added:

$$V = \{(i, l, r) : i = 1, \ldots, m,\ l = 1, \ldots, n+1,\ r = l-1, \ldots, n\} \cup \{D, D'\}.$$

Between two vertices (i, l, r) and $(i+1, l', r')$ there is an edge if the corresponding leaf positions are consistent with the ICC, i.e. if $l' \le r+1$ and $r' \ge l-1$. In addition E contains all edges from D to the first layer and from the last layer m to D', so

$$E = E_+(D) \cup E_-(D') \cup \bigcup_{i=1}^{m-1} E_+(i), \text{ where}$$

$$\begin{aligned} E_+(D) &= \{(D, (1, l, r)) \ :\ (1, l, r) \in V\}, \\ E_-(D) &= \{((m, l, r), D') \ :\ (m, l, r) \in V\}, \\ E_+(i) &= \{((i, l, r), (i+1, l', r')) : l' \le r+1,\ r' \ge l-1\}. \end{aligned}$$

There is a bijection between the possible leaf positions and the paths from D to D' in H. This is illustrated in Fig. 1 which shows a path in H for $m = 4$, $n = 2$, corresponding to the segment

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

Assume, for every triple (i, l, r), $1 \le i \le m$, $1 \le l \le r+1 \le n+1$, we have already determined some upper bound $u_0(i, l, r)$ for the coefficient u in an admissible segmentation pair (u, S) where S is a segment with $l_i = l$ and $r_i = r$. That is $u \le u_0(i, l_i, r_i)$ for all i if (u, S) is an admissible segmentation pair and l_i, r_i $(i = 1, \ldots, m)$ are the parameters of S. We put

$$\hat{u} = \max\{u \ :\ \text{There is a path } D, (1, l_1, r_1), \ldots, (m, l_m, r_m), D' \\ \text{in } H \text{ with } u_0(i, l_i, r_i) \ge u \text{ for } i = 1, \ldots, m\}.$$

Clearly, $\hat{u}$ is an upper bound for the coefficient u in an admissible segmentation pair (u, S). Now we describe an algorithm which constructs an admissible segmentation pair (u, S) with maximal u. Fix u and assume we have already determined the first $i-1$ rows of a segment. If it is possible to complete these $i-1$ rows to obtain a segment S such that (u, S) is an admissible segmentation pair, then the procedure `Complete Segment`(i) determines $l_i, \ldots, l_m$ and $r_i, \ldots, r_m$ which realize such a completion. Here `MaxLength`(i) denotes the maximal length of a path in $\overrightarrow{G}$ that has all its vertices in the first i rows.

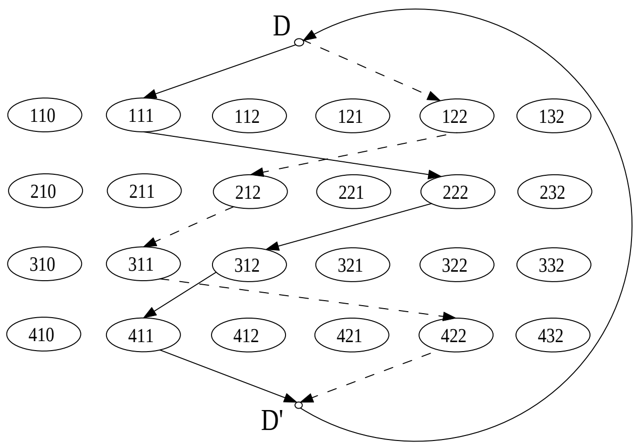

Figure 1: The vertices of H for $m = 4$, $n = 2$ and a (D, D')–path.

Procedure `Complete Segment`(i)
for (l_i, r_i) with $1 \leq l_i \leq r_{i-1} + 1$,
  $\max\{l_i, l_{i-1}\} - 1 \leq r_i \leq n$ and
  $u_0(i, l_i, r_i) \geq u$ **do**
  if `MaxLength`$(i) \leq c(A) - u$ **then**
    if $i < m$ **then**
      `Complete Segment`$(i + 1)$
    else
      finished:=true
    end if
  end if
end for

Now the pair (u, S) is constructed as follows:

Procedure `Construct Segment`
$u := \hat{u}$
finished:=false
$l_{-1} := 1$, $r_{-1} := n + 1$
while not finished **do**
  `Complete Segment`(1)
  if not finished **then**
    $u := u - 1$
  end if
end while

Clearly, the efficiency of the backtracking depends very much on the quality of the bounds $u_0(i, l, r)$. We give some bounds that turned out to be quite good in numerical experiments. Trivially, in an admissible segmentation pair (u, S) we have, for all i,

$$u \leq v_1(i, l_i, r_i) := \min\{a_{i,j} \ : \ l_i \leq j \leq r_i \leq n\}.$$

For $(i, j) \in V$ we use the notation

$$\alpha_1(i,j) = \max\{\delta(P) \ : \ P \text{ is a } (0,(i,j)) - \text{path in } \overrightarrow{G}\}, \tag{5}$$

$$\alpha_2(i,j) = \max\{\delta(P) \ : \ P \text{ is a } ((i,j),1) - \text{path in } \overrightarrow{G}\}. \tag{6}$$

To avoid case distinctions we also put $\alpha_1(i, 0) = \alpha_2(i, n+1) = 0$. Fix an admissible segmentation pair (u, S), denote by δ' the length function on $\overrightarrow{G}$ corresponding to $A' = A - uS$ and let

$$\alpha'_1(i,j) = \max\{\delta'(P) \ : \ P \text{ is a } (0,(i,j)) - \text{path in } \overrightarrow{G}\}, \tag{7}$$

$$\alpha'_2(i,j) = \max\{\delta'(P) \ : \ P \text{ is a } ((i,j),1) - \text{path in } \overrightarrow{G}\}. \tag{8}$$

The upper bounds below are based on the following simple observations

1. The only edges e with $\delta'(e) < \delta(e)$ are of the form $e = ((i, l_i - 1), (i, l_i))$ $(1 \leq i \leq m)$, and for these edges $\delta'(e) \geq \delta(e) - u$.

2. For edges of the form $e = ((i, j), (i \pm 1, j))$ $(l_i \leq j \leq r_i)$ we have $\delta'(e) = \delta(e) + u$.

3. If $j < l_k$ for some $k \in [m]$ then, on every $(0, (k, j))-$path P, the number of edges of the form $((i, l_i - 1), (i, l_i))$ is equal to or less than the number of edges of the form $((i, j), (i \pm 1, j))$ with $l_i \leq j \leq r_i$.

4. If $j \geq l_k$ for some $k \in [m]$ then, for every $((k, j), 1)-$path P, the number of edges of the form $((i, l_i - 1), (i, l_i))$ is equal to or less than the number of edges of the form $((i, j), (i \pm 1, j))$ with $l_i \leq j \leq r_i$.

The third and the fourth observations are illustrated in Figure 2.
It follows, for $1 \leq i \leq m$,

$$\begin{aligned} \alpha'_1(i,j) &\geq \alpha_1(i,j) && \text{for } j < l_i, \\ \alpha'_2(i,j) &\geq \alpha_2(i,j) && \text{for } j \geq l_i. \end{aligned}$$

Lemma 1. *Let (u, S) be an admissible segmentation pair with $l_i = l$ and $r_i = r$. Then $u \leq v_2(i, l, r)$ where*

$$v_2(i, l, l-1) = c(A) - \alpha_1(i, l-1) - \max\{0, d_{i,l}\} - \alpha_2(i, l),$$

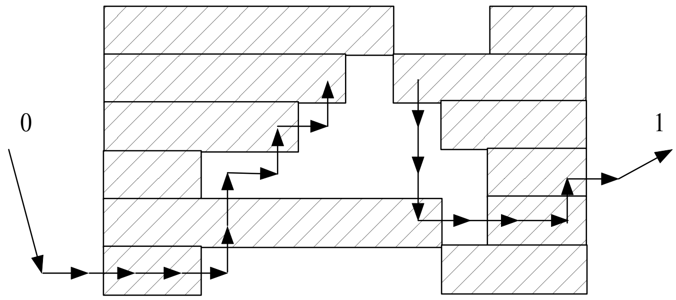

Figure 2: The area which is covered by the leafs is shaded. On any path with both end vertices in the shaded region, for every arc that enters the white region there must be an arc leaving the white region. If the last vertex is left of the white region (as in 3. above) every arc leaving the white region is a vertical one. Similarly, if the starting point of the path is right of the white region (as in 4. above), for every horizontal arc entering the white region there must be a vertical arc leaving the white region.

and if $r \geq l$ *then* $v_2(i,l,r) = \min\{\gamma_1, \gamma_2, \gamma_3, \gamma_4\}$, *where*

$$\gamma_1 = c(A) - \alpha_1(i, l-1) - \alpha_2(i, l),$$

$$\gamma_2 = c(A) - \alpha_1(i, l-1) - \sum_{j=l+1}^{r} \max\{0, d_{i,j}\} - \alpha_2(i, r+1),$$

$$\gamma_3 = c(A) - \alpha_1(i, l-1) - d_{i,l} - \sum_{j=l+1}^{r} \max\{0, d_{i,j}\} - d_{i,r+1} - \alpha_2(i, r+1),$$

$$\gamma_4 = \frac{1}{2}\left(c(A) - \alpha_1(i, l-1) - \sum_{j=l+1}^{r} \max\{0, d_{i,j}\} - d_{i,r+1} - \alpha_2(i, r+1)\right).$$

Proof. Let P be the the concatenation of the paths P_1, P_2 and P_3, where P_1 is a $(0,(i,l-1))-$path with $\delta(P_1) = \alpha_1(i,l-1)$, P_2 is the path $((i,l-1),(i,l),\ldots,(i,r+1))$, and P_3 is an $((i,r+1),1)-$path with $\delta(P_2) = \alpha_2(i,r+1)$.

Case 1: $r = l-1$. Using the above observations, we obtain

$$c(A) - u = c(A') \geq \delta'(P) \geq \alpha_1(i,l-1) + \max\{0, d_{i,l}\} + \alpha_2(i,l),$$

and thus $u \leq c(A) - \alpha_1(i,l-1) - \max\{0, d_{i,l}\} - \alpha_2(i,l)$.

Case 2: $r \geq l$. Now

$$\delta'(P) = \delta'(P_1) + \max\{0, d_{i,l} - u\} + \sum_{j=l+1}^{r} \max\{0, d_{i,j}\} \\ + \max\{0, d_{i,r+1} + u\} + \delta'(P_2),$$

and thus

$$\alpha_1(i, l-1) + \max\{0, d_{i,l} - u\} + \sum_{j=l+1}^{r} \max\{0, d_{i,j}\} \\ + \max\{0, d_{i,r+1} + u\} + \alpha_2(i, r+1) \leq c(A) - u,$$

or

$$u + \max\{0, d_{i,l} - u\} + \max\{0, d_{i,r+1} + u\} \leq c(A) - \alpha_1(i, l-1) \\ - \sum_{j=l+1}^{r} \max\{0, d_{i,j}\} - \alpha_2(i, r+1),$$

which implies $u \leq \gamma_i$ ($i = 2, 3, 4$). To see $u \leq \gamma_1$, consider the path Q that is the concatenation of P_1, the edge $((i, l-1), (i, l))$ and an $((i, l), 1)-$path P_4 with $\delta(P_4) = \alpha_2(i, l)$. Then

$$\delta'(Q) \geq \alpha_1(i, l-1) + \alpha_2(i, l),$$

and thus $u \leq \gamma_1$.

□

Lemma 2. *Suppose (u, S) is an admissible segmentation pair, fix some i, $2 \leq i \leq m-1$, and put*

$$\lambda_1 = \max_{l_i \leq t \leq r_i} \{\alpha_1(i-1, t) - a_{i-1,t} - a_{i,t} + \alpha_2(i+1, t)\},$$
$$\lambda_2 = \max_{l_i \leq t \leq r_i} \{\alpha_1(i+1, t) - a_{i+1,t} - a_{i,t} + \alpha_2(i-1, t)\}.$$

Then

$$u \leq v_3(i, l_i, r_i) := c(A) - \min\{\lambda_1, \lambda_2\}.$$

Proof. By symmetry, w.l.o.g. $\lambda_1 \leq \lambda_2$. Assume $u > c(A) - \lambda_1$, and let t be the index where the maximum in the definition of λ_1 is attained. Let P be the concatenation of the three paths P_1, P_2 and P_3, where P_1 is an

$(0,(i-1,t))-$path with $\delta(P_1)=\alpha_1(i-1,t)$, $P_2=((i-1,t),(i,t),(i+1,t))$ and P_3 is an $((i+1,t),1)-path$ with $\delta(P_3)=\alpha_2(i+1,t)$. Then

$$\delta'(P)\le c(A')=c(A)-u<\lambda_1=\delta(P).$$

By the above observations, we have $\delta'(P_1)\ge\delta(P_1)-u$, $\delta'(P_3)\ge\delta(P_3)-u$ and

$$\delta'(P_2)=\begin{cases}\delta(P_2)+2u & \text{if } l_{i-1}\le t\le r_{i-1},\\ \delta(P_2)+u & \text{otherwise.}\end{cases}$$

So $\delta'(P)<\delta(P)$ implies

$$\begin{aligned}&\delta'(P_1)<\delta(P_1),\\ &\delta'(P_2)=\delta(P_2)+u,\\ &\delta'(P_3)<\delta(P_3).\end{aligned}$$

And from this follows

$$l_{i-1}\le t\quad\text{and}\quad l_{i+1}>t.$$

Now denote by t' the index where the maximum in the definition of λ_2 is attained. Since $u>c-\lambda_1\ge c-\lambda_2$, by an analoguous argument we obtain

$$l_{i+1}\le t'\quad\text{and}\quad l_{i-1}>t'.$$

But this is a contradiction to $l_{i+1}>t$ if $t'\le t$ and to $l_{i-1}<t$ if $t'>t$. □

Thus we may put

$$u_0(i,l,r)=\min\{v_k(i,l,r)\ :\ k=1,2,3\}. \tag{9}$$

Theorem 3. *If the* $u_0(i,l,r)$ *are determined according to* (9) *the algorithm* `Construct Segment` *yields an admissible segmentation pair* (u,S) *such that* $u'\le u$ *for any admissible segmentation pair* (u',S')*.*

3 Results

To test our algorithm we computed segmentations for 15×15–matrices with random entries from $\{0,1,\dots,L\}$ for $3\le L\le 16$. Table 1 shows the results. The numbers in the comlumns TNMU (new) and NS (new) are the average total number of monitor units and the average number of segments, where we have averaged over 10000 matrices with randomly chosen entries from $\{0,\dots,L\}$ (uniformly distributed). The remaining comlumns show the corresponding results for some other algorithms that were proposed for the segmentation problem. These numbers are taken from [8]. The columns labeled X-V, B, G contain the results for the algorithms of Xia and Verhey [8], Bortfeld *et al.* [3] and Galvin *et al.* [5], respectively. On an 1.3GHz–PC the computation of the two new entries in a row of the table, i.e. the segmentation of 10000 matrices, took approximately 1 hour. But it should be mentioned that the algorithm is fast for the vast majority of the matrices, while there are some very rare exceptions. This is illustrated in Table 2.

L	TNMU (new)	TNMU (X–V)	TNMU (B)	TNMU (G)	NS (new)	NS (X–V)	NS (B)	NS (G)
3	15.4	19.5	17.7	19.7	12.6	13.3	17.7	13.4
4	19.5	29.6	22.8	40.5	14.5	18.6	22.8	20.4
5	23.6	30.9	27.9	40.1	16.0	19.0	27.9	20.4
6	27.6	46.8	32.8	44.2	17.2	20.3	32.8	21.5
7	31.7	45.6	37.9	67.1	18.2	20.0	37.9	27.1
8	35.7	63.4	42.8	72.3	19.1	24.3	42.8	28.2
9	39.8	67.1	47.8	72.3	19.9	24.3	47.8	28.3
10	43.8	68.6	52.6	76.5	20.7	25.7	52.6	28.9
11	47.7	68.6	57.6	81.4	21.3	25.7	57.6	30.9
12	51.8	101.1	62.4	106.8	21.9	27.0	62.4	34.8
13	55.7	100.6	67.3	101.1	22.5	26.9	67.3	35.5
14	59.8	100.0	72.2	112.7	23.0	26.9	72.2	35.6
15	63.8	98.0	77.1	116.0	23.5	26.7	77.1	35.9
16	67.7	124.9	82.0	154.5	24.0	30.0	82.0	41.7

Table 1: Test results for $m = n = 15$.

CPU–time	≤ 1 sec	≤ 2 sec	≤ 10 sec	≤ 1 min
number of matrices	7879	9972	9992	9999

Table 2: The table shows the numbers of matrices that were treated in the time given in the first row, when altogether segmentations for 10000 15×15–matrices with random entries between 0 and 16 were computed.

4 Discussion

We presented an algorithm for the construction of a treatment plan realizing a given intensity map with a multileaf collimator in the static mode which takes into account the interleaf collision constraint and constructs a segmentation with minimal TNMU and a small NS. The algorithm is based in the repeated subtraction of segments with the maximal possible coefficient. The drawback of the method is the computational complexity. There are matrices where the search for the maximal possible coefficient is very time–consuming, and due to the used backtracking the computation time grows rapidly with the problem size. So further research with the aim to overcome these problems should be devoted to the determination of the maximal coefficient.

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. under revision for NETWORKS, 2002.

[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] K. Engel. A new algorithm for optimal multileaf collimator field segmentation. Preprint 03/5, Fachbereich Mathematik, Uni Rostock, 2003.

[5] 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.

[6] Thomas Kalinowski. An algorithm for optimal multileaf collimator field segmentation with interleaf collision constraint. Preprint 03/2, Fachbereich Mathematik, Uni Rostock.

[7] 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.

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